Inaugural-Dissertation

zur Erlangung der Doktorwürde der

Gesamtfakultät für Mathematik,
Ingenieur- und Naturwissenschaften

der

Ruprecht-Karls-Universität
Heidelberg

vorgelegt von

Ghadeer Mobasher, M.Sc.

aus Kairo

Tag der mündlichen Prüfung:

WeLT: Weighted Loss Trainer for Biomedical Joint Entity and Relation Extraction

by

Ghadeer Mobasher

Supervisors: Prof. Dr. Michael Gertz
Priv. -Doz. Dr. Wolfgang Müller

Abstract

The exponential growth of unstructured textual data has emphasised the need for Information Extraction (IE) to transform raw text into actionable knowledge. IE involves automatically identifying and categorising relevant entities, relationships, and events within large text corpora. The ability to extract pertinent information from vast and complex datasets automatically and accurately has profound implications, from advancing personalised medicine and clinical research to enhancing the efficiency of information flow in news and media outlets. Pre-annotations generated by IE systems help alleviate the labour-intensive workload of data annotators by automating the initial labelling of entities, relationships, and events. This automation reduces the need for manual identification, allowing annotators to focus on verifying and refining the pre-annotated data, which significantly speeds up the annotation process.

Supervised learning is one of the primary IE approaches that involve using labelled datasets to train models. Thus, there are considerable efforts by domain experts to curate gold-standard datasets. However, real-world data frequently inherit class imbalance, which remains a significant challenge in IE, where more frequent majority classes often overshadow minority classes that represent rare but critical entities. This imbalance leads to degraded performance, particularly in recognising and extracting under-represented classes.

Current literature offers several approaches to mitigate class imbalance, such as undersampling, over-sampling, and static weighting loss. However, these methods have notable drawbacks. Oversampling can lead to over-fitting while undersampling risks discarding valuable data. Fixed weighting loss schemes require extensive manual hyper-parameter tuning, which is time-consuming and often fails to adapt to the unique characteristics of a dataset. These approaches do not address the core issue: the need for the model to adaptively learn from the natural class distribution without biasing its performance towards majority classes.

In response to these limitations, this thesis introduces the **We**ighted **L**oss **T**rainer (WeLT), a novel adaptive loss function designed to address class imbalance. WeLT adjusts class weights based on the relative frequency of each class within the dataset, ensuring that misclassifications of minority classes are penalised more heavily. This approach allows the model to remain sensitive to minority classes without requiring extensive manual tuning or compromising data integrity.

Evaluations conducted on gold-standard datasets, including biomedical and newswire datasets, focused on Named Entity Recognition (NER) and Joint Named Entity Recognition and Relation Extraction (JNERE). Specifically, WeLT was tested on two JNERE paradigms: (a) span-based and (b) table-filling approaches. Additionally, the impact of WeLT NER on Named Entity Linking was compared to vanilla NER methods that neglect class imbalance. Our experiments demonstrate that WeLT effectively addresses class imbalance issues, outperforming traditional fine-tuning approaches and proving advantages over existing weighting loss schemes.

Die exponentielle Zunahme unstrukturierter Textdaten steigert die Notwendigkeit von Informationsextraktion (IE), also von Techniken, unstrukturierte, menschenlesbare Texte in besser verwertbares Wissen umzuwandeln. IE umfasst das automatische Identifizieren und Kategorisieren relevanter Entitäten, Beziehungen und Ereignisse in großen Textkorpora. Die Fähigkeit, automatisch und präzise relevante Informationen aus umfangreichen und komplexen Datensätzen zu extrahieren, hat breite Anwendungsmöglichkeiten: Sie reichen von der personalisierten Medizin und klinischer Forschung bis hin zur Effizienzsteigerung im Informationsfluss von Nachrichten- und Medienportalen. Von IE-Systemen generierte Annotations-Vorschläge tragen dazu bei, die Belastung von Datenannotatoren zu verringern, indem sie die Kennzeichnung von Entitäten, Beziehungen und Ereignissen automatisieren. Diese Automatisierung reduziert die Notwendigkeit manueller Identifikation und ermöglicht es den Annotatoren, sich auf das Überprüfen und Verfeinern der vorher annotierten Daten zu konzentrieren, sowie auf das Erkennen komplexer Relationen. Dies hat das Potential, den Prozess der Datenkuratierung erheblich zu beschleunigen. Überwachtes Lernen ist einer der wichtigsten Ansätze zur IE, der die Nutzung von bereits annotierten Datensätzen zur Modellentwicklung beinhaltet. Daher gibt es beträchtliche Bemühungen von Fachexperten, Standard-Datensätze zu erstellen. Jedoch weisen reale Daten häufig eine Ungleichverteilung der Daten auf, die in der IE eine große Herausforderung darstellt, da häufig vorkommende Klassen (Mehrheitsklassen) die Minderheitsklassen, die seltene, aber eigentlich interessante Entitäten repräsentieren, oft überlagern. Diese Ungleichverteilung führt zu Leistungseinbußen, insbesondere bei der Erkennung und Extraktion unterrepräsentierter Klassen. Die aktuelle Literatur bietet mehrere Ansätze zur Behandlung von Klassenungleichheit, darunter Undersampling, Oversampling und statische Gewichtung der Verlustfunktion. Diese Methoden haben jedoch erhebliche Nachteile: Oversampling kann zu Overfitting, also einem Verlust der Generalisierungsfähigkeiten des Modells führen, während beim Undersampling potentiell wertvolle Daten aus der Trainingsmenge gestrichen werden. Feste Verlustgewichtungsschemata erfordern eine umfassende manuelle Einstellung der Hyperparameter, die zeitaufwendig ist und häufig nicht in der Lage ist, sich an die spezifischen Merkmale eines Datensatzes anzupassen. Diese Ansätze adressieren nicht das Kernproblem: Das Erfordernis, dass das Modell adaptiv aus der natürlichen Klassenverteilung lernt, ohne seine Leistung rein auf Mehrheitsklassen auszurichten. Als Antwort auf diese Einschränkungen führt diese Dissertation den **We**ighted **L**oss **T**rainer (WeLT) ein, eine neuartige adaptive Verlustfunktion zur Bewältigung von Klassenungleichverteilung. WeLT passt Klassengewichtungen basierend auf der relativen Häufigkeit jeder Klasse im Datensatz an und stellt sicher, dass Fehlklassifikationen von Minderheitsklassen beim Lernen stärker gewichtet werden. Dieser Ansatz ermöglicht es dem Modell, empfindlich gegenüber Minderheitsklassen

zu bleiben, ohne umfangreiche manuelle Anpassungen oder den Verlust von Datenintegrität zu erfordern.

Evaluierungen, die im Rahmen dieser Dissertation auf Standard-Datensätzen – darunter biomedizinische und Nachrichten-Korpora – durchgeführt wurden, konzentrierten sich auf die Erkennung benannter Entitäten (NER) und die gleichzeitige Erkennung benannter Entitäten und Relationsextraktion (Joint Named Entity and Relation Extraction, JNERE). Konkret wurde WeLT in zwei JNERE-Ansätzen getestet: (a) Span-basierte und (b) tabellenfüllende Ansätze. Darüber hinaus wurde die Auswirkung von WeLT NER auf die Verknüpfung benannter Entitäten mit der von herkömmlichen NER-Methoden verglichen, die die Klassenungleichverteilung unberücksichtigt lassen. Unsere Experimente zeigen, dass WeLT das Problem der Klassenungleichverteilung effektiv angeht und herkömmliche Fine-Tuning-Ansätze übertrifft und gegenüber bestehenden Verlustgewichtungsschemata Vorteile bietet.

Though absent you are very near; still loved, still missed, and always dear.

To the late Professor Osman Ibrahim and my late grandmother Fatma Ali.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Dr. Michael Gertz and Priv.-Doz. Dr. Wolfgang Müller. Micheal, your unwavering commitment to my academic development has been truly invaluable. Your insightful feedback and constructive criticism have consistently elevated the quality of my work. I especially appreciate your emphasis on clarity and precision in research, teaching me that simplicity often leads to stronger, more impactful results. Your patience with my writing mistakes and your willingness to guide me through them have been remarkable. Beyond your academic expertise, your mentorship has deeply shaped my research approach, teaching me to prioritise careful thought and precision in every aspect of my work. For all of this, I am deeply thankful. Wolfgang, you have been much more than just a supervisor and boss. You have truly become like family to me. From our very first day at HITS, when you warmly welcomed me and drove me back to Mathematikon, to the countless moments of support you provided throughout my journey, I have felt incredibly fortunate to work with you. I have learned from you that life is a marathon, not a sprint. Your wisdom has helped me navigate both the challenges of research and the larger journey of life. I have always been amazed by your engaging presentation skills and your great sense of humour, and I aspire to develop these skills myself. I am also deeply grateful for your efforts to prolong my contract and secure me financially throughout my PhD journey, which provided me with the stability to focus on my research. Thank you for giving me the freedom to explore different directions in my work and for always being supportive of my ideas. Our weekly Jour Fixe meetings were a cornerstone of my progress, and I am deeply grateful for them. Our hiking talks during group retreats were also particularly special, allowing me to open up and share my thoughts and feelings. “Thank you” feels like an understatement for all that you have done for me. Your support, both professionally and personally, has been invaluable.

I would also like to extend my heartfelt thanks to my greatest mentor, Prof. Dr. Gerard McKee. Despite not being my official supervisor, your unwavering dedication to supporting me throughout both my master’s and PhD journeys has been nothing short of extraordinary. Your meticulous proof-reading of my theses and our weekly discussions have had a remarkable impact on my work and personal growth. Your encouragement during the challenging stages of writing and your patience with my mistakes have been a beacon of guidance. I truly miss working with you and sincerely hope that our paths will cross again in the future. In addition, I would like to thank Prof. Dr. Francisco M. Couto and Pedro Ruas for their support and collaboration during

my secondment at LASIGE, Faculty of Sciences, University of Lisbon. Your guidance and insights during my time there were invaluable, and I am grateful for the opportunity to work alongside you both. I also wish to extend my gratitude to Dr. Rolland Roller from the DKFI NLP team for accepting me for a short-term secondment. Your help in planning my thesis outline was greatly appreciated.

I am grateful to have been part of the PoLiMeR project, where I had the honour of meeting all the principal investigators and my fellow 14 PhD colleagues. The collaborative environment and shared experiences within the project were invaluable to my development. I would also like to extend my heartfelt thanks to my colleagues at HITS, both past and present, who have been a part of this journey. Olga Krebs, thank you for helping me bridge the gap between biomedical and computer science researchers, and for cooking the most delicious brunches. Lukrécia Mertová, thank you for being the best work buddy. Special thanks to Ulrike Wittig and Maja Rey for reviewing the German abstract. Dennis Aumiller, I am grateful for your kindness both professionally and personally, and for your valuable suggestions when reading parts of my published papers. Ashish Chouhan, I cherish our relaxed coffee meet-ups and appreciate you listening to my complaints and sharing your insights. Thank you also for reading chapters of my thesis. Satya Almasian, thank you for taking the time to read a chapter of my thesis while you were wrapping up your own. I would also like to thank Maha Riad, with whom I had the pleasure of sharing our academic journey up to the PhD. Our morning and evening productive online writing sessions were invaluable.

To my friends who are like family, Heba Mohamed *(the greatest mentor)*, Basma Hathout *(the best twin version of me)*, Mehwish Fatima *(the most optimistic)*, Valentina Gárate Calderón *(the hard-working)*, and Jenny Yassa *(the childhood best friend)*. Thank you for always being patient when I could not catch up regularly. I am truly blessed to have this sisterhood.

To my family, I cannot express enough gratitude for your unwavering support. To my father, *Hesham*—your drive and motivation have been the forces behind my academic progress. Your constant encouragement always fires me up to achieve more. To my mother, *Ghada*—thank you for cheering me on, praying for me, and patiently listening to my complaints. You both have sacrificed so much to raise three children and invest in our education despite all the external pressures, and for that, I am forever grateful. To my sister, *Mahinour*—our future doctor and my companion in this academic journey—thank you for believing in me even when I did not believe in myself. Your support has meant the world to me, from travelling all the way from Paris just to spend a weekend with me, despite your hectic lab work, to sharing your insights as we navigated our PhD journeys together. To my little brother, *Mostafa*—your smile has the power to change my entire mood, and your encouraging messages light up my world. Thank you for always being there for me.

I would like to acknowledge the funding I received from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement PoLiMeR, No 812616, which made this work possible.

Contents

1 Introduction 1
1.1 Motivation . . . 2
1.2 Contributions . . . 4
1.3 Structure . . . 7

2 Background 9
2.1 Biomedical Natural Language Processing . . . 9
2.1.1 Biomedical Named Entity Recognition and Linking . . . 10
2.1.2 Biomedical Relation Extraction . . . 14
2.1.3 Joint Entity and Relation Extraction . . . 17
2.2 Biomedical Gold-Standard Datasets and Evaluation Metrics . . . 19
2.2.1 Named Entity Recognition Datasets . . . 19
2.2.2 Named Entity Recognition Evaluation Metrics . . . 21
2.2.3 Joint Named Entity and Relation Extraction Datasets . . . 24
2.2.4 Joint Named Entity and Relation Evaluation Metrics . . . 26
2.3 Transformer-based Language Models . . . 27
2.3.1 Language Models . . . 31
2.3.2 Fine-tuning . . . 38
2.4 Summary . . . 39

3 Weighted Loss Trainer 41
3.1 Biomedical Information Extraction . . . 43
3.2 Biomedical Pre-trained Language Models . . . 44
3.2.1 State-of-the-Art in BioNER . . . 46
3.2.2 Choice of Language Models . . . 54
3.2.3 Vanilla Fine-tuning Approaches . . . 56
3.2.4 Summary and Open Issues . . . 58
3.3 Class Imbalance . . . 58
3.3.1 State-of-the-Art in Class Imbalance . . . 60
3.3.2 Summary and Open Issues . . . 68

3.4 A Cost-sensitive Fine-tuning Approach . . . 69
3.4.1 WeLT Fine-tuning Approach . . . 70
3.4.2 WeLT's Application Example . . . 73
3.4.3 WeLT vs. Others . . . 74
3.4.4 Evaluating WeLT on BioNER . . . 75
3.4.5 Results and Discussion . . . 77
3.4.6 Error Analysis . . . 82
3.5 Study Limitations . . . 83
3.6 Impact of Recognised WeLT Entities on BioNEL . . . 84
3.6.1 Utilising WeLT-Recognised Entities for BioNEL . . . 85
3.6.2 Experiments Overview . . . 85
3.6.3 Results and Discussion . . . 86
3.6.4 Study Limitations . . . 88
3.7 Summary and Discussion . . . 88

4 Span-based Joint Named Entity and Relation Extraction Using WeLT 91
4.1 Related Work . . . 95
4.2 SpERT Approach . . . 98
4.2.1 Span Classification . . . 99
4.2.2 Span Filtering . . . 100
4.2.3 Relation Classification . . . 101
4.2.4 Negative Sampling Strategy . . . 102
4.2.5 Joint Loss for JNERE . . . 102
4.3 Cost-Sensitive SpERT Using WeLT . . . 103
4.3.1 WeLT Span Classifier . . . 103
4.3.2 WeLT Relation Classifier . . . 104
4.3.3 WeLT-SpERT's Dummy Example . . . 106
4.3.4 WeLT Joint Loss Functions . . . 109
4.4 Evaluating WeLT-SpERT . . . 110
4.4.1 Evaluation Results . . . 114
4.4.2 Error Analysis . . . 117
4.5 Summary and Discussion . . . 119

5 Attention Weight Mechanism JNERE Using WeLT 121
5.1 JNERE with Attention Weight Mechanism . . . 124
5.1.1 Span Classifier . . . 124
5.1.2 Span Filtering . . . 128

5.1.3 Relation Classification . . . 128
5.1.4 Negative Sampling Strategy . . . 129
5.1.5 ASpERT Loss Functions . . . 129
5.1.6 Impact of the Novel Span Classifier . . . 130
5.2 Cost-sensitive ASpERT using WeLT . . . 130
5.2.1 WeLT Span Classifier . . . 131
5.2.2 WeLT Relation Classifier . . . 133
5.2.3 WeLT Joint Loss Functions . . . 134
5.3 Evaluating WeLT-ASpERT . . . 135
5.3.1 Evaluation Results . . . 139
5.3.2 Error Analysis . . . 142
5.4 Summary and Discussion . . . 144

6 Table-filling JNERE Using WeLT 147
6.1 Related Work . . . 149
6.2 Table Labelling Using CNNs . . . 151
6.2.1 Table Representation . . . 152
6.2.2 Word Embeddings . . . 152
6.2.3 Prediction Model . . . 153
6.2.4 Training and Prediction . . . 153
6.3 Cost-sensitive TablERT-CNN Using WeLT . . . 154
6.3.1 WeLT Span Classifier . . . 157
6.3.2 WeLT Relation Classifier . . . 158
6.3.3 WeLT Joint Loss Functions . . . 159
6.4 Evaluating WeLT-TablERT-CNN . . . 160
6.4.1 Evaluation Results . . . 161
6.4.2 Error Analysis . . . 164
6.5 Summary and Discussion . . . 166

7 Conclusions and Future Work 169
7.1 Key Insights . . . 169
7.1.1 WeLT's Application . . . 170
7.1.2 Performance of WeLT-Based Models Against Comparable Models . . . 172
7.1.3 Final Assessment of WeLT Models . . . 179
7.2 Outlook . . . 180

Acronyms 183

Glossary 187

Bibliography 191

1 Introduction

Picture a bustling biomedical research lab where scientists are under pressure to complete a literature survey on a very rare disease named as Arboleda-Tham Syndrome (ARTHS), also known as "KAT6A Syndrome". A researcher has been tasked with sifting through a vast amount of scientific papers and clinical reports via PubMed,[1] as shown in Figure 1.1, to identify and extract critical information; such as the rare gene "KAT6A", novel treatment approaches, and specific patient outcomes. The goal is to find complementary studies that can validate or enhance their own groundbreaking research on this rare disease. Despite the use of advanced information extraction tools, the researcher often encounters significant frustration. These tools frequently miss vital details, especially when dealing with obscure entities like specific rare gene variants or unique case studies that are infrequently mentioned.

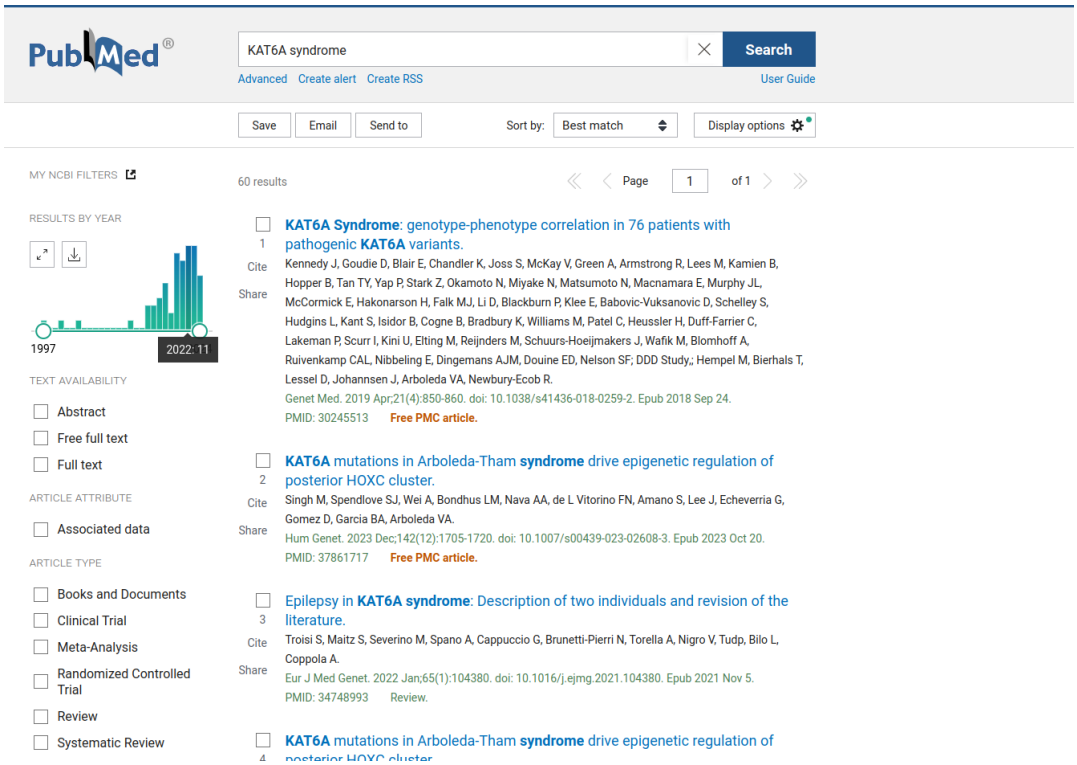

Figure 1.1: PubMed search results for the query "KAT6A Syndrome" show that the highest number of articles published in a single year is 11. The screenshot was taken on 28.08.2024.

[1] PubMed: https://pubmed.ncbi.nlm.nih.gov/, last accessed: 28.08.2024.

Now, put yourself in the position of a university hospital's medical data curator. New research is released every day, contributing to the extensive body of biomedical literature. Ensuring that these studies are appropriately annotated with relevant entities such as gene names, illnesses, and drug interactions is the curator's responsibility. However, this work is tedious due to the sheer number of data, particularly when dealing with uncommon diseases or therapies that are scattered over hundreds of documents (Tasci et al., 2022; Yang et al., 2020a). Common entities are usually prioritised over uncommon but important ones, which may be under-annotated or even overlooked by automated systems designed to help this process. Consequently, the resulting database may lack crucial connections, potentially impeding researchers who rely on this information to advance patient care.

Last but not least, consider a reporter looking into an unidentified fraud gang who hopes to find trends by contrasting the present case with earlier news stories. Information on key players, the sites of fraudulent activity, and any common strategies or linkages to the fraud gang must be gathered and examined by the journalist. The journalist sifts through a variety of government records, investigative papers, and newswire articles to find clues that could connect the current scam to previous instances. However, many fraudulent activities are difficult for current information extraction technologies to detect because they are obscure or have not received much attention (Wei et al., 2013; Tomar et al., 2021; Ahmed and Saini, 2023). As a result, the reporter must manually sort through a substantial number of stories, assembling disparate pieces of information to locate significant similarities that can help clarify the present fraud case.

Across all these scenarios, one issue is consistently apparent, whether it's a rare disease in biomedical research, a niche genetic disorder, or an obscure legal precedent, entities that appear less frequently in the data are often under-represented in the models designed to identify them (Song et al., 2021; Henning et al., 2023). This imbalance not only skews the results but also places a heavy burden on human annotators and curators, who must manually correct and complete the annotations, resulting in an incredibly time-consuming and labour-intensive process.

1.1 Motivation

What if there were a way to ease the burden of the class imbalance problem in information extraction? Imagine leveraging pre-annotations machine-generated predictions of entity and relation labels to assist human annotators in a semantic annotation[2] tool, as shown in Figure 1.2.

By providing an initial set of annotations specifically designed to account for class imbalance, these systems would enable curators and researchers to focus on refining and correcting the pre-annotations rather than starting from scratch.

[2]Semantic annotation tool INCEpTION: https://inception-project.github.io/, last accessed: 28.08.2024.

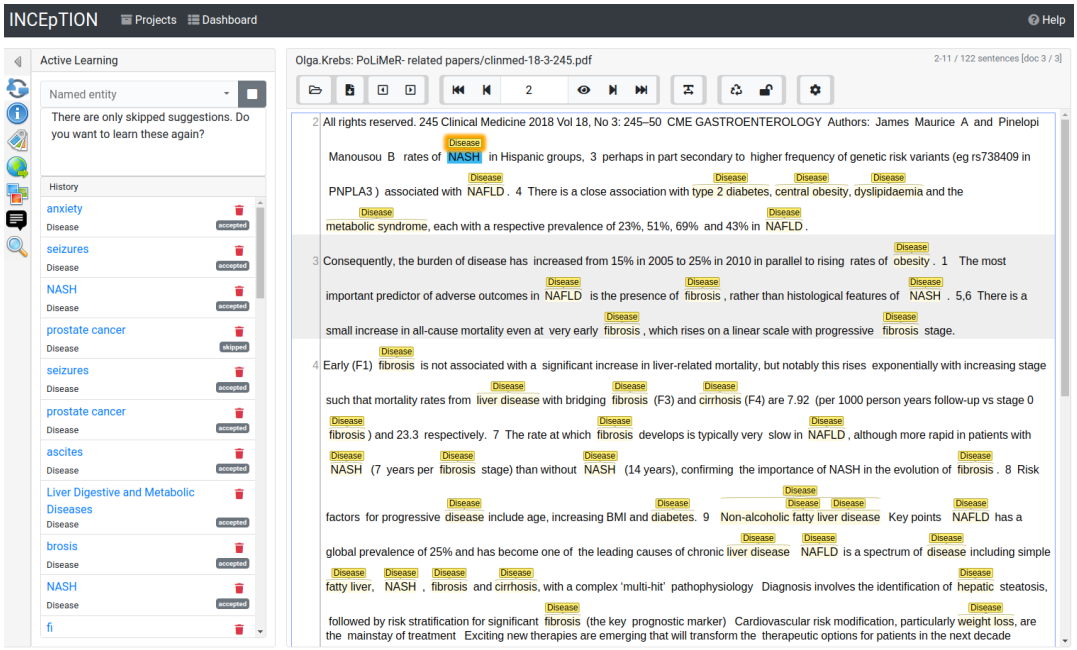

Figure 1.2: Pre-annotations generated by WeLT, as discussed in Section 3.4 *(highlighted in yellow)* and further elaborated on subsequent chapters, are reviewed by data curator Olga Krebs *(highlighted in blue)*. Using the Active Learning feature in the semantic annotation tool INCEpTION (Klie et al., 2018), the data curator can accept or reject these annotations.

This approach would not only speed up the annotation process but also help ensure that rare and crucial entities receive the attention they deserve, reducing the risk of important information being overlooked.

Consider again the scientist in biomedical research, the medical data curator at university hospital, and the legal journalist. Each of these professionals could benefit immensely from a more balanced and efficient information extraction system. By developing innovative methods to address class imbalance in Named Entity Recognition (NER) and Relation Extraction (RE) and by integrating pre-annotations to streamline the work of human annotators, we can create tools that enable data curators to focus on missed annotations, enhancing overall accuracy, reliability, and the ability to capture the full spectrum of essential information in these critical fields.

Despite the significant advancements in Deep Learning (DL) methods, particularly with pre-trained language models (PLMs) and the power of transfer learning for fine-tuning models to specific downstream tasks, standard fine-tuning approaches (Devlin et al., 2019; Liu et al., 2019b; Clark et al., 2020), also known as vanilla fine-tuning approaches assume that training data is balanced, meaning that all classes are represented equally. However, in real-world applications, especially in the biomedical domains (Dogan et al., 2014; Li et al., 2016; Krallinger et al., 2015; Smith et al., 2008; Luo et al., 2022a; Gerner et al., 2010; Gurulingappa et al., 2012) and newswire domains (Roth and Yih, 2004), datasets are often highly skewed, as discussed later in Section 2.2, which may

negatively impact overall prediction performance. Some attempts have been made to address the class imbalance problem through data-level approaches such as sampling or augmentation (Shi et al., 2022; Henning et al., 2023); yet studies show that PLMs, despite their power, tend to re-learn removed concepts (Lo et al., 2024).

This thesis focuses on exploring cost-sensitive learning solutions to address the problem of class imbalance by fairly adjusting class weights during fine-tuning. The objective is not to develop a specialised cost-sensitive approach tailored to a specific domain, but rather to create a method sufficiently generalisable to be adapted to various information extraction tasks. Unlike previous approaches (Lin et al., 2017; Cui et al., 2019), we avoid the need for manual hyper-parameter tuning. Instead, our cost-sensitive approach is data-driven, adjusting class weights based on the natural distribution of classes within the dataset. By utilising pre-annotations generated from this approach, we aim to enhance the accuracy and efficiency of information extraction systems, ultimately reducing the workload for human annotators and improving the comprehensiveness of biomedical databases. The outcomes of this research have the potential to significantly impact any domain where accurate and complete information extraction is crucial, whether in journalism, law, environmental science, or beyond.

The following section summarises our contributions in developing a cost-sensitive information extraction approach to mitigate class imbalance, and provides an overview of the thesis structure.

1.2 Contributions

We have already outlined several issues and challenges in the field of information extraction that are pertinent to this thesis. At this stage, we aim to provide a more structured summary of the contributions of this work and set clear expectations for readers:

1. We investigate a series of limitations in existing information extraction research, with a particular focus on biomedical named entity recognition and linking, span-based joint entity and relation extraction, and table-filling joint entity and relation extraction. Our findings highlight three main limitations. First, the widely adopted vanilla fine-tuning paradigm for language models (LMs) in downstream tasks (Radford et al., 2019; Devlin et al., 2019; Raffel et al., 2020) is far from ideal. Fine-tuning can predispose pre-trained models to over-fitting and issues with out-of-distribution (OOD) data, particularly due to the large model size and the relatively small size of domain-specific datasets (Zhao et al., 2019; Radiya-Dixit and Wang, 2020; Guo et al., 2021; Gordon et al., 2020; Zaken et al., 2022). In class imbalance scenarios, fine-tuned models are especially prone to over-fitting under-represented classes due to their limited representation (ValizadehAslani et al., 2022).

Secondly, OOD samples that are data not encountered during training (Zhang et al., 2021a), typically have a different distribution than the training data, leading to a distribution shift. These OOD samples are critical for testing the generalization ability of new approaches. Research by Kumar et al. (2022b) demonstrates that fine-tuning can distort pre-trained features, resulting in poor OOD accuracy (Kumar et al., 2022a).

Thirdly, even when employing specialized class-balanced techniques such as re-sampling, models still tend to overfit (He and Garcia, 2009; Buda et al., 2017; Horn and Perona, 2017).

2. Concerning class imbalance, existing research primarily focuses on three paradigms: data-level approaches, such as resampling and augmentation, algorithm-level re-weighting, which adjusts the loss function to account for class distribution, and hybrid approaches, combining both data-level and algorithmic methods. Data-level techniques may reduce generalisation performance on unseen data and increase memory and computational costs due to duplicated examples (Lee et al., 2022). Algorithm-level methods can lead to over-fitting, complex hyper-parameter tuning, and limited sensitivity to imbalance. Additionally, they often focus solely on inverse class frequency weighting, overlooking its broader impact on the dataset. Hybrid approaches inherit these issues while also adding complexity and computational demands. Motivated by the empirical findings and theoretical insights, we introduce **We**ighted **L**oss **T**rainer (WeLT), a novel adaptive loss function. WeLT addresses the limitations of previous weighting by adjusting class weights based on the complement of each class's relative frequency. Consequently, the majority classes (i.e., classes with a higher number of samples) receive less weight, while minority classes (i.e., classes with fewer samples) are assigned more weight.

3. We introduce a cost-sensitive fine-tuning approach based on WeLT. We investigate the impact of the WeLT cost-sensitive approach by fine-tuning eight biomedical gold-standard datasets for named entity recognition and linking tasks. Extensive evaluations are conducted against vanilla fine-tuning and other weighting schemes. Furthermore, we extend the application of the WeLT fine-tuning approach to complex nested named entities that exist within the boundaries of other entities, forming a hierarchical structure. In addition to overlapping named entities that share common spans, where parts of one entity can be part of another entity. Our empirical analysis demonstrates the benefits of addressing the class imbalance problem through WeLT.

4. We also explore span-based joint extraction models, which have shown significant advancements in both entity and relation extraction, such as SpERT (Eberts and Ulges, 2020). These models treat text spans as candidate entities and span pairs as candidate relationship tuples, achieving state-of-the-art results. However, span-based models face challenges, particularly

in managing a substantial number of non-entity spans and irrelevant span pairs during joint extraction tasks, which can significantly impair model performance. Beyond the inherent class imbalance in datasets, these models also encounter additional imbalance due to strong negative sampling (Xue and Lu, 2023). To address these limitations, we propose WeLT-SpERT, a WeLT span-based joint entity and relation extraction approach. We develop four WeLT-SpERT joint loss functions to tackle both class imbalance and strong negative sampling, comparing WeLT-SpERT with original span-based approaches.

5. In addition to the challenges posed by strong negative sampling in span-based extraction models, we have identified further limitations, particularly lack of boundary supervision and sole BERT encoding dependency. To address these issues, Jianquan Ouyang (2022) introduced the ASpERT model, which employs an attentional contribution degree algorithm combined with a multilayer perceptron and a softmax-based span classification framework. Although ASpERT offers improvements over traditional span-based models, it still relies on the strong negative sampling approach. Inspired by ASpERT's advancements, we propose WeLT-ASpERT, which incorporates three WeLT joint loss functions adapted from ASpERT's enhanced methodology. We evaluate WeLT-ASpERT's ability to mitigate class imbalance and reduce the impact of strong negative sampling, comparing its performance with that of the original ASpERT model.

6. In addition to span-based joint extraction models, table-filling approaches (i.e., matrix-like table to simultaneously identify both entities and their relations in a single framework) have impressive performance (Ma et al., 2020). Although, such approaches cannot handle nested overlapping entities relying on token-level tagging schemes. However, recently Ma et al. (2022) propose TablERT-CNN, a joint named entity and relation extraction model by stacking convolutional neural networks. TablERT-CNN outperforms traditional span-based models and does not adapt the strong negative sampling. However, TablERT-CNN relies on filling the entity labels based on conventional token-labelling schemes and directed relation labels. Thus, TablERT-CNN inherits the usual class imbalance problem. Hence, we present WeLT-TablERT-CNN, which incorporates three WeLT joint loss functions adapted from TablERT's enhanced methodology. We investigate the impact of addressing the class imbalance problem on the table-filling joint extraction model and compare it to the original model.

7. Finally, we evaluate the WeLT-based information extraction models alongside other weighting schemes, noting that they are smaller in size and have fewer training parameters compared to large language models (LLMs).

1.3 Structure

The remainder of this work is organised as follows. In Chapter 2, we begin with a formal introduction to the task of information extraction, with a particular emphasis on biomedical applications. This chapter also addresses the highly skewed gold-standard datasets fine-tuned for various downstream tasks and the different types of transformer-based language models. In Chapter 3, we examine the significant limitations of existing biomedical pre-trained language models and the shortcomings of conventional fine-tuning approaches. We discuss current paradigms aimed at addressing class imbalance and their associated limitations. To address these challenges, we propose the WeLT approach, a cost-sensitive fine-tuning method for biomedical named entity recognition. We compare WeLT with conventional and other existing weighting schemes and extend our investigation to assess the impact of WeLT-recognised entities on the subsequent task of entity linking. Building on the success of WeLT in these tasks, Chapter 4 explores its performance on overlapping (often nested) entities, adapting it for both biomedical and newswire applications. We also examine the adaptation of WeLT joint loss functions within span-based models, highlighting the limitations of strong negative sampling in these models. In Chapter 5, we identify further limitations in existing span-based approaches related to span classification and adapt WeLT to an enhanced span-based model that continues to employ strong negative sampling. Additionally, in Chapter 6 we explore an alternative joint extraction model that incorporates a table-filling approach and propose a WeLT-based table-filling model. Additionally, we conduct an extensive evaluation of all WeLT-based models developed in Chapters 4 to 6, comparing them to other pre-trained language models with a particular focus on large language models. Finally, in Chapter 7, we summarise the findings, discuss remaining open questions, and suggest potential directions for future research.

2 Background

Information Extraction (IE) is a fundamental process in the field of Natural Language Processing (NLP), aimed at automatically identifying structured information from unstructured or semi-structured text data (Hobbs, 2002). The extracted information typically includes predefined types of data, such as named entities (e.g., organizations, people, occupations), relationships between these entities, and other relevant events or facts (Jiang, 2012). IE serves as a foundational requirement for a wide range of downstream tasks, such as question answering, information retrieval, and content summarization. Typically, IE tasks consist of Named Entity Recognition (NER), Relation Extraction (RE), and Event Extraction (EE).

In this thesis, we focus on NER and RE, particularly in the context of biomedical applications. However, we also explore a general domain application, identifying entities and their corresponding relations from newspaper articles. This chapter presents several foundational concepts and task contexts essential to this thesis, particularly for extracting information from biomedical documents.

Structure. Section 2.1 introduces basic concepts of biomedical natural language processing and information extraction. We present biomedical named entity recognition, and normalization. We highlight the biomedical relation extraction task and its distinctive characteristics. In addition, we review various paradigms for joint entity and relation extraction. Section 2.2 discusses the gold-standard biomedical datasets and evaluation metrics, focusing on named entity and joint entity-relation extraction datasets. Section 2.3 reviews different transformer-based language models, with an emphasis on recent large language models (LLMs) and fine-tuning approaches. Finally, in Section 2.4, we provide a summary and discussion.

2.1 Biomedical Natural Language Processing

Biomedical Natural Language Processing (BioNLP) refers to the application of NLP techniques to biomedical texts, often from scientific literature. BioNLP aims to extract, understand, and analyse information from unstructured texts, which is critical for tasks in biomedical research and healthcare management. BioNLP methods include techniques such as Biomedical Named Entity Recognition (BioNER) and Biomedical Relation Extraction (BioRE). BioNER identifies and classifies entities like chemicals and diseases, while BioRE focuses on relationships, such as drug-disease associations or protein-protein interactions (Perera et al., 2020).

BioNLP shared tasks bring together experts to develop and evaluate algorithms for various biomedical text mining challenges, providing gold-standard datasets and evaluation metrics to foster collaboration and competition. Examples include the following:

- BioCreative is a series of challenges focusing on information extraction and text mining in biological and biomedical domains. BioCreative has facilitated advancements in biomedical annotation, entity normalization, relation extraction, and text classification (Smith et al., 2008). BioCreative datasets cover biomedical entities (e.g., genes, proteins, diseases, and chemicals) contributing to tools and resources in the biomedical text mining community. BioCreative I addressed tasks such as gene mention tagging and gene normalization (Kinoshita et al., 2005; Yeh et al., 2005). BioCreative II-VII expanded the tasks to include protein-protein interaction extraction, chemical-disease relation extraction, and COVID-19 literature analysis (Krallinger et al., 2008; Leaman et al., 2023).

- BioNLP Shared Tasks (BioNLP-ST) are organized by the University of Tokyo in 2009, BioNLP-ST primarily focuses on tasks like BioNER, BioRE, event extraction, and text classification. Participants propose methods to extract structured information from biomedical literature using annotated corpora and evaluation metrics (Kim and Pyysalo, 2013). BioNLP Shared Task 2009 focused on event extraction and gene expression events (Kim et al., 2009). BioNLP Shared Tasks 2021-2023 addressed challenges like medical video question answering and summarization of biomedical research articles (Goldsack et al., 2023; Gupta and Demner-Fushman, 2022).

BioNLP techniques and shared tasks collectively contribute to the development of advanced models and benchmarks for the extraction of structured information from unstructured biomedical texts. Through initiatives like BioCreative and BioNLP-ST, the research community continues to push the boundaries of biomedical text mining and information extraction.

2.1.1 Biomedical Named Entity Recognition and Linking

BioNER is one of the tasks in BioNLP that focuses on recognising and classifying specific entities or concepts within biomedical textual data. These various entities typically include genes, proteins, diseases, drugs, species, and other relevant terms. BioNER's primary goal is to automatically extract and classify these entities from unstructured biomedical text data, such as scientific articles, clinical notes, or biomedical patents. By identifying and categorising these entities, BioNER facilitates various downstream biomedical applications including relation extraction, literature mining, information retrieval, and clinical decision support. BioNER systems typically employ machine and deep learning and NLP techniques to recognise and classify biomedical entities. These

techniques may involve the use of annotated datasets for training supervised learning models, which learn to predict the entity type of each token in a given text.
Figure 2.1 depicts a simple example of recognising and classifying chemical and disease entities (Eriksson and Saldeen, 1989). This snippet is part of BioCreative V Chemical Disease Relation corpus (BC5CDR)'s training data.

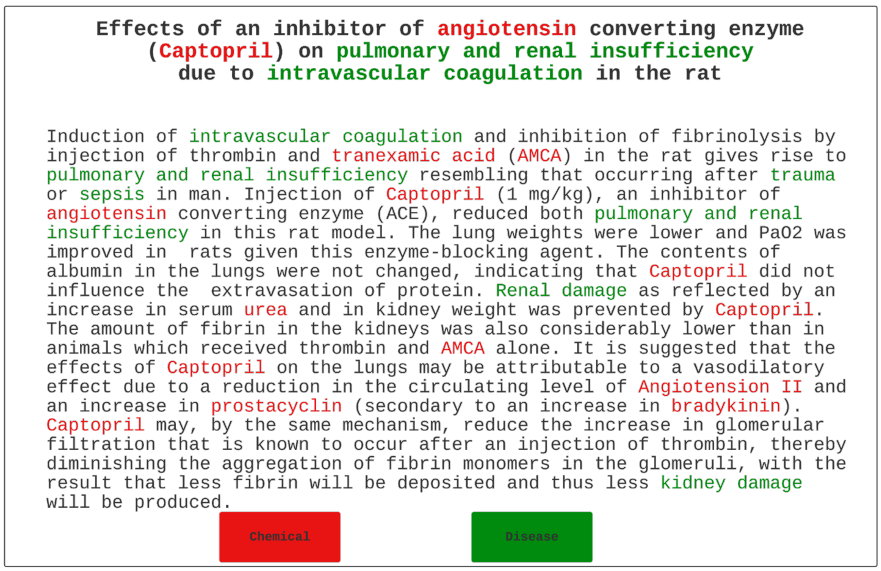

Figure 2.1: A simple chemical and disease BioNER example. This is part of the training data of BC5CDR for chemical-disease relationships (Li et al., 2016).

BioNER can be mathematically formulated as follows:
Let d represent a document consisting of n tokens, $\mathcal{T} := [t_1, t_2, \ldots, t_n]$, where each token t_i corresponds to a subword unit, word, or punctuation mark, depending on the tokenisation strategy as later discussed in Section 2.3. Each token t_i belongs to a predefined set of biomedical entity types or categories $\mathcal{E} := \{e_1, e_2, \ldots, e_c\}$, where c denotes the total number of entity types.
The objective of BioNER is to assign an entity type e_i to each token t_i in d based on its semantic meaning and context within the document.[1] In other words, this is a sequence labelling problem, where each token is labelled with its corresponding entity type. Due to the nature of biomedical text, which often contains overlapping entities (frequently nested ones) (Wang et al., 2022), we distinguish between different categories of named entities as follows:

- **Single-label entities**: each token is assigned a single category from the predefined set of categories $\mathcal{E}$. This is also referred to as flat named entities. For instance, as illustrated in Figure 2.1, the token "angiotensin" is labelled as "chemical", and the tokens "intravascular coagulation" are labelled as "disease".

[1]The glossary provides the notations used throughout the thesis.

- **Multi-label entities**: a token may belong to multiple categories simultaneously. This occurs when a token represents multiple concepts or entities in the same context. For example, in a document discussing drug interactions, the token "aspirin" might be labelled with both "drug" and "treatment". This is valid since aspirin is commonly used as a treatment for conditions like headaches, while also being classified as a "drug" due to its pharmacological properties.

- **Nested entities**: these occur when one entity is contained within another. Nested entities are hierarchically structured and involve overlapping spans, where the contained entity shares part of its text span with the containing entity. For instance, Figure 2.2 shows nested entities (Lee et al., 1995): "B2 subunit promoter region", where "B2 subunit" is tagged as "protein", "promoter region" is tagged as "DNA", and "B2 subunit promoter region" is also tagged as "DNA".

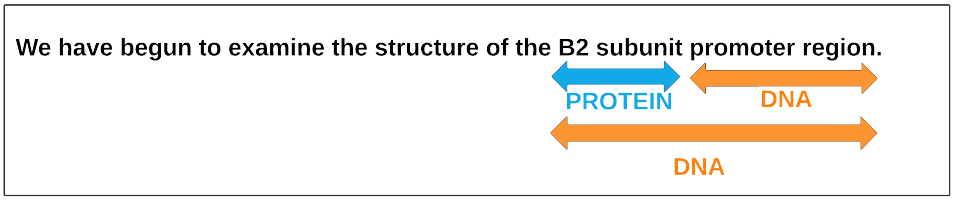

Figure 2.2: A simple example of nested named entities.

- **Overlapping entities**: these occur when two named entities overlap without one being completely contained within the other. For example, in the sentence, "President Barack Obama visited the White House", "President" is a position entity, "Barack Obama" is a person entity and "White House" is a location entity. There exist an overlap between the position "President" and the person "Barack Obama" without one being fully contained within the other. Thus, the overlap exists without a nested structure.

The objective of nested entity recognition (NestedNER) is to assign an entity type e_i to each token t_i in d based on its semantic meaning and context within the document while handling nested structures. Let $\mathcal{A}$ represent the set of all possible annotations in the document d, where each annotation $a \in \mathcal{A}$ is represented as a tuple $(start_a, end_a, type_a)$ indicating the start and end positions of the annotation and its corresponding entity type $type_a \in \mathcal{E}$. The goal of NestedNER is to find the optimal set of annotations $\mathcal{A}$ that maximizes some objective function $f(A)$ subject to the constraint that no two annotations in $\mathcal{A}$ can overlap or be nested within each other and that continuous text spans are restricted by a length threshold ε.

NestedNER can be mathematically formulated as follows:

$$\begin{aligned} &\max_{\mathcal{A}} f(\mathcal{A}) \quad \text{subject to:} \\ &\quad \forall a_1, a_2 \in \mathcal{A}, \quad (a_1 \text{ is nested in } a_2 \text{ or vice versa}), \\ &\quad \forall a \in \mathcal{A}, \quad \text{length}(a) \leq \varepsilon, \end{aligned} \tag{2.1}$$

where the condition a_1 is nested in a_2 or vice versa is defined as:

$$(\text{start}_{a_2} \leq \text{start}_{a_1} \text{ and } \text{end}_{a_1} \leq \text{end}_{a_2}) \quad \text{or} \quad (\text{start}_{a_1} \leq \text{start}_{a_2} \text{ and } \text{end}_{a_2} \leq \text{end}_{a_1}).$$

The overlapping entities (OverlapNER) are constrained such that the annotations can share common text spans without one annotation being fully contained within another. OverlapNER can be mathematically formulated as:

$$\begin{aligned} &\max_{\mathcal{A}} f(\mathcal{A}) \quad \text{subject to:} \\ &\quad \forall a_1, a_2 \in \mathcal{A}, \quad a_1 \text{ overlaps with } a_2, \\ &\quad \forall a \in \mathcal{A}, \quad \text{length}(a) \leq \varepsilon, \end{aligned} \tag{2.2}$$

where the condition a_1 overlaps with a_2 is defined as:

$$\text{start}_{a_1} < \text{end}_{a_2} \quad \text{and} \quad \text{start}_{a_2} < \text{end}_{a_1},$$

and neither is nested within the other:

$$\neg(\text{start}_{a_2} \leq \text{start}_{a_1} \quad \text{and} \quad \text{end}_{a_1} \leq \text{end}_{a_2}) \quad \text{and} \quad \neg(\text{start}_{a_1} \leq \text{start}_{a_2} \quad \text{and} \quad \text{end}_{a_2} \leq \text{end}_{a_1}).$$

In summary, the difference between NestedNER and OverlapNER lies in how annotations are spatially related, both constrained by the text span length threshold ε:

- nested entities refer to cases where one annotation is fully contained within another.
- overlapping entities occur when annotations share parts of their spans but are not fully contained within one another.

Biomedical named entity linking (BioNEL) maps named entities to standardised identifiers or concepts (Ruas et al., 2020). Particularly, BioNEL involves linking named entities mentioned in biomedical texts to relevant concepts or entities in knowledge bases or ontologies.
BioNEL is also known as entity normalisation and this enhances interoperability and facilitates the integration of information from different biomedical resources. There are various biomedical knowledge bases, for instance, the following:

- Comparative Toxicogenomics Database (CTD Taxonomy): organises chemicals, genes, diseases, and species based on their associations with toxicological and pharmacological effects (Wiegers et al., 2009).
- Medical Subject Headings (MeSH): a controlled vocabulary and ontology developed by the National Library of Medicine (NLM) for indexing biomedical information. MeSH encompasses hierarchically structured concepts related to diseases, chemicals, anatomy, drugs, and medical procedures (Lipscomb, 2000).
- Online Mendelian Inheritance in Man (OMIM): a database that contains human genes and genetic disorders. It is a widely used resource in genomics and medical genetics research. Thus, OMIM provides valuable insights into the molecular mechanisms underlying inherited disorders (Hamosh et al., 2002).
- Medical Dictionary for Regulatory Activities (MEDIC): is an enriched resource as a result of merging CTD taxonomy with OMIM terms, synonyms, and identifiers (Davis et al., 2012). MEDIC allows practitioners to explore associated CTD data at different levels for meta-analysis. MEDIC is updated monthly, and the latest version can only be downloaded.
- Unified Medical Language System Semantic Network (UMLS): is a network of concepts and relationships that provide a semantic framework for linking and integrating biomedical terminologies and ontologies. UMLS encompasses various biomedical semantic types such as diseases, chemicals, genes, and anatomical structures along with relationships between them (Bodenreider, 2004).

2.1.2 Biomedical Relation Extraction

Biomedical Relation Extraction (BioRE) focuses on identifying and extracting relationships between various types of entities mentioned in textual data. As discussed in Section 2.1.1, entities typically include genes, proteins, chemicals, diseases, drugs, and other biomedical concepts. The goal of BioRE is to automatically identify and extract meaningful associations or interactions between these entities from unstructured text data, such as scientific articles or biomedical literature. This process helps in uncovering new associations between genes, proteins, and diseases, contributing to biological understanding and discovery. Additionally, BioRE provides clinicians with relevant information about potential interactions between genes, diseases, and treatments, supporting clinical decision-making.
BioRE can be mathematically formulated as follows:
Let $\mathcal{R}$ represent a set of relations between entities in the document d. Each relation $r_j \in \mathcal{R}$ can be defined as a tuple (e_h, e_t, r_i), where e_h and $e_t \in \mathcal{E}$ are the identified head and tail entities, respectively in d, and r_i is the relation between them.

The objective of BioRE is to extract relations between entities in the document d based on their semantic meaning and context. In other words, BioRE aims to identify and classify relationships between pairs of entities within the document. This task can be represented as a classification problem, where the goal is to predict the relation label r_i for each pair of entities (e_h, e_t) in d.
For example, consider the sentence: "Having at least one APOE e4 gene doubles or triples the risk of getting Alzheimer's disease". In this case, BioRE would automatically identify that:

- "APOE e4" is a gene and "Alzheimer's" is a disease.
- The relationship is gene-related disease, meaning that patients with at least one "APOE e4" gene are more likely to develop "Alzheimer's disease".

This is a simple example, but with the rapid growth of biomedical literature, developing robust BioRE frameworks is crucial. Overall, BioRE plays a significant role in leveraging the vast amount of biomedical text available to extract valuable knowledge and insights for biomedical research and healthcare applications.
In general, relationships in BioRE can be binary or n-ary. Specifically, BioRE can involve either binary or n-ary relationships, depending on the complexity of interactions among entities. N-ary biomedical relationships involve interactions between three or more entities, often representing more complex biological phenomena. For example, metabolic pathways include multiple enzymatic reactions that convert substrates into products. The pathway "glycolysis" involves several enzymatic reactions that convert glucose into pyruvate.
Furthermore, the direction of relation types can be symmetrical or asymmetrical. In symmetrical relations, the relationship between entities remains the same regardless of their order. For instance, in protein-protein interaction (PPI) networks, if protein A interacts with "protein B", it's likely that "protein B" also interacts with "protein A". Hence, PPI relations are typically symmetrical, and the same relation label can be assigned to pairs of entities regardless of their order. In contrast, asymmetrical relations depend on the direction of the interaction, making the order of entities significant.

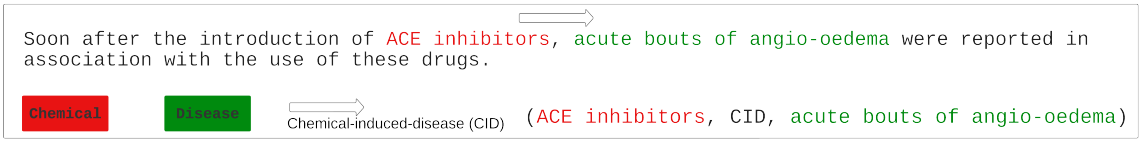

Figure 2.3: An asymmetrical chemical-induced-disease relation example (Li et al., 2016). This snippet is part of BC5CDR training data.

Figure 2.3 illustrates an asymmetrical chemical-induced-disease relationship, where the head entity is a chemical and the tail entity is a disease.
There are two types of relation extraction models as follows:

- **Intra-sentence relationships**: these involve connections between entities or concepts within the same sentence. Such relationships may include subject-verb relationships, noun-modifier relationships, or other grammatical and semantic connections. For example, in the sentence "Mutation in the BRCA1 gene is associated with an increased risk of breast cancer", an intra-sentence relationship is identified between the gene "BRCA1" and the disease "breast cancer".
- **Inter-sentence relationships**: these relationships often require understanding the context and content of multiple sentences to establish connections between entities. They may involve references or mentions of the same entity in different parts of the text. For example, consider the following sentences: Sentence (1) "Patient A is male and was diagnosed with diabetes. " Sentence (2) "He started taking insulin injections. " The inter-sentence relationship connects "Patient A" from Sentence (1) with "He" from Sentence (2), indicating that "Patient A" is the person who started taking insulin injections.

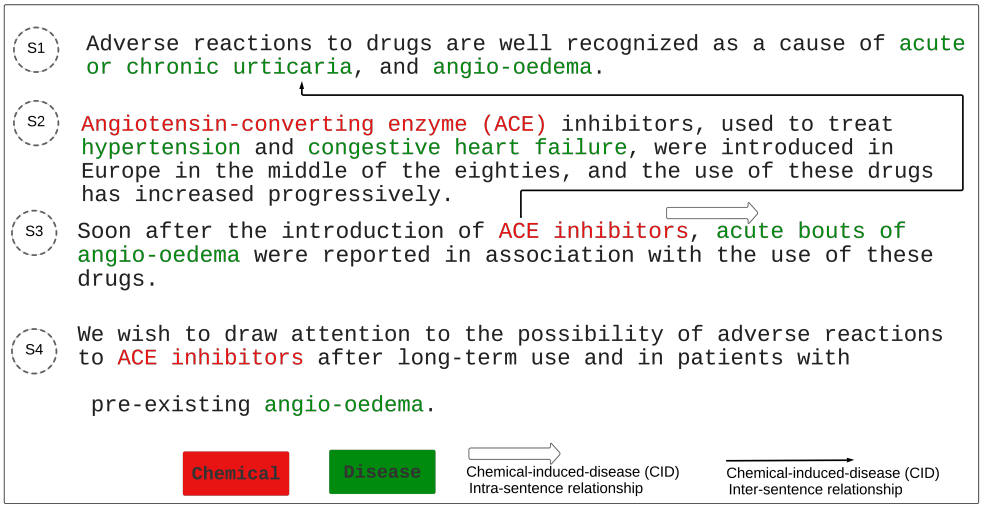

Figure 2.4: A simple inter-sentence and intra-sentence relation example. This snippet is part of BC5CDR for chemical-induced-disease relationships.

Figure 2.4 provides a simple example of inter-sentence and intra-sentence chemical-induced-disease (CID) relations. This example is part of the BC5CDR training data (KOZEL et al., 1995). The figure include the following:

- four sentences, one chemical entity "Angiotensin-converting enzyme (ACE)", and four diseases: "acute or chronic urticaria", "angio-oedema", "hyperparameterspertension", and "congestive heart failure".
- sentence (3) (i.e., denoted by S3 as shown in the figure) illustrates an intra-sentence relationship between "ACE inhibitors" and "acute bouts of angio-oedema".

To infer the inter-sentence relationship between "angiotensin-converting enzyme inhibitor (ACE)" and "urticaria", consider Sentence (1), which suggests that adverse drug reactions cause urticaria and angioedema. According to Sentence (3), the use of ACE inhibitors can cause acute angioedema. Therefore, it can be inferred that ACE inhibitors are also a cause of urticaria. In BC5CDR, there is a CID relationship between "angiotensin-converting enzyme inhibitor (ACE)" and "urticaria". Inter-sentence relation extraction requires logical inference across multiple sentences, which is beyond the scope of sentence-level relation extraction models.
Most existing benchmark BioRE datasets primarily focus on binary relationship types at the sentence level, with a few exceptions. According to Qian et al. (2022), only BC5CDR and the Gene Disease Associations (GDA) datasets include some inter-sentence relationships (Wu et al., 2019). BC5CDR contains approximately 27.4 % of its training data as inter-sentence relationships, while GDA has around 16.3 %. Due to the limited availability of gold-standard datasets, this thesis focuses on binary asymmetric sentence-level relations, excluding n-ary and inter-sentence relations.

2.1.3 Joint Entity and Relation Extraction

Initially, NER and RE were treated as separate tasks. In this approach, entity pairs are extracted first and then fed into the relation extraction model. Consequently, any errors in entity classification can lead to incorrect or missed relation extraction. To address these limitations, various joint entity and relation extraction (JNERE) models have been proposed. JNERE aims to simultaneously extract entities and their corresponding semantic relations, thereby enhancing information interaction between both tasks and reducing the high dependency of RE on the results of NER.
Figure 2.5 illustrates that JNERE can be categorized into three main paradigms:

- **Tagging and span classification**: the token tagging strategy treats joint extraction as a sequence labelling task. Each token is labelled according to schemes such as IOB or its variant BILOU. For details on these tagging schemes, refer to Sections 2.2.1 and 2.2.3. These models generate fixed-size semantic representations for token-level labels and use them to create relation semantic representations (Katiyar and Cardie, 2017; Ye et al., 2019; Bekoulis et al., 2018). For instance, Zhao et al. (2020) proposed a deep neural architecture that captures fine-grained token-level interactions. However, this approach is limited by the inability to handle nested or overlapping entities due to its token tagging strategy. Span-based approaches were introduced to address this issue by performing a detailed search on all possible spans. Other studies focus on span classification to support JNERE, predicting entities based on all possible enumerated spans and filtering named entities accordingly (Lai et al., 2021; Wan et al., 2021). Luan et al. (2018) introduced a multitask learning framework that predicts entity types from all possible named entity spans and then extracts relations from these recognized spans. The Dynamic Graph IE (DyGIE) model by Luan et al.

(2019) extends this by adding graph propagation to capture interactions between different spans. An enhanced version, DYGIE++, was proposed by Wadden et al. (2019), replacing the bidirectional long short-term memory (LSTM) encoder with BERT. Simpler models that omit graph propagation are discussed in Chapters 4 and 5.

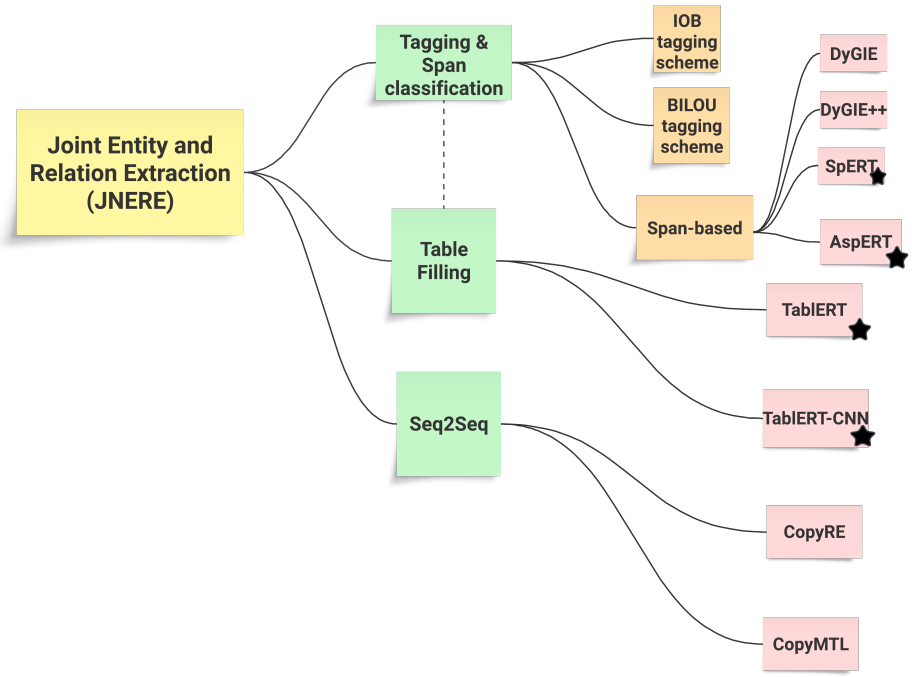

Figure 2.5: A hierarchical representation of JNERE state-of-the-art (SOTA) approaches that address the class imbalance problem. The ⋆ denotes models comparable to the proposed JNERE-WeLT models discussed in Chapters 4– 6.

- **Table-filling**: this approach frames NER and RE as a table-filling problem. A two-dimensional (2D table) is constructed where each entry captures the interaction between two entities within a sentence. NER is treated as a sequence labelling problem and assigned diagonally in the table using one of the tagging schemes mentioned earlier. Relation labels are placed in the off-diagonal entries (Miwa and Sasaki, 2014). A drawback of this method is its reliance on a single encoder for both tasks, limiting its ability to fully exploit the table structure. To address this, Wang and Lu (2020) designed separate encoders for entities and relations, using pairwise self-attention weights by BERT to capture word-word interactions for the relation encoder. Further variants of table-filling approaches are discussed in Chapter 6.

- **Sequence-to-Sequence (Seq2Seq)**: this model retains sentence features from unstructured text as input and decodes the entity-relation triples sequentially. The Seq2Seq approach mimics the human annotation process, where annotators first read the sentence, infer

semantic meaning, and then sequentially highlight entity-relation pairs (Sutskever et al., 2014). One powerful Seq2Seq baseline is CopyRE, which uses a copy mechanism in the decoder to avoid out-of-vocabulary issues (Gu et al., 2016). However, CopyRE has the following drawbacks: entity copying can be unstable due to reliance on an unnatural mask to differentiate between head and tail entities, and it struggles with multi-token entities as the copy-based decoder points only to the last token. To address these issues, Zeng et al. (2020) proposed CopyMTL, a multi-task learning model with an improved architecture for entity copying that adds a sequence labelling task to the CopyRE encoder.

2.2 Biomedical Gold-Standard Datasets and Evaluation Metrics

Biomedical gold-standard datasets for BioNER are crucial for training and evaluation. These datasets consist of annotated biomedical texts where named entities such as genes, proteins, diseases, and chemicals are labelled. The datasets are annotated by domain experts and serve as benchmarks for evaluating the performance of BioNER systems. Evaluation metrics are used to assess the performance of BioNER systems on gold-standard datasets. Common evaluation metrics for BioNER include precision, recall, and F1-score.
Section 2.2.1 presents the BioNER gold-standard datasets, standard tagging scheme, their statistics, and imbalance ratio. In Section 2.2.2, we highlight different evaluation scripts for both the BioNER and the BioNEL. Section 2.2.3 highlights the JNERE datasets, tagging scheme and their statistics. Finally, we present evaluation scripts tailored to JNERE in section 2.2.4.

2.2.1 Named Entity Recognition Datasets

Annotators and curators in the biomedical domain play a crucial role in advancing research in biomedical text mining tasks, especially BioNER. Domain experts annotate and curate biomedical texts, marking entities like diseases, chemicals, genes, proteins, and species. This text annotation is primarily a manual process that involves reading through texts, identifying relevant entities, and labelling them accordingly. This process can be tedious and time-consuming, especially when dealing with large volumes of text or complex entity types. These annotations serve as gold standards for training and evaluating BioNER models. Thus, domain experts follow standardized annotation guidelines. These guidelines ensure consistency across annotations, facilitating the development of reliable BioNER models and are essential for training models that generalize well to unseen data.
The commonly used tagging scheme for BioNER is the Inside–outside–beginning (IOB) format (Shen and Sarkar, 2005). It consists of three classes as follows:

- the **B** tag represents the beginning or first token of a biomedical entity.
- the **I** tag denotes the continuation of the first token as an inside biomedical entity.

- the **O** tag represents a token that is not part of a biomedical entity.

Thus **B** and **I** classes are the positive samples while the **O** class is the negative sample. Figure 2.6 depicts an IOB example for disease entities. In addition, these entities are linked to their corresponding MeSH IDs.

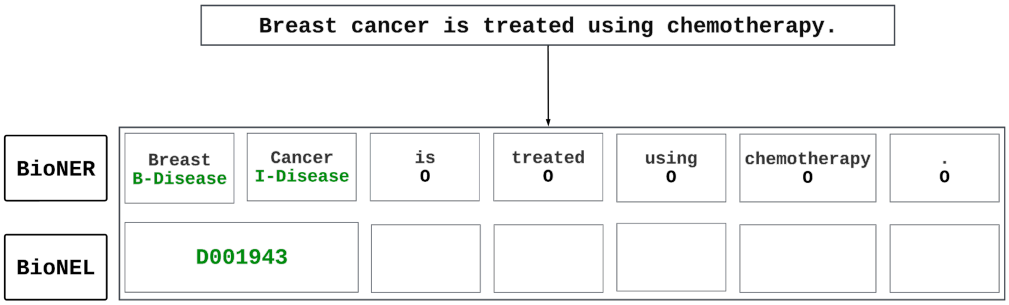

Figure 2.6: A simple disease BioNER and BioNEL example using IOB tagging scheme and corresponding MeSH IDs, respectively.

We provide the detailed description, statistics, and class frequencies of the used gold-standard BioNER datasets as follows:

- NCBI Disease: the American National Institutes of health released the National Centre for biotechnology information (NCBI) disease corpus to promote disease NER research (Dogan et al., 2014). The public release of the NCBI disease corpus contains 6,892 disease mentions, which are mapped to 790 unique disease concepts.
- BC5CDR-Disease and BC5CDR-Chemical: the BioCreative V chemical disease relation (CDR) corpus was created for the Chemical Disease Relation (CDR) Task (Li et al., 2016). It consists of human annotations of all chemicals, diseases, and their interactions in 1,500 PubMed articles.
- BC4CHEMD: the BioCreative IV chemical and drug (BC4CHEMD) named entity recognition task corpus (Krallinger et al., 2015). It contains 10,000 abstracts annotated for mentions of chemical and drug names.
- BC2GM: the BioCreative II gene mention task corpus (Smith et al., 2008). BC2GM consists of 20,000 sentences from biomedical publication abstracts, annotated genes, and proteins.
- BioRED: the biomedical relation extraction dataset (BioRED) corpus was created for multiple biomedical relations. BioRED consists of human annotations of all different biomedical entities and their interactions in 600 PubMed abstracts (Luo et al., 2022a). In this thesis, we only focus on BioRED-Disease and BioRED-Chemical for chemical and disease instances respectively.

- Linnaeus: this corpus has 153 PubMed full-text documents for 4,077 species annotations (Gerner et al., 2010).

Table 2.1 presents the statistics of each dataset including the number of sentences for training, development, and test data.

Dataset	num_training	num_validation	num_test
NCBI (Dogan et al., 2014)	5,433	924	941
BC5CDR-Disease (Li et al., 2016)	4,561	4,582	4,798
BC5CDR-Chemical (Li et al., 2016)	4,561	4,582	4,798
BC4CHEMD (Krallinger et al., 2015)	30,683	30,640	26,365
BC2GM (Smith et al., 2008)	12,575	2520	5,039
BioRED-Chemical (Luo et al., 2022a)	4,432	1,140	1,108
BioRED-Disease (Luo et al., 2022a)	4,432	1,140	1,108
Linnaeus (Gerner et al., 2010)	11,936	4,079	7,143

Table 2.1: Number of sentences in biomedical ground-truth datasets for training, development, and test data.

Table 2.2 shows the imbalanced nature of BioNER corpora for multiple entity types, including chemical, disease, gene, and species entities. Based on the statistics in Table 2.2, the biomedical

Dataset	O	B	I
NCBI (Dogan et al., 2014)	74.44	12.67	12.89
BC5CDR-Disease (Li et al., 2016)	93.99	3.54	2.47
BC5CDR-Chemical (Li et al., 2016)	93.99	4.40	1.61
BC4CHEMD (Krallinger et al., 2015)	92.69	3.30	4.01
BC2GM (Smith et al., 2008)	89.50	4.28	6.22
BioRED-Chemical (Luo et al., 2022a)	96.72	2.34	0.94
BioRED-Disease (Luo et al., 2022a)	94.78	3.00	2.22
Linnaeus (Gerner et al., 2010)	98.84	0.75	0.41

Table 2.2: Class distribution percentage for biomedical ground-truth training datasets.

ground-truth training datasets are highly skewed. With such high class imbalance, BioNER may be biased towards the **O** class, thus, they often misclassify entities (B and I classes).

2.2.2 Named Entity Recognition Evaluation Metrics

BioNER evaluation involves the assessment of identifying and classifying biomedical named entities in text data. As a supervised learning approach, the predicted entities are compared against the gold-standard datasets, also referred to as the ground truth. Thus, in the context of NER, the metrics of precision, recall, and F1 score are used to evaluate how well an NER system can identify and classify named entities in a given text.

Originally, the concepts of precision and recall are generally attributed to Cyril W. Cleverdon, who led the Cranfield experiments in the 1960s, one of the earliest systematic studies in information retrieval (Cleverdon, 1997). The F1 score is a combination of precision and recall into a single metric, which is the harmonic mean of precision and recall. The idea behind the F1 score was to provide a balanced measure when precision and recall are of equal importance. The F1 score and related F-measures were first formally introduced by David Lewis and others in the context of text classification and information retrieval (Lewis, 1995).

Biomedical predicted labels can be evaluated at both the entity-level and token-level to assess the performance of NER systems in identifying biomedical named entities. Entity-level metrics evaluate the performance at the level of entire entities. Thus, entity-level metrics provide an assessment of BioNER's prediction model to correctly identify complete entities in the text, and partial matches are not considered. In other words, this means that the model's predictions are evaluated based on whether they correctly identify entire entities as a whole. In contrast, token-level metrics assess the performance at the level of individual tokens allowing partial matches. Consequently, the model's predictions are evaluated based on whether each token in the sequence is correctly predicted as part of an entity. If there is any overlap between predicted and true entities, they are considered a match.

In this thesis, to maintain consistency with previous studies that use micro-averaging evaluation, we adopt the entity-level evaluation approach. A high entity-level precision reveals that the model accurately identifies and classifies named entities with minimal false positives. Entity-level recall computes the proportion of correctly identified entities out of all true entities in the dataset. Thus, recall captures the model's ability to identify all relevant entities in the text including those that may be missed by the model. Finally, the entity-level F1 score is the harmonic mean of entity-level precision and recall.

We present the mathematical equations for the entity-level micro-averaging evaluation metrics used in BioNER as follows:

- In micro-averaged precision P_{micro}, the true positives (TP), false positives (FP), and false negatives (FN) across all entity classes are aggregated before computing precision. P_{micro} is mathematically defined as:

$$P_{micro} := \frac{\sum_{i=1}^{c} TP_i}{\sum_{i=1}^{c}(TP_i + FP_i)} \tag{2.3}$$

  where:

  – TP_i is the number of true positive entities for class i,

  – FP_i is the number of false positive entities for class i, and

  – c is the total number of entity classes.

- In micro-averaged recall R_{micro}, the true positives (TP), and false negatives (FN) across all entity classes are aggregated before computing recall. R_{micro} is mathematically defined as:

$$R_{micro} := \frac{\sum_{i=1}^{c} TP_i}{\sum_{i=1}^{c}(TP_i + FN_i)} \tag{2.4}$$

  where FN_i is the number of false negative entities for class i.

- The micro-averaged F1 score $F1_{micro}$ is the harmonic mean of micro-averaged precision and micro-averaged recall. $F1_{micro}$ is mathematically defined as:

$$F1_{micro} := 2 \times \frac{P_{micro} \times R_{micro}}{P_{micro} + R_{micro}} \tag{2.5}$$

There are various tools and libraries available for NER evaluation, including seqeval (Nakayama, 2018), which provides functions for calculating precision, recall, and F1 scores for NER tasks based on gold standard annotations. seqeval has two modes: default and strict. In the default mode, correct entity labels require an exact boundary match over the surface string, regardless of the type, as shown in Example 2.1.

Example 2.1. *(Example of seqeval default evaluation)*
Given a disease entity, "Breast Cancer". The ground-truth labels are as follows: Breast is tagged as "B-Disease" and Cancer is labelled as "I-Disease". If the predicted labels are "B-Disease" for Breast and "B-Disease" for Cancer, it will be a true positive for both cases using seqeval's default mode.

Lee et al. (2020b) assessed BioBERT for BioNER based on entity-level exact matches using seqeval which outputs micro-averaged F1 score. For a fair comparison, we follow BioBERT's evaluation. However, we also use additional evaluation scripts:

- The FairEval is one of the latest metrics on which Ortmann argues that the traditional evaluation metric causes double penalties for close-to-correct annotations (Ortmann, 2022). Therefore, Ortmann developed FairEval, which ensures that every error is counted once. FairEval also provides more fine-grained metrics for error analysis, as it outputs true positives and separates boundary errors from false positives and false negatives.

- We use the BioCreative VII NLM track's official evaluation script[2] on the experiments that tested the impact of WeLT's recognized entities on biomedical entity linking (Leaman et al., 2023). This script measures the precision, recall, and F-score measures in a strict and approximate evaluation setting. The strict mode requires that the start and end offsets match

[2]BioCreative VII's evaluation script: https://ftp.ncbi.nlm.nih.gov/pub/lu/BC7-NLM-Chem-track/BC7T2-evaluation_v3.zip, last accessed: 01.08.2024.

exactly with the correct entity type, while approximate only requires that they overlap having the identical entity type.

2.2.3 Joint Named Entity and Relation Extraction Datasets

In Chapters 4 and 5, we evaluate span-based JNERE on Adverse Drug Events (ADE) (Gurulingappa et al., 2012) as a biomedical dataset and CoNLL04 (Roth and Yih, 2004) as a general domain one as presented in Table 2.3. In addition, in Chapter 6, we include a third dataset which is SciERC (Luan et al., 2018) derived from artificial intelligence papers.

We provide the detailed description, statistics, and class frequencies of the used gold-standard JNERE datasets as follows :

Dataset	ADE	CoNLL04	SCiERC
Entity types	2	4	6
Relation types	1	5	7
Sentences	4,272	1,441	2,687
Training sentences	(10-fold)	1,153	1,861
Test sentences	(10-fold)	288	551

Table 2.3: Statistics of CoNLL04 and ADE datasets for joint entity and relation extraction.

- ADE: this dataset is extracted from medical reports with a description of adverse effects arising from prescribed drugs. ADE consists of 4,272 sentences with 6,821 relations (Gurulingappa et al., 2012). To be consistent with previous studies, we conduct a 10-fold cross-validation. ADE encompasses two entity types “Adverse-Effect” and “Drug” and a single relationship which is “ Adverse-Effect” as presented in Table 2.4.

RelationType	Number of Relations	Entity Type	Number of Entities
Adverse-Effect	6,821(100.00 %)	Adverse-Effect	5,776 (53.29 %)
Total	6,821	Drug	5,063 (46.71 %)
		Total	10,839

Table 2.4: ADE’s entities and relation statistical class distribution as pre-processed by (Eberts and Ulges, 2020).

Figure 2.7 illustrates a sentence from the ADE dataset with two entity spans, e_1 and e_2 (positive entity samples), and one relation r_1 (positive relation sample):

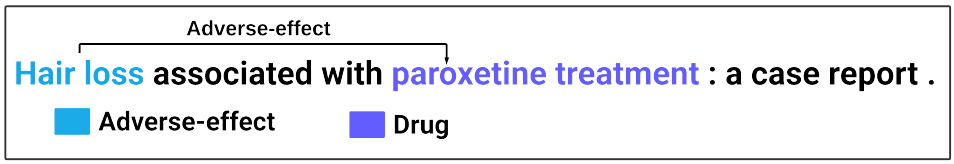

Figure 2.7: An example of adverse drug events. "Hair Loss" and "Paroxetine treatment" are two pre-defined entity types, while "Adverse-effect" is a pre-defined relation type. This snippet is part of the ADE dataset (Gurulingappa et al., 2012).

- CoNLL04: this dataset is extracted from news articles. CoNLL04's training data consists of 1,153 sentences and 288 sentences for test data (Roth and Yih, 2004). CoNLL04 has the following four entity types (a) "Location", (b) "Organization", (c) "People", and (d) "Other". CoNLL04 has five relations as follows: (a) Works for: denoted as *Work_for*, (b) Lives in: denoted as *Live_in*, (c) Kills: denoted as *Kill*, (d) Located in: denoted as *Located_in*, and (e) Organization based in: as *Organization_Based_In* as presented in Table 2.5.

RelationType	Number of Relations	Entity Type	Number of Entities
Work_for	401 (19.6 %)	Location	1,968 (36.8 %)
Live_in	521 (25.4 %)	Organization	984 (18.4 %)
Kill	268 (13.1 %)	People	1,691(31.6 %)
Located_in	406 (19.8 %)	Other	706(13.2 %)
Organization_Based_In	452 (22.1 %)	Total	5,349
Total	2,048		

Table 2.5: CoNLL04's entities and relations statistical class distribution as pre-processed by (Eberts and Ulges, 2020).

- SciERC: this dataset is derived from 500 abstracts of AI papers. SCiERC's training data consists of 1,861 sentences, 275 sentences in development dataset and 551 sentences from test data (Luan et al., 2018). The training datasets contain 6,281 entities and 3,606 relations. In Section 6.3, we give further details about the statistical class distributions of BILOU tags for NER and directed relation labels for RE.

In Chapter 6, we evaluate JNERE as a token-level in table filling context. For this, ADE, CoNLL04 and SciERC are converted into BILOU tagging scheme. As previously discussed in Section 2.2.1, the common tagging scheme is IOB. However, the BILOU is another variant that has two new tags "L-" and "U-". The "L-" represents the last /final multi-token entity. The "U-" shows a single-token entity.

Figure 2.8 depicts the differences between both schemes as follows: "Resistance" is the last token in the disease entity, thus it is labelled as "L-Disease", and "Metformin" is a single-token and labelled as "U-Drug".

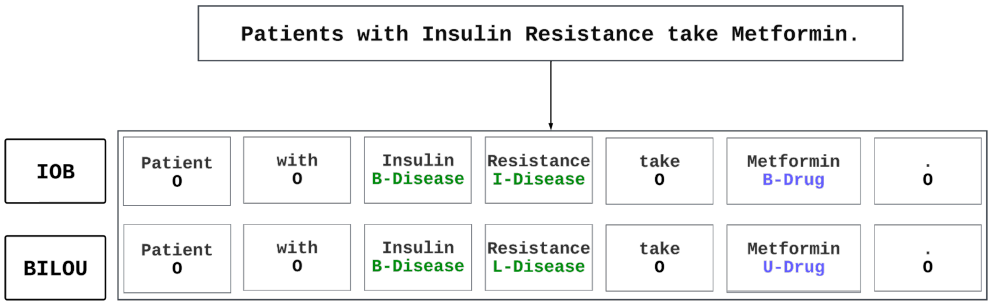

Figure 2.8: Disease and drug sequence-level tagging example using IOB and BILOU schemes.

2.2.4 Joint Named Entity and Relation Evaluation Metrics

For the span-based JNERE evaluation, we assess the performance in terms of entity recognition only, relation extraction without considering entity types, and relation extraction without considering entity types. However, for a fair comparison, we follow the baseline's evaluation method on which a correct entity has the right span and entity label and a correct relation has the right relation type and both related entities are correct as previously mentioned. We measure precision, recall, and F1 score for both tasks.

To be consistent with previous studies, we report both micro-averaged and macro-averaged for CoNLL04 and macro-averaged values for the ADE dataset. In section 2.2.2, we only evaluated BioNER using default micro-averaged evaluation. For this, we show the main differences between micro and macro-averaged evaluation as follows:

- **Micro-averaged evaluation**: it considers all predictions and instances in the dataset as a whole. Thus, it treats the dataset as a single entity and therefore, it calculates metrics based on overall true positives, false positives, and false negative counts. It provides a measure of overall performance across all classes, with more weight given to higher-frequency classes.
- **Macro-averaged evaluation**: it aggregates performance metrics by taking each class independently into account as shown below. Thus, it calculates metrics separately for each class and afterwards averages the results across all classes giving equal importance to each class regardless of its frequency.

We present the mathematical equations for the macro-averaging evaluation metrics as follows:

- Macro-averaged precision P_{macro} is calculated by first computing the precision for each class independently and then taking the average across all classes. P_{macro} is mathematically defined as:

$$P_{macro} := \frac{1}{c} \sum_{i=1}^{c} P_i \tag{2.6}$$

where:

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{2.7}$$

  - c is the total number of entity classes,
  - TP_i is the number of true positive entities for class i, and
  - FP_i is the number of false positive entities for class i.

- Macro-averaged recall R_{macro} is calculated similarly, where the recall for each class is computed and then averaged. R_{macro} is mathematically defined as:

$$R_{macro} := \frac{1}{c} \sum_{i=1}^{c} R_i \tag{2.8}$$

where:

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{2.9}$$

where FN_i is the number of false negative entities for class i.

- The macro-averaged F1 score $F1_{macro}$ is then calculated as the average of the F1 scores for each class. $F1_{macro}$ is mathematically defined as:

$$F1_{macro} := \frac{1}{c} \sum_{i=1}^{c} F1_i \tag{2.10}$$

where:

$$F1_i = 2 \times \frac{P_i \times R_i}{P_i + R_i} \tag{2.11}$$

In summary, micro-averaged evaluation focuses on the overall performance across all classes and is influenced by class distribution, giving more weight to the most frequent classes. By treating each class equally, macro-averaged evaluation provides a more balanced assessment of model performance, offering deeper insights into how well the model performs on individual classes, especially those that are underrepresented in the dataset. This makes it a valuable metric for scenarios where performance on minority classes is important.

2.3 Transformer-based Language Models

In this section, we review and categorize early transformer-based pre-trained language models (PLMs) based on their neural architectures: encoder-only, decoder-only and encoder-decoder models.

1) Encoder-only PLMs: as the name is self-descriptive, the encoder only models consist of an encoder network. At each stage, the attention layers access all the words in the initial sentence. Encoder-only models process the entire input sequence simultaneously levering the self-attention mechanism to build a contextualized representation of the input.
Originally, encoder-only PLMs were developed for language understanding tasks such as text classification on which these models predict a class label for input text. One of the earliest encoder-only models is Bidirectional encoder representations from transformers (BERT) and its variants (e.g., RoBERTa, ALBERT and XLM).
BERT remains a foundational model in the field of natural language processing (NLP). BERT proved to be a significant advancement in the field of NLP in various tasks such as language understanding, sentiment analysis and question answering. Devlin et al. (2019) developed BERT, which is built upon the transformer architecture (Vaswani et al., 2017). Transformers employ self-attention mechanisms to weigh the importance of different words in a sentence when encoding or decoding sequences of data.
As shown in Figure 2.9, BERT consists of three modules: an embedding module that converts input text into a sequence of embedding vectors, a stack of transformer encoders that converts embedding vectors into a contextual representation vectors, and a fully connected layer that converts the representation vectors to one-hot vectors. BERT is designed to understand the context of a word by considering both right and left context simultaneously, unlike previous NLP models which process words in a unidirectional manner (either left-to-right or right-to-left). Thus, this bidirectional approach allows BERT to capture richer semantic meaning from the surrounding words. BERT comprises multiple layers of transformer encoders.

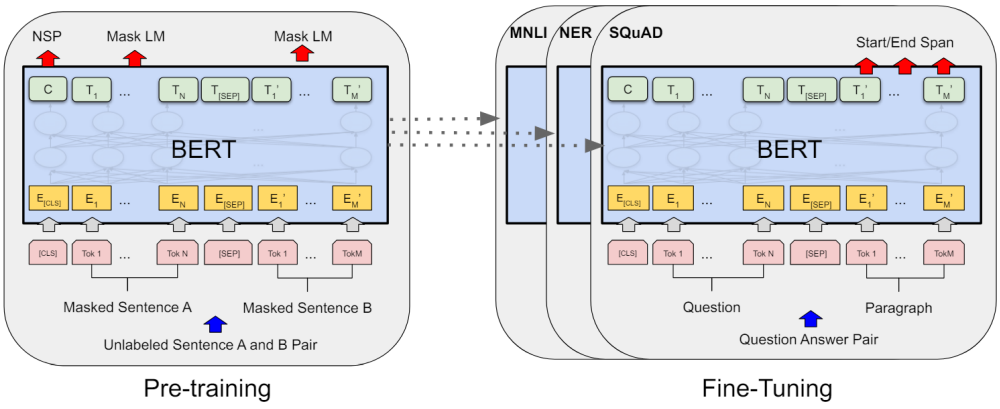

Figure 2.9: BERT's pre-training and fine-tuning procedures' illustration. Image taken from Devlin et al. (2019).

BERT tokenizes input text into subword units using WordPiece (WP). WP tokenization process iteratively matches the longest subword unit from the vocabulary to substrings of words in

the input text. If a word is not present in the vocabulary, it is broken down into individual subword units. Thus, BERT handles out-of-vocabulary words and captures fine-grained linguistic information.

BERT is pre-trained on vast amounts of unlabelled text data such as Wikipedia articles in multiple languages, book corpora and various web pages and articles. Hence, BERT is exposed to a wide range of linguistic patterns, writing domains and styles. This unsupervised training of BERT has two main objectives: masked language modelling (MLM) and next sentence prediction (NSP). For the MLM task, around 15 % of words in each input sentence are masked and BERT is trained to predict them based on the context provided by the surrounding words. Regarding NSP, BERT predicts whether a given pair of sentences appear consecutively in the original text or not.

During pre-training BERT learns bidirectional contextualized representations of words and sentences by optimizing MLM's objective, which includes predicting masked tokens within the input sequences. These pre-trained representations capture semantic information about the relationships between tokens in the input text.

Afterwards, BERT can be fine-tuned on specific downstream tasks using labelled data by leveraging the learned contextualized representations during pre-training to predict the class labels of tokens in the input sequences. Task-specific layers are added on top of the pre-trained model and the entire network is trained on the labelled task-specific data. BERT adjusts the parameters based on labelled examples to minimize a task-specific loss function, such as cross-entropy loss for single-label scenarios and binary cross-entropy loss for multi-label scenarios. Hence, BERT fine-tunes the model in a supervised learning setup.

RoBERTa is another variant of BERT that significantly improves the robustness of BERT using a set of model design choices and training strategies (Liu et al., 2019b). For instance, modifying a few key hyperparameters, removing the next-sentence pre-training objective and training with larger mini-batches and learning rates. ALBERT applies two parameter-reduction techniques as follows: factorized embedding parametrization and cross-layer sharing leading to lower memory consumption and increase the training speed of BERT (Lan et al., 2020).

ELECTRA (short for efficiently learning an encoder that classifies token replacements accurately), is another variant of encoder-only PLM architecture. Clark et al. (2020) designed ELECTRA by introducing a novel training objective, known as the "replaced token detection" task. Instead of using MLM as in BERT, ELECTRA replaces a subset of tokens in the input with plausible alternatives and trains a discriminator to distinguish between the original and replaced tokens. ELECTRA has two main components: generator and discriminator. Replaced tokens are produced by the generator and the discriminator is trained to distinguish between the original tokens and generated replacements. Both the generator and discriminator are jointly trained using adversarial learning. The generator aims to fool the discriminator and the discriminator's objective is to accurately classify the tokens. ELECTRA employs the same transformer architecture as

BERT. However, it modifies the training objective and introduces a generator-discriminator setup to improve efficiency and effectiveness. Consequently, ELECTRA's replaced token objective is computationally more efficient than BERT's masked language model. This allows larger batch sizes during training and ELECTRA has demonstrated competitive performance and achieved SOTA results on various NLP benchmarks.

Cross-lingual language model (XLMs) adapted BERT to cross-lingual language models using two approaches: unsupervised method which relies on monolingual data and supervised method that leverages parallel data with novel cross-lingual language model objective (Conneau and Lample, 2019). As depicted in Figure 2.10, the MLM objective is similar to BERT but with continuous streams of text as opposed to sentence pairs. The translation language modelling objective extends MLM to pairs of parallel sentences. XLMs are considered to be one of the SOTA results on cross-lingual classification, supervised and unsupervised machine translation.

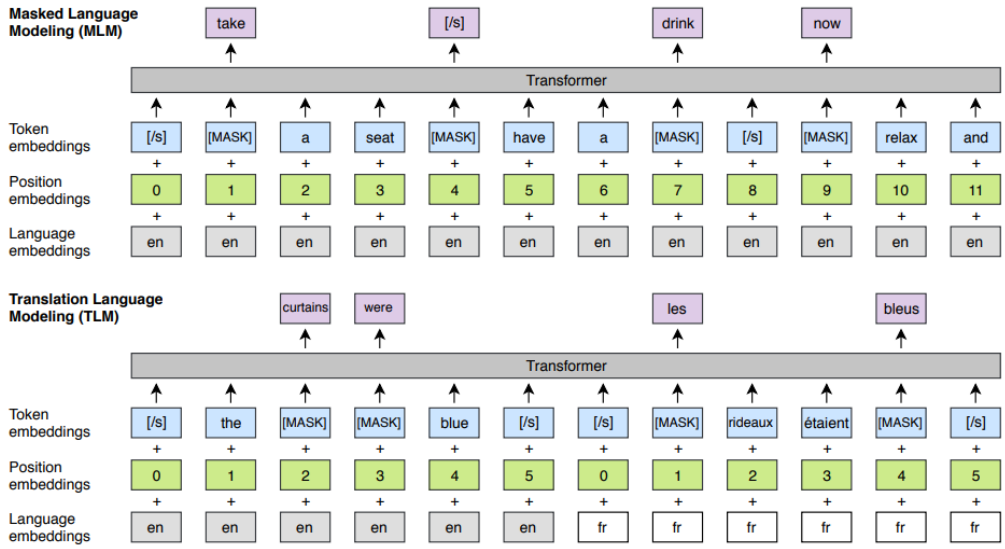

Figure 2.10: Cross-lingual language model pre-training. Image taken from Conneau and Lample (2019).

2) Decoder-only PLMs: comprised solely of a decoder stack. These models are also known as auto-regressive models. The pre-training of these models is usually formulated as predicting the next word in the sequence. Decoder-only models are best suited for tasks involving text generation. OpenAI developed two of the most popular decoder-only PLMs, namely GPT-1 (Radford, 2018) and GPT-2 (Radford et al., 2019). GPT-1 and GPT-2 models lay the foundation for more powerful large language models (LLMs), as discussed later in Section 2.3.1. The evolution of GPT-1, equipped with 117 million parameters, paved the way for subsequent GPT models with modified architecture and improved performance on various language tasks. GPT-1 achieved good performance on various corpora of unlabelled text in a self-supervised learning fashion, followed by discriminative fine-tuning on downstream tasks. GPT-2, with a model size of one and a half billion parameters,

demonstrated that language models are capable of performing various NLP tasks without explicit supervision, having been trained on large web text datasets (i.e., millions of webpages). GPT-2 is a modified version of GPT-1 with several changes as follows: normalization layer is moved to each sub-block's input, an additional normalization layer is added after the final self-attention block, the vocabulary is expanded to 50,257, context size is increased from 512 to 1,024 tokens, and initialization is modified to consider the accumulation on the residual path and to scale the weights of residual layers. GPT-2 was able to generate text that was not only coherent but also contextually relevant, raising the bar for automated text generation. However, the advancements of GPT-2 also sparked many discussions about the potential misuse of the technology(e.g., generation of fake and misleading content).

3) Encoder-Decoder PLMs: these models consist of both an encoder and a decoder stack. The encoder typically captures the input sequence's context, and the decoder generates the output sequence based on this context. These are designed as unified models that perform both natural language understanding and generation tasks. The Text-to-Text Transfer Transformer (T5) is developed as a sequence-to-sequence generation model for various NLP tasks (Rohanian et al., 2024). A multilingual version of T5, named mT5, is pre-trained on a new Common Crawl-based dataset consisting of text in 101 languages (Xue et al., 2021). BART is a denoising auto-encoder for pre-training sequence-to-sequence models (Lewis et al., 2020). BART is pre-trained by corrupting text with an arbitrary noising function; afterwards, the model learns to reconstruct the original text. Song et al. (2019) proposed MAsked Sequence to Sequence Pre-training (MASS). MASS adopts the encoder-decoder framework to reconstruct a sentence fragment given the remaining part of the sentence. The MASS encoder takes a sentence with randomly masked fragments as input, and the MASS decoder predicts the masked fragment. Thus, MASS jointly trains the encoder and decoder for language embedding and generation, respectively.

2.3.1 Language Models

With the evolution of medium to very large language models, we highlight various types of LMs. For this, in this section, we follow the same LM categorizations as proposed by Minaee et al. (2024), as depicted in Figure 2.11. LMs can be categorized based on parameters, originality, availability and type.

LMs vary in the number of training parameters. Typically, a small LM has less than or equal to one billion parameters. A medium LM has between one to ten billion parameters (exclusive), and a large LM includes between ten billion to 100 billion training parameters (exclusive). Finally, a very large LM has more than 100 billion training parameters. Original LMs are trained from scratch as foundation models. Fine-tuned LMs are those fine-tuned on different datasets using an original model. Some LMs are public, with weights shared publicly, while others are private.

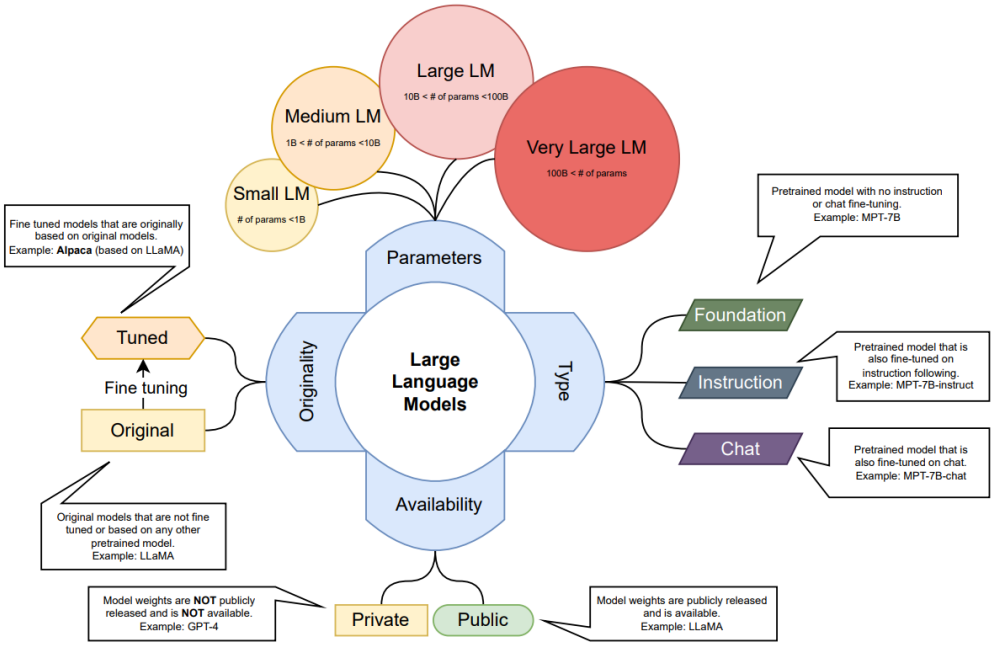

Figure 2.11: Overview of the different perspectives on LM categorizations. Image taken from Minaee et al. (2024).

Finally, LMs can be classified as follows: foundation models, which are pre-trained without instruction or chat fine-tuning; instruction-based models, pre-trained with only instruction fine-tuning; and chat-based models, pre-trained with both instruction and chat fine-tuning.

The latest GPT models including GPT-3 and GPT-4 have made remarkable strides and gained considerable attention from the research community (Kalyan, 2024). One of the special characteristics of GPT models is the exponential growth of their parameters. GPT-3 and GPT-4 feature around 175 billion and 170 trillion parameters, respectively (Koubaa, 2023). In contrast, GPT-2 has 1.75 billion parameters. Such models with this training parameter magnitude are commonly referred to as LLMs. The enhancement of LLMs is achieved via reinforcement learning with human feedback in the loop, thereby aligning text generation with human preferences. For example, GPT-3.5 builds upon the foundation of GPT-3 using reinforcement learning techniques leading to significantly improved performance in natural language understanding.

The launch of ChatGPT, a chatbot using GPT-3.5 and GPT-4 has marked a milestone in generative artificial intelligence. For instance, GPT-4 passed over 20 academic exams including the Uniform Bar Exam, SAT Evidence-based Reading and Writing and Medical Knowledge Self-Assessment Program (OpenAI, 2023).

In this section, we focus on discussing various transformer-based LLMs that contain tens to hundreds of billions of training parameters. LLMs are not only large in model size but also exhibit

improved language understanding and greater generation abilities than smaller language models, as discussed in Section 2.3.

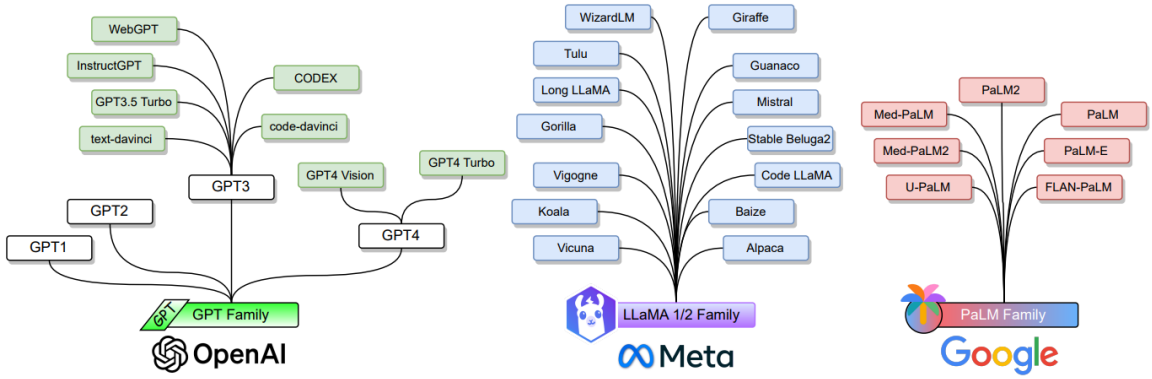

Figure 2.12: Summary of LLMs developed by OpenAI, Meta and Google. Image taken from Minaee et al. (2024).

As shown in Figure 2.12, the main LLMs are categorised as follows: GPT presented by OpenAI, Meta Llama, formerly known as LLaMa, presented by Meta, and PaLM presented by Google:

- Generative Pre-trained Transformers (GPT): they are decoder-only transformer-based language models. Early GPT including GPT-1 and GPT-2 are open-source models. Recent models such as GPT-3 and GPT-4 are close-source models that are accessed via application programming interface (APIs). GPT-1 and GPT-2 have been discussed in Section 2.3. We review other recent GPTs:
  - GPT-3: Brown et al. (2020) proposed GPT-3 as a pre-trained autoregressive language model with 175 billion parameters. GPT-3 exhibits emergent ability of in-context learning (ICL). ICL is also known as few-shot learning for LLMs. ICL enables generalizing and adapting to new tasks by providing examples within the prompt. ICL leverages the model's pre-existing knowledge and its ability to understand the context to infer generating appropriate responses (Dong et al., 2022). Thus, GPT-3 is applied to to various downstream tasks without further fine-tuning or gradient updates. GPT-3 demonstrated strong performance on many NLP taks such as translation, question-answering and others that require on-the-fly reasoning.
  - WebGPT: Nakano et al. (2021) fine-tuned GPT-3 to answer open-ended questions using a text-based web browser. WebGPT facilitates users to search and navigate the web. WebGPT is trained in three followsing steps: mimicking human browsing behaviours using human demonstration data, learned reward function to predict human preferences and refined to optimize the reward function via reinforcement learning and rejection sampling.

– InstructGPT: Ouyang et al. (2022) proposed to align language models with user intent on various range of tasks by fine-tuning with human feedback as depicted in Figure 2.13. The authors collected tailored datasets by submitting set of labeller-written prompts through OpenAI API. Afterwards, the authors fine-tuned GPT-3 on their collected dataset. In addition, a dataset of human-ranked model outputs is collected to additionally fine-tune the model via reinforcement learning. As shown in Figure 2.13, the authors applied " Reinforcement Learning from Human Feedback " (RLHF) (Ouyang et al., 2022). InstructGPT models exhibit improvements in truthfulness and reductions in toxic output generation.

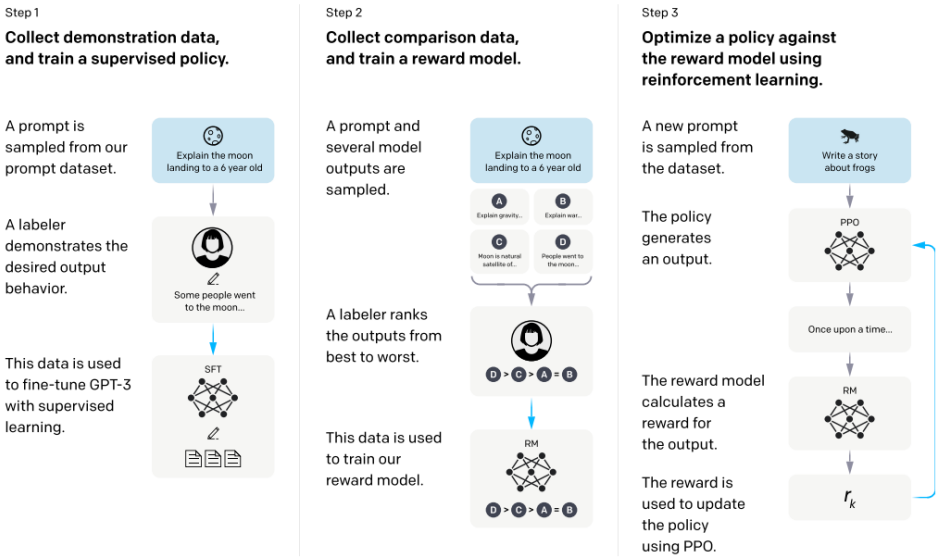

Figure 2.13: Overview of InstructGPT methods including (1) supervised fine-tuning, (2) reward model and (3) reinforcement learning via proximal policy optimization (PPO). The authors stated that blue arrows indicate that the data is used to train the model and boxes A-D in step (2) are sample from the proposed model that are ranked by labellers. Image taken from Ouyang et al. (2022).

– ChatGPT[3]: the launch of Chat Generative Pre-trained Transformer (ChatGPT) by OpenAI was in November 2022 as a chatbot that enables users to have open conversation with wide range of tasks. For instance, question-answering, text summarization, general information seeking and many more. Initially, ChatGPT was powered by GPT-3.5. GPT-3.5 is trained to follow an instruction as a prompt and generate response accordingly.

– CODEX: in March 2023, OpenAI released CODEX, a general-purpose programming model that is able to parse natural language and generate code accordingly (Chen et al., 2021). CODEX is a fine-tuned version of GPT-3 for programming applications on code corpora collected from GitHub.

[3]Introducing ChatGPT: `https://openai.com/index/chatgpt/`, last accessed: 01.08.2024.

  - GPT-4: OpenAI (2023) released GPT-4 as multi model LLM. GPT-4 was pre-trained to predict next tokens on large text corpora and then fine-tuned with RLHF. GPT-4 enables users to ask in form of free text and images and GPT-4 generate responses accordingly. Although, GPT-4 is still less capable then humans in real-world applications. However, GPT-4 exhibits a human-level performance on professional and academic benchmarks.

- Large Language Model Meta AI (Meta Llama): they are collection of open-source foundation language models released by Meta. Since Meta shares the model weights to the research community under a non-commercial license, Meta Llama grows rapidly. Such open-source models that are developed by researchers compete with closed-source ones tailored to various applications:
  - Touvron et al. (2023a) released the first set of LLaMa models ranging from seven billion to 65 billion parameters which are pre-trained on trillions of tokens from publicly available datasets. LLaMa adapted the transformer architecture of GPT-3 and added minor architectural amendments as follows: used root-mean-squared layer normalization instead of standard layer-normalization, used the Swish-Gated Linear Unit (SwiGLU) activation function instead of Rectified Linear Unit (ReLU), and utilized rotary positional embeddings instead of absolute positional embedding. One of the released models "LLaMA-13B" outperforms GPT-3 with 175 billion training parameters on most benchmarks.
  - Meta and Microsoft released "LLaMA-2" collection that include foundation language models and chat models fine-tuned on dialogue datasets named as "LLaMA-2 Chat" (Touvron et al., 2023b). LLaMA-2 Chat was pre-trained on publicly available online data, then initial version of the model is fine-tuned in a supervised fashion. Subsequently, the model is refined iteratively using RLHF, PPO and rejection sampling.
  - Alpaca[4]: is fine-tuned from Meta's LLaMA 7B model on 52K instruction-following demonstrations generated in the style of self-instruct using OpenAI's text-davinci-003. Alpaca is a smaller and cost-effective for training than GPT-3.5. Alpaca is mainly applied for academic research and performs as good as GPT-3.5.
  - Vicuna-13B[5]: the Vicuna team presented a 13B chat model that fine-tuned LLaMA on user-shard conversations collected from ShareGPT.[6] Vicuna-13B has relative limited computational demand for model training since the training cost is around three hundred dollars.

[4]Stanford Alpaca: https://crfm.stanford.edu/2023/03/13/alpaca.html, last accessed: 01.08.2024.
[5]Vicuna: An Open-Source Chatbot https://lmsys.org/blog/2023-03-30-vicuna/, last accessed: 01.08.2024.
[6]ShareGPT: https://sharegpt.com/, last accessed: 01.08.2024.

  - Mistral-7B: Jiang et al. (2023) proposed a seven billion training parameters language model. Mistral-7B leverages grouped-query attention with a fast interface. Mistral-7B is coupled with sliding window attention to effectively handle sequences of arbitrary length with a reduced inference cost. Results show that Mistral-7B outperforms LLaMA-2-13B across all evaluated benchmarks. Moreover, Mistral-7B outperforms LLaMA-34B in reasoning, mathematics and code generation.

  There are many more emerging LLaMA models such as Gorilla (Patil et al., 2023), Giraffe (Pal et al., 2023), Vigogne Huang (2023), Tulu 65B (Wang et al., 2023), Long LLaMA (Tworkowski et al., 2023) and Beluga2 (Mahan et al., 2023).

- Pathways Language Models (PaLM): Google developed the first PaLM model in April 2022 and remained private until March 2023 (Chowdhery et al., 2023). PaLM model is pre-trained on high quality text corpus consisting of 780 billion tokens from a wide range of natural language tasks and use cases. PaLM is a 540 billion parameter transformer-based LLM. With the evolution of the first PaLM model, there are various PaLM-based models built upon:

  - PaLM-2: Anil et al. (2023b) proposed a better compute-efficient LLM with multilingual and reasoning capabilities than the predecessor PaLM. PaLM-2 exhibits an improved performance on downstream tasks across different model sizes with faster and more efficient interface than PaLM.

  - Med-PaLM: a domain-specific PaLM designed to provide high quality answers to clinical questions (Singhal et al., 2022). Med-PaLM is fine-tuned on PaLM using instruction prompt tuning. Med-PaLM is a parameter-efficient approach for aligning LLMs to new domains using a few exemplars. Singhal et al. (2023) proposed Med-PaLM 2 and improved MedPaLM via medical domain fine-tuning and ensemble prompting.

  - U-PaLM: Tay et al. (2023) proposed three models of 8, 62 and 540 billions training parameters. U-PaLM is trained on PaLM and has reported 2x computational saving rate. Flan-PaLN is instruction-fined-tuned on U-PaLM (Chung et al., 2022). Flan-PaLM's fine-tuning is performed using larger number of tasks, model sizes and chain-of-thought data. The fine-tuning data includes 473 datasets and 1,836 tasks including 146 different types. Results show that Flan-PaLM outperforms instruction-following models.

- Additional uncategorised LLMs: besides the previously mentioned three categories of LLMs, we briefly highlight other popular LLMs. While we acknowledge that it is challenging to keep track of the continuously emerging LLMs, we have endeavoured to report the most popular and significant ones:

– Retrieval Enhanced Transformer(RETRO): Borgeaud et al. (2022) designed an enhanced auto-regressive language models by conditioning on document chunks. These document chunks are retrieved from a large corpus based on local similarity with preceding tokens. RETRO combines a frozen BERT retriever as differentiable encoder and a chunked cross-attention mechanism.

– Galactica: an LLM that can store, combine and reason scientific knowledge (Taylor et al., 2022). The authors trained Galactica on large scientific corpus of papers, knowledge bases, reference material and many other resources. Galactica performed well on mathematical reasoning tasks.

– BLOOM: Scao et al. (2022) presented BLOOM, a 176 billion parameter open-access LM designed and built by collaboration of hundreds of researchers. BLOOM is trained on 46 natural and 13 programming languages.

– FLAN: Figure 2.14 illustrates the comparison between pre-train fine-tuning approach, prompting and FLAN instruction tuning Wei et al. (2022). The authors improved the zero-shot learning abilities of LMs via instruction tuning. The authors utilized a 137 billion training parameter pre-trained language model and instruction tune it over 60 NLP datasets verbalized via natural language instruction templates. The results reveal that instructions improve zero-shot performance on unseen tasks.

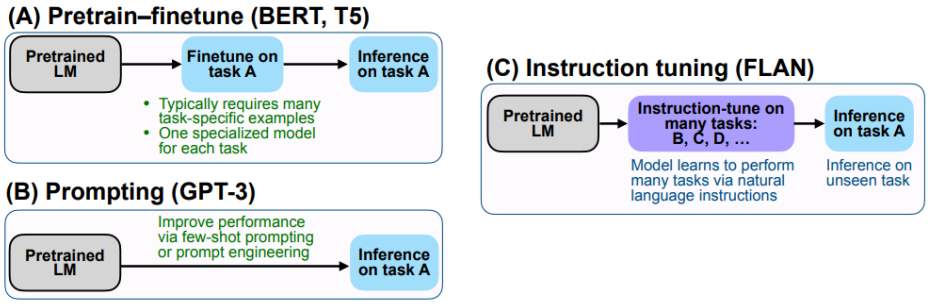

Figure 2.14: Overview of FLAN instruction tuning compared to pre-train fine-tuning and prompting approaches. Image taken from Wei et al. (2022).

– PaLM-2: Anil et al. (2023b) proposed a compute-efficient LLM. PaLM-2 has better multi-lingual and reasoning capabilities than it's predecessor PaLM. The results reveal that PaLM-2 improves the model performance on downstream tasks across different model sizes. Besides, PaLM-2's outperformance, it exhibits faster and more efficient inference than PaLM.

– Gemini: Anil et al. (2023a) introduced a new family of multimodal models. Gemini exhibits promising capabilities across audio, image, video and text understanding.

Additionally, there are other LLMs designed for efficient development frameworks, such as, Megatron-Turing NLG (Smith et al., 2022), LongFormer (Beltagy et al., 2020), Gorilla (Patil et al., 2023), PAL (Gao et al., 2023), Claude 3.5 Sonnet,[7] OPT-IML (Iyer et al., 2022), MeTaLM (Hao et al., 2022) , Dromedary (Sun et al., 2023), FuseLLM-7B (Wan et al., 2024), TinyLlama-1.1B (Zhang et al., 2024), LLaMA-Pro-8B (Wu et al., 2024), Zephyr (Tunstall et al., 2023) and many more.

2.3.2 Fine-tuning

Fine-tuning LM is the process of adapting a pre-trained model to a specific task and/or domain. Fine-tuning helps to improve the model's performance on tasks via specialized dataset. We categorize the fine-tuning into the following techniques:

- Supervised Fine-tuning (SFT) : follows a standard approach to adapt pre-trained models to tailored tasks. SFT adapts pre-trained models by further training them on new labelled datasets via supervised learning techniques (Arase and Tsujii, 2019; Zhou and Srikumar, 2022; Mosbach et al., 2021). Later, we further discuss the vanilla fine-tuning approach to biomedical applications in Section 3.2.3.

- Instruction Fine-tuning (IFT) : is also known as prompt-based fine-tuning. IFT is one of the recent adaption of SFT that is mainly applied to LLMs. IFT focuses on teaching LLMs to follow instructions to perform various tasks without explicit task-specific architecture amendments or separate training for each task. IFT is different than SFT as it enables the usage of diverse set of instructions or prompts, covering multiple tasks within the same training process (Zhang et al., 2023b).

- Parameter-Efficient Fine-tuning (PEFT) : evolved with emergence of LLMs. PEFT is designed to adapt large pre-trained model to specific downstream tasks while minimizing additional new parameters or amending only a fraction of the existing model (Han et al., 2024). PEFT aims to utilize most of the pre-trained knowledge, reduce computational costs and alleviate the potential risk of over-fitting when the dataset is small. One of the earliest PEFT strategy is Low-rank adaption of large language models (LoRA) (Xu et al., 2024). LoRA introduces low-rank matrices that are multiplied with the weight matrices of certain layers during forward and backward passes. LoRA learns task-specific modifications without the changing the original weights directly. Thus, LoRA reduces the number of additional new parameters.

Directly applying the general-domain advancements in NLP such as BERT and ELECTRA in biomedical text mining tasks often yields unsatisfactory results. This is due to the word distribution

[7] Claude 3.5 Sonnet: https://www.anthropic.com/news/claude-3-5-sonnet, last accessed: 01.08.2024.

shift from general domain corpora to the biomedical domain (Beltagy et al., 2019; Lee et al., 2020b; Gu et al., 2022). We discuss the adaptation of general-domain pre-trained language models to the biomedical domain later in Section 3.2.

2.4 Summary

In summary, this chapter provides the main foundations and concepts that are directly related to the thesis's main contributions. We introduce the basic concepts of BioNLP. We have discussed the significance of various downstream tasks like BioNER and BioRE that facilitate the process of information extraction from unstructured textual text.

In Section 2.1, we highlighted the two most popular BioNLP shared tasks on which biomedical domain experts and computer scientists collaborate to provide the community with gold-standard datasets. We introduced the BioNER concept and discussed different biomedical entity types in Section 2.1.1. We also presented the main differences between single-label, multi-label, and nested entities. In addition, we briefly provided some insights about the entity linking task and the most commonly used biomedical knowledge bases.

In Section 2.1.2, we discussed the concept of relation extraction whether binary or n-ary ones. In addition, we presented the intra-sentence and inter-sentence relations. We have clearly identified the scope in terms of entity and relation types that we are addressing in this thesis. Furthermore, we have presented the main advantages of joint entity and relation extraction and the three existing tailored paradigms in Section 2.1.3. Moreover, we have provided detailed descriptions, statistical data, and class frequencies for the BioNER and JNERE gold-standard datasets. Finally, we have presented the evaluation schemes and metrics for both tasks as well.

As discussed in Section 2.2, the biomedical gold-standard datasets are grossly imbalanced, especially NER ones. For this purpose, Chapter 3 discusses the biases in the vanilla fine-tuning approach. In addition, we present our weighted loss trainer that addresses the class imbalance problem.

Finally, we highlight different types of general-domain specific transformer-based language models in Section 2.3. In addition, we recap the recent advances in large language models up to the time of writing this thesis in Section 2.3.1. However, given the rapid pace of developments in this field, it is challenging to remain fully up-to-date. In Chapter 3, we discuss the adapted large language model to biomedical applications.

3 Weighted Loss Trainer

The core theme of this thesis focuses on addressing the class imbalance problem on recognising named entities and relations. Despite the significant advancements made by various pre-trained language models, traditional fine-tuning approaches do not effectively mitigate the class imbalance problem. The vanilla fine-tuning method uses the standard loss function which typically treats all classes equally and does not account for differences in class distribution. Thus, such models are often biased towards majority classes and struggle to classify minority classes. Therefore, this is problematic in real-world applications with highly skewed datasets. In addition, traditional fine-tuning may use evaluation metrics such as precision, recall, or accuracy. This might not adequately reflect a model's performance with imbalanced gold-standard datasets. In this case, if the model has a high accuracy, this may be misleading, as the model primarily predicts majority classes. To this end, in this chapter, we investigate the impact of applying a cost-sensitive fine-tuning approach to deal with biomedical imbalanced gold-standard datasets.

Fine-tuning biomedical pre-trained language models (BioPLMs) such as BioBERT has become a widespread practice dominating leader-boards across various natural language processing tasks. Despite major advancements and wide usage, recent work report fine-tuning instabilities for general-domain NLP tasks (Devlin et al., 2019). Besides, Lee et al. (2020a) report that small training datasets (i.e., less than 10,000 examples) are one of the potential reasons for fine-tuning instabilities. However, we argue that prevailing fine-tuning approaches for NER train BioPLMs on targeted datasets without considering class distributions. This is critical especially for most of the biomedical entities are under-represented as illustrated in Table 2.2. Consequently, the disparities between misclassification errors for different class labels are significant; making them crucial factors to consider. Thus, the error costs for rare classes in a trainer's loss function should be higher. As previously mentioned, most of the real-world biomedical datasets are highly imbalanced (Akkasi et al., 2018). Nevertheless, the impact of class imbalance before fine-tuning biomedical datasets is often not explored, especially not for NER. A few studies point out the positive impact of handling the class imbalance before fine-tuning (ValizadehAslani et al., 2022). Additionally, the authors state that fine-tuning BioPLMs on highly skewed datasets negatively affect the overall performance.

As discussed in Section 2.2.1, since these gold-standard biomedical datasets are curated by domain experts, we avoid using traditional resampling approaches as,

- removing the majority class examples leads to possible information loss, and

- duplicating the minority class examples may lead to poor performance of language models (Lee et al., 2022) and also places a burden on computational resources.

Therefore, we investigate the impact of handling the class imbalances while fine-tuning. We present a **We**ighted **L**oss **T**rainer (WeLT) that addresses the class imbalance problem by introducing coefficients that penalize majority classes and give higher weights to the rare ones.

Contributions. In this chapter, we make the following contributions:

- We propose WeLT, a class-balanced re-weighting scheme that modifies a trainer's loss functions for fine-tuning models.
- We compare WeLT to a vanilla fine-tuning approach and existing cost-sensitive class weighting methods (Suri, 2022; Cui et al., 2019). We conducted experiments on several transformers such as BERT and ELECTRA using BioBERT's entity-level evaluation script.
- We release the code[1] for the WeLT fine-tuning approach for BioNER, along with the hyperparameters needed to reproduce our research results. Besides, we release all the fine-tuned models[2] on the Hugging Face Hub (Wolf et al., 2020).
- We adapt WeLT to fine-tune BioNER models and investigate the impact of addressing the class imbalance on BioNEL. Hence, we compare WeLT recognised entities against vanilla recognized entities on the performance of BioNEL. We conduct our fine-tuning experiments on mixed-domain and domain-specific BERT models.
- We additionally evaluate WeLT using BioCreative VII NLM-Chem evaluation script (Leaman et al., 2023), assessing BioNER and BioNEL, respectively. We report both approximate and strict mode results. However, our main findings for BioNER are based on the strict mode, on which a correctly recognised entity has the right exact span, and entity type.
- We release the additional code[3] of BioNER's impact on BioNEL experiments.

Structure. We first discuss the biomedical information extraction pipeline. Furthermore, we present relevant downstream application examples in Section 3.1. In Section 3.2, we provide an overview of related work on biomedical named entity recognition, with a primary focus on pre-trained language models and traditional fine-tuning approaches. In Section 3.3, we introduce the concept of the class imbalance problem and explore various existing approaches. In Section 3.3.2, we delineate the main trade-offs between existing approaches highlighting the

[1]WeLT code: https://github.com/mobashgr/WeLT, last accessed: 01.08.2024.
[2]Fine-tuned models at Hugging Face Hub: https://huggingface.co/mobashgr, last accessed: 01.08.2024.
[3]Recognised WeLT entities code: https://github.com/mobashgr/WeLT-impact-on-BioNEL, last accessed: 30.10.2024.

research gaps. Building upon this background, we then justify the need for a custom cost-sensitive approach for BioNER models and present WeLT in Section 3.4. In Section 3.6, we evaluate the impact of recognised entities from WeLT on another biomedical entity linking and compare it to the recognised entities from vanilla models. Finally, the summary and discussion are presented in Section 3.7.

References. Parts of this chapter are based on the peer-reviewed publications:

- Ghadeer Mobasher, Wolfgang Müller, Olga Krebs, and Michael Gertz. 2023. WeLT: Improving biomedical fine-tuned pre-trained language models with cost-sensitive learning. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 427–438, Toronto, Canada. Association for Computational Linguistics

- Robert Leaman, Rezarta Islamaj, Virginia Adams, Mohammed Alliheedi, João Rafael Almeida, Rui Antunes, Robert Bevan, Yung-Chun Chang, Arslan Erdengasileng, Matthew Hodgskiss, Ryuki Ida, Hyunjae Kim, Keqiao Li, Robert E. Mercer, Lukrécia Mertová, Ghadeer Mobasher, Hoo-Chang Shin, Mujeen Sung, Tomoki Tsujimura, Wen-Chao Yeh, and Zhiyong Lu. 2023. Chemical identification and indexing in full-text articles: an overview of the NLM-Chem track at BioCreative VII. *Database J. Biol. Databases Curation*, 2023

3.1 Biomedical Information Extraction

Biomedical information extraction (BioIE) from unstructured text involves extracting relevant information from text sources such as scientific articles, clinical notes, and biomedical literature. BioIE typically involves several steps:

1. Text pre-processing in which text data are tokenized by breaking down the text into smaller units (i.e., usually words and subwords). Tokenization helps to convert raw text into a format suitable for computational analysis and allows algorithms to understand its structure and meaning. Afterwards, there are further preprocessing steps, such as lowercasing, removing punctuation, special characters, and stop words.

2. NER aims to locate and classify biomedical entities like chemicals, diseases, genes, and proteins. For instance, “glucose” is recognised as a chemical entity.

3. NEL links the identified entities to biomedical knowledge bases like MeSH IDs as discussed in Section 2.1.1. The annotation of text with concepts from these knowledge bases can facilitate interoperability and data integration across different sources.

4. RE extracts relations or associations between identified entities. For example, this involves identifying which “drug” is used to treat a particular “disease” or which “gene” is associated with a “specific disorder”.

Figure 3.1 depicts the BioIE pipeline that is composed of three stages:

- BioNER: the identified entities are classified based on pre-defined categories. For example, “ angiotensin” and “ captopril ” are recognised as chemical entities. “Pulmonary and renal insufficiency” and “intravascular coagulation ” are classified as disease entities.
- BioNEL: on which the identified entities are linked to MeSH IDs or other relevant knowledge bases as discussed in Section 2.1.1. For instance, “ captopril ” is linked to the MeSH ID D002216.[4]
- BioRE: the extracted chemical-induced disease relationships (CID) tuple is (tranexamic acid (AMCA) , CID , intravascular coagulation). Since “CID” is an asymmetrical relationship, the entity head is the chemical entity “tranexamic acid (AMCA)” and the entity tail is the disease “intravascular coagulation”.

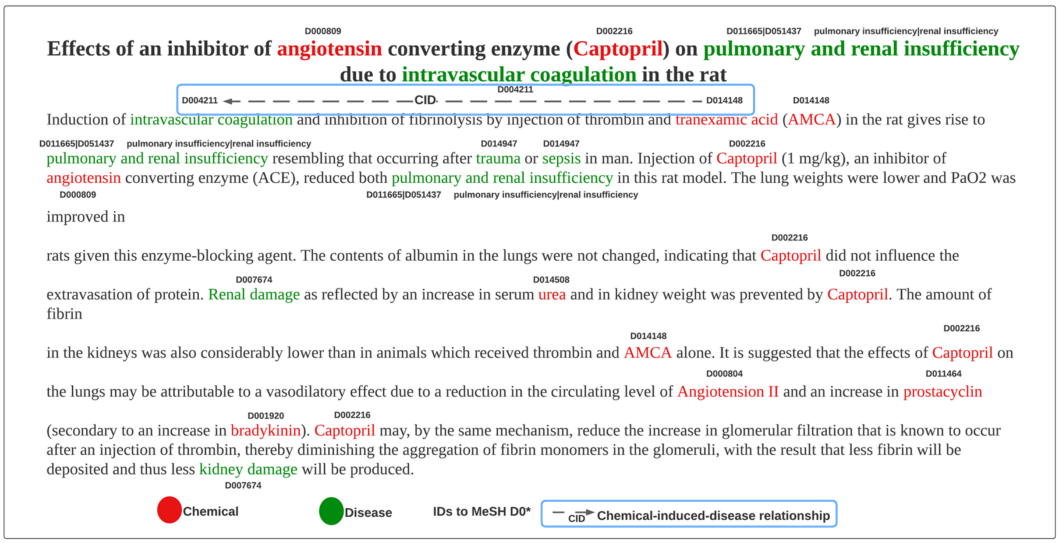

Figure 3.1: A biomedical information extraction example encompassing BioNER, BioNEL and BioRE (Eriksson and Saldeen, 1989). This snippet is part of the training data of BC5CDR for chemical-induced-disease relationship.

3.2 Biomedical Pre-trained Language Models

Directly applying the state-of-the-art NLP encoder-only, decoder-only and encoder-decoder approaches to BioNLP offers several limitations. Typically, encoder-only models such as BERT

[4] captopril MeSH descriptor data: https://meshb.nlm.nih.gov/record/ui?ui=D002216, last accessed: 01.08.2024.

are trained and tested on datasets containing general domain texts (e.g., Wikipedia). In addition, the word distributions of general and biomedical corpora are different (Habibi et al., 2017; Lee et al., 2020b). Thus, motivated by the success of general-domain language models as discussed in Section 2.3, several studies adapted language models like BERT and ELECTRA.

Lee et al. (2020b) proposed BioBERT as the first biomedical encoder-only model trained on large-scale unlabelled free text available from PubMed abstracts (PubMed)[5] and PubMed Central (PMC)[6] full-text articles. PubMed is a free resource containing over 30 million citations and abstracts of biomedical literature. PMC has open access to over five million full-text biomedical and life science research articles. BioBERT is tailored to biomedical text processing tasks and bioinformatics tasks. As shown in Figure 3.2, BioBERT is initialized via BERT's weights that are pre-trained on English general domain corpora. BioBERT is developed based on further pre-training of general BERT on biomedical texts.

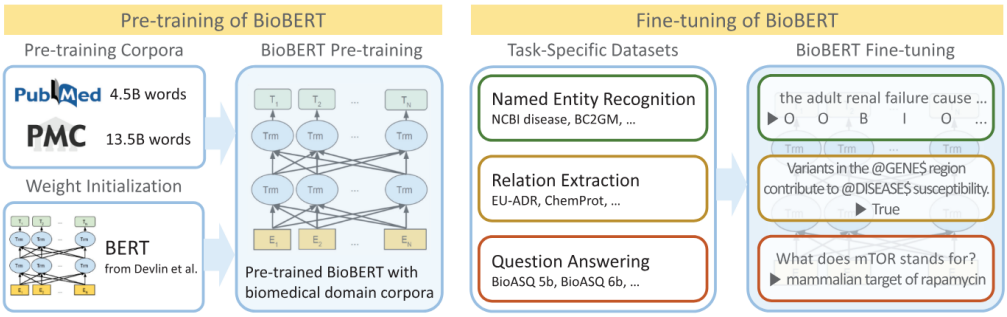

Figure 3.2: Mixed BioBERT's pre-training procedure initialized by BERT's weight using biomedical corpora including PubMed & PMC. BioBERT can be fine-tuned to various tasks. Image taken from (Lee et al., 2020b).

In contrast, BioELECTRA adapts ELECTRA; however, it is pre-trained from scratch in the biomedical domain using PubMed abstracts and PMC full-text articles (Kanakarajan et al., 2021). BioELECTRA leverages the same replaced token detection objective and generator-discriminator architecture as ELECTRA but is pre-trained on biomedical text data to capture domain-specific knowledge and terminology as shown in Figure 3.3.

Overall, while BioBERT has been a remarkable advancement in BioNLP, the evolution of other encoder-only variants and advancements on transformer models have further expanded the capabilities and applications of biomedical transformer models, paving the way for innovative solutions in biomedical research. Recently, decoder-only and encoder-decoder models have emerged, significantly shaping BioNLP research.

[5] PubMed: https://pubmed.ncbi.nlm.nih.gov/, last accessed: 01.08.2024.
[6] PubMed Central: https://www.ncbi.nlm.nih.gov/pmc/, last accessed: 01.08.2024.

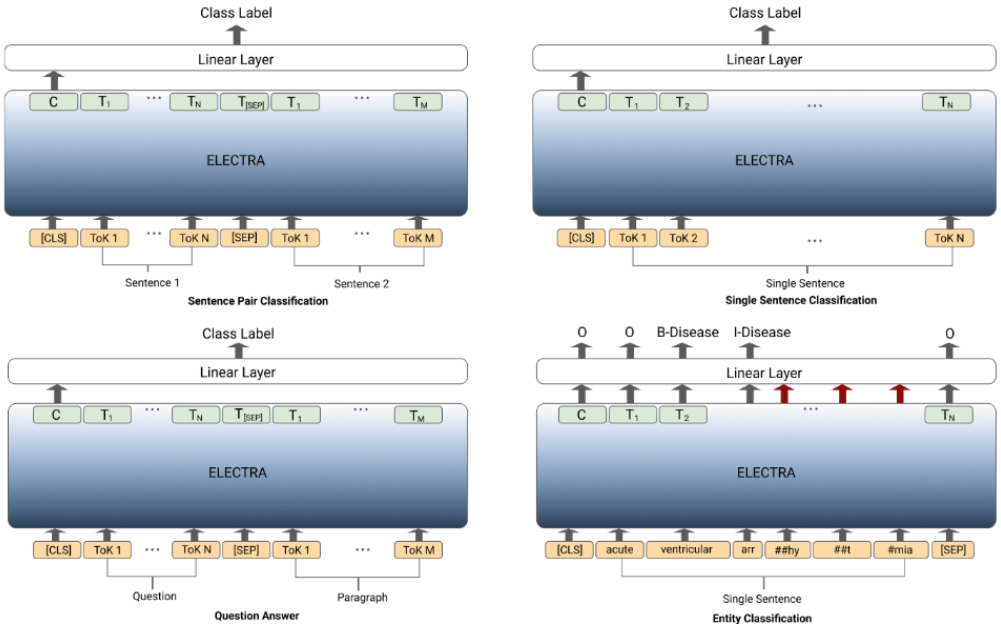

Figure 3.3: Illustration of BioELECTRA's model fine-tuning. Image taken from (Kanakarajan et al., 2021).

In Section 3.2.1, we categorise different variants of BioPLMs based on the model architecture, similarly to Section 2.3. For encoder-only models, we focus on those using WP tokenisation and categorise them according to the pre-training approach, including continual/mixed and from-scratch training. We also give particular attention to various compact biomedical BERT models. In Section 3.2.2, we review BioNER scores achieved by various language models and provide justifications for continuing to refine encoder-based models. Additionally, we present the standard loss functions for both single-label and multi-label scenarios in Section 3.2.3. Finally, the summary and open issues are discussed in Section 3.2.4.

3.2.1 State-of-the-Art in BioNER

The rapid growth of biomedical literature poses a challenge to manual curation and knowledge discovery. BioNLP has emerged as one of the potent solutions facilitating automatic information extraction. Recently, Large Language Models (LLMs) have emerged and have impressive performance on general domain applications. However, there remains a critical gap in the effectiveness of LLMs in BioNLP tasks, specifically BioNER as discussed later in Section 3.2.2.

BioNER's technical contributions enable advanced applications in literature mining, clinical decision support, drug discovery and further improvements in healthcare and biomedical sciences as highlighted in Section 2.1.1.

The state-of-the-art BioNER systems typically leverage deep learning architectures such as transformer-based models. BioBERT is the first biomedical encoder-only transformer-based

model. BioBERT is pre-trained on large-scale biomedical corpora. BioBERT afterwards is fine-tuned on BioNER datasets to capture domain-specific knowledge and terminology effectively.
Biomedical language models have adopted its architecture and can be categorised into (1) encoder-based, masked language models using the encoder from the transformer architecture such as BERT family including BioBERT (Lee et al., 2020b) and PubMedBERT (Peng et al., 2019), (2) decoder-based, generative language models using the decoder from the transformer architecture such as the GPT family including BioMedLM (Bolton et al., 2024) and BioGPT (Luo et al., 2022b), and (3) encoder-decoder-based, using both encoders and decoders such as Scifive (Phan et al., 2021) and BioBART (Yuan et al., 2022). BioNLP studies fine-tuned those language models and demonstrated that they achieved state-of-the-art performance in various BioNLP applications (Peng et al., 2019; Gu et al., 2022; Beltagy et al., 2019).

3.2.1.1 Encoder-only Models

The success of BioBERT has spurred the development of numerous variants, each aiming to address specific challenges for distinct biomedical applications. These variants are driven by the need to improve in performance, efficiency and adaptability in BioNLP. The evolution of BioBERT variants involves innovations in pre-training strategies, model architectures and training methodologies. Researchers explore techniques such as continual pre-training with different training objectives, domain-specific pre-training and compact lightweight versions. In this section, we delve into the details of different biomedical variants of BERT and ELECTRA fine-tuned for BioNER, highlighting their pre-training strategies and key characteristics:
Mixed/continual pre-training: involves training the model using both general and in-domain corpora. In continual pre-training, the model is initially pre-trained over general domain text such as books and Wikipedia. Afterwards, the model is further trained on biomedical domain-specific corpora such as PubMed and PMC full-text articles (Kalyan et al., 2022). A couple of BioPLMs are based on continual pre-training approaches:

- BioBERT is initialized with general BERT weights and then the model is further pre-trained on in-domain corpora.
- BlueBERT (Peng et al., 2019) is based on the BERT-base mode with additional pre-training in the biomedical domain from PubMed (Fiorini et al., 2018) and MIMIC-III clinical notes (Johnson et al., 2016).

Domain-specific pre-training (DSPT) models involve training from scratch on biomedical text data to capture specialized knowledge and terminology.
As previously mentioned, BioBERT and other mixed-domain PLMs continue pre-training based on the general domain vocabulary. The main drawback of this pre-training approach is the splitting of

biomedical words into several subwords, which obstructs the model learning during pre-training as well as fine-tuning. Moreover, the length of the input sequence also increases due to the splitting of in-domain words into several subwords. DSPT allows the model to have in-domain vocabulary. In the biomedical domain, PubMed and PMC full-text articles are the in-domain vocabulary. The following BioPLMs are trained from scratch:

- PubMedBERT is the first biomedical pre-trained language model that is trained from scratch. Gu et al. (2022) proposed PubMedBERT and pre-trained it on unlabelled biomedical corpora such as PubMed and PMC full-text articles as shown in Figure 3.4. The results show that PubMedBERT's pre-training approach outperforms continual pre-training of generic language models in various BioNLP tasks such as NER, relation extraction, sentence similarity, document classification and question answering.

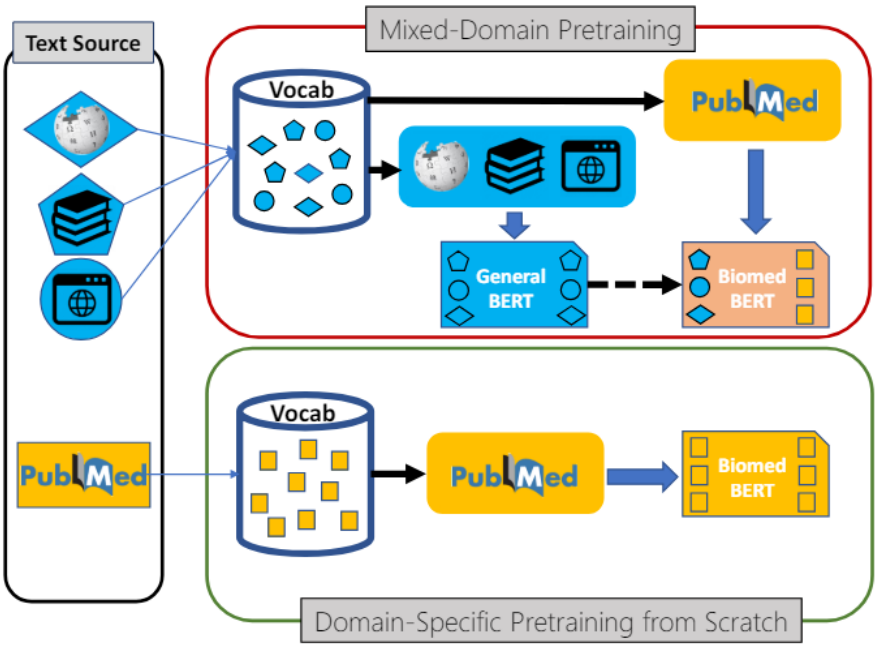

Figure 3.4: The top image illustrates mixed-domain pre-training paradigm and bottom image depicts domain-specific pre-training from scratch. Image taken from (Gu et al., 2022).

- BioELECTRA adapts ELECTRA pre-trained from scratch on biomedical unlabelled data including PubMed and PMC full-text articles. The results show that pre-training from scratch with biomedical domain text enables the model to learn better contextual representations. Kanakarajan et al. (2021) note that the pre-trained domain-specific model performs better than a model that is pre-trained on both general and biomedical text with initial weights from general domain corpora.

- SciBERT is proposed by Beltagy et al. (2019) in which the model leverages unsupervised pre-training on large multi-domain corpora of computer science and biomedicine scientific publications. Thus, from the perspective of biomedical applications, SciBERT still adopts the mixed-domain pre-training approach, as computer science articles are out-domain vocabulary.

Compact Biomedical BERT and ELECTRA: are lightweight versions of biomedical ELECTRA and BERT designed to reduce the computational resources required for training and inference while maintaining reasonable performance levels. Compact lightweight versions prioritize efficiency and scalability, making them suitable for deployment in resource-constrained environments or on-edge devices for BioNER applications. The following variants achieve efficiency by reducing model size, pruning parameters, or employing knowledge distillation techniques.

- Bio-ELECTRA and Bio-ELECTRA++ are two small ELECTRA models that are eight times smaller than BERT and BioBERT (Ozyurt, 2020). The author presented a compact BioELECTRA that is trained from scratch on PubMed and BioELECTRA++, which is a further pre-trained version of Bio-ELECTRA trained on PMC full-text articles. The author reported that small domain-specific language representation models achieve comparable or even better downstream performance on various BioNLP tasks compared to BERT which has eight time more parameters.

- Compact biomedical transformers are lightweight models proposed by (Rohanian et al., 2023). The authors pre-train three general-domain compact models, DistillBERT (Sanh et al., 2019), TinyBERT (Jiao et al., 2020), and MobileBERT (Sun et al., 2020). The authors applied two distillation techniques:
  - first, distilled versions of BioBERT result in three compact models: (1) DistilBioBERT, inspired by knowledge distillation techniques from a larger BERT model into a smaller efficient version while preserving the performance, (2) TinyBioBERT, which compresses the BERT model to reduce the model's size and computational requirements by applying knowledge distillation and pruning techniques, and (3) CompactBioBERT, a combined distillation approach of DistilBERT and TinyBERT, and
  - the second approach involves additional pre-training of a compact model on biomedical corpora via PubMed using continual learning, resulting in three models: the first two models are BioDistilBERT, BioTinyBERT, which are different variants of DistilBERT and TinyBERT, respectively. BioTinyBERT compresses the BERT model to reduce the model's size and computational requirements by applying knowledge distillation and

pruning techniques. Lastly, BioMobileBERT optimizes BERT models by reducing the model's width/hidden size.

- Bioformer reduces the model size by 60 % compared to $\text{BERT}_{\text{BASE}}$ that has 110M trainable parameters and the hidden embedding size is 768. Fang et al. (2023) pre-trained two Bioformer versions from scratch on PubMed and PMC full-text articles. "Bioformer_{8L}" is a pre-trained model with eight transformer layers and the hidden embedding size is 512. "Bioformer_{16L}" has 16 transformer layers and the hidden embedding size is 384.

3.2.1.2 Decoder-only Models

GPT models have demonstrated significant abilities on generation tasks, however directly applying general-domain GPT models have demonstrated poor performance when directly applying them to biomedical domain (Moradi et al., 2021; Gutierrez et al., 2022; Luo et al., 2022b). To this end, several domain-specific generative per-trained transformer-based language models for biomedical text generation and mining:

- BioGPT: is a generative pre-trained language model tailored to biomedical text generation (Luo et al., 2022b). BioGPT adopts the GPT-2 model (Radford et al., 2019) as its backbone. BioGPT is pre-trained from scratch on PubMed abstracts and the vocabulary is constructed via byte pair encoding to segment the words in the corpus into word pieces and learn the collected in-domain corpus. Luo et al. (2022b) evaluated BioGPT on three downstream tasks as follows: relation extraction, question answering, and document classification. Several non-published attempts[7,8,9] fine-tuned BioGPT on BioNER tasks. The results show that the best F1 score is 72.55 % on the NCBI Disease dataset, while the SOTA F1 score is 89.86 %, achieved by PubMedBERT, which is also a much smaller language model.

- Curie-FineTuned: Bousselham et al. (2024) fine-tuned GPT-3 using biomedical datasets. The authors chose to utilize "Curie" with 6.7 billion parameters. The results show that the Curie-FineTuned model achieves a lower BioNER performance on the BC5CDR dataset, with an F1 score of 75.02 %, compared to the SOTA F1 score of 91.9 % achieved using the PubMedBERT model as an encoder (Zhang et al., 2023a).

- Taiyi-LLM: Luo et al. (2023a) proposed Taiyi as a bilingual fine-tuned LLM model for biomedical NLP tasks. The authors utilized QLoRa (Dettmers et al., 2023) to fine-tune

[7] Fine-tuned version of BioGPT on the NCBI dataset conducted by Helin Wang: https://huggingface.co/westbrook/bio_gpt_ner, last accessed: 01.08.2024.
[8] Fine-tuned version of BioGPT on the NCBI dataset conducted by Timothy Lee: https://huggingface.co/timlee14/biogpt-finetuned-ner, last accessed: 01.08.2024.
[9] Fine-tuned version of BioGPT on the NCBI dataset conducted by Anna Favaro: https://huggingface.co/annafavaro/BIO_GPT_NER_FINETUNED_NEW_2, last accessed: 01.08.2024.

a general domain LLM, Qwen-7B-base (Bai et al., 2023) on medical data. Taiyi had been assessed on several BioNLP tasks on English and Chinese datasets. For the BioNER task, Taiyi had been fine-tuned on BC5CDR-Chemical, BC5CDR-Disease, BC4CHEMD and NCBI datasets.

The results show that Taiyi did not achieve SOTA F1 scores on any of the BioNER datasets. For example, the SOTA F1 score for BC5CDR-Chemical is 93.50 %, while Taiyi scored 80.20 %. A similar pattern is observed with BC5CDR-Disease, where $\text{BioBERT}_{\text{BASE}}$ v1.1 attained the SOTA F1 score of 87.15 %, and Taiyi scored 69.10 %. Additionally, PubMedBERT achieved the SOTA F1 score of 89.86 % on the NCBI dataset, while Taiyi scored 73.10 %. Finally, $\text{BioBERT}_{\text{LARGE}}$ v1.1 achieved the SOTA F1 score of 92.67 % on the BC4CHEMD dataset, with Taiyi scoring 79.90 %.

- iNERD: Deußer et al. (2023) presented informed named entity recognition decoding (iNERD), which leverages the language understanding capabilities of GPT models. The authors evaluated five generative language models on eight NER datasets. For their experimental setup, they utilized the following decoder-only models: GPT2-XL with 1.5 billion parameters (Radford et al., 2019), BioMedLM with 2.7 billion parameters (Bolton et al., 2024), RedPajama with 3 billion parameters,[10] Falcon with 7 billion parameters (Almazrouei et al., 2023), Llama with 7 billion and 13 billion parameter versions (Touvron et al., 2023a), and Llama-2 with 7 billion parameters (Touvron et al., 2023b).

  The results on NCBI dataset show that $\text{iNERD}_{\text{+GPT2-XL}}$ scored 83.79 %, $\text{iNERD}_{\text{+BioMedLM}}$ scored 86.37 %, $\text{iNERD}_{\text{+RedPajama}}$ scored 85.75 %, $\text{iNERD}_{\text{+Llama-7b}}$ scored 80.81 %, and $\text{iNERD}_{\text{+Llama-13b}}$ scored 85.07 %. However, PubMedBERT, with 110 million parameters, achieved the highest score of 89.86 %.

- There are various zero-shot and one-shot experiments that evaluated GPT-3.5 and GPT-4 as follows:
  - Chen et al. (2023) evaluated GPT-3.5-turbo-0301 and GPT-4-0314 for BioNLP tasks. Additionally, the authors fine-tuned PubMedBERT for BioNER and BioRE tasks and compared its performance against GPT-3.5 and GPT-4 in zero-shot and one-shot settings. For the BioNER task, the authors fine-tuned BC5CDR-Chemical and NCBI as evaluation datasets for chemicals and diseases mentioned in biomedical literature. For the BioRE task, they fine-tuned ChemProt for chemical-protein interactions (Islamaj Doğan et al., 2019) and DDI2013 for drug-drug interactions (Segura-Bedmar et al., 2013).

[10] RedPajama is an open-source initiative that provides a large-scale, high-quality dataset and pre-trained language models: https://www.together.ai/blog/redpajama, last accessed: 01.08.2024.

Due to the high costs of GPT-4, the authors randomly sampled 180 sentences with entities and 20 sentences without entities from each BioNER test set. The same approach was applied to the BioRE datasets, where 180 sentences with positive relation types and 20 with negative instances were sampled. The results reveal that GPT models significantly underperformed in extractive and classification tasks compared to fine-tuned PubMedBERT models, with performance gaps ranging from approximately 10 % to nearly 30 %. Specifically, PubMedBERT achieved the highest score for BC5CDR-Chemical (93.50 %), while GPT-3.5 and GPT-4, in zero-shot and one-shot settings, scored between 68.36 % and 82.43 %. A similar trend was observed for the NCBI dataset, where PubMedBERT achieved the highest score (89.86 %), while GPT-3.5 and GPT-4 scored between 38.02 % and 58.39 %.

The same pattern applied to the BioRE task. PubMedBERT achieved the best scores for ChemProt and DDI2013 (78.32 % and 80.23 %, respectively), while GPT-3.5 and GPT-4 scored between 57.43 % and 66.82 % for ChemProt, and between 33.49 % and 63.25 % for DDI2013.

– Gutierrez et al. (2022) observed that using GPT-3 for BioNER performs worse than smaller, fine-tuned pre-trained language models. Monajatipoor et al. (2024) explored the use of LLMs for BioNER by applying various prompting techniques. The authors discussed the significant role of in-context learning and the impact of input-output format on GPT-3.5-turbo and GPT-4. The results reveal that BioClinicalRoBERTa achieved the best performance among all other generative pre-trained transformer models.

– Feng et al. (2024) evaluated five LLMs, including Flan-T5-XXL (Chung et al., 2022), Azure-based GPT-3.5-Turbo, GPT-4,[11] Zephyr-7B-Beta (Tunstall et al., 2023), and fine-tuned MedLLaMA-13B on medical text data sources (Wu et al., 2023), across various BioNLP downstream tasks. Regarding the BioNER task, the authors adopted various prompting strategies, including short and long, zero-shot; short and long, random few-shot; and short and long, semantically similar few-shot. The authors did not evaluate the performance of these five language models on the full gold-standard datasets.

  For the BC5CDR-Chemical dataset: Flan-T5-XXL scored between 49.74 % and 66.98 %, GPT-3.5-Turbo scored between 59.09 % and 66.41 %, GPT-4 scored between 75.06 % and 78.23 %, MedLLaMA-13B scored between 24.55 % and 55.15 %, and Zephyr-7B-Beta scored between 53.16 % and 59.34 %. However, PubMedBERT achieved 93.50 % as the SOTA highest score.

[11] Azure OpenAI Service models: https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models, last accessed: 01.08.2024.

For the BC5CDR-Disease dataset: Flan-T5-XXL scored between 34.29 % and 54.67 %, GPT-3.5-Turbo scored between 41.50 % and 48.98 %, GPT-4 scored between 55.52 % and 63.93 %, MedLLaMA-13B scored between 11.23 % and 33.94 %, and Zephyr-7B-Beta scored between 30.84 % and 38.26 %. However, $\text{BioBERT}_{\text{BASE}}$ v1.1 achieved 87.15 % as the SOTA highest score.

Regarding the NCBI dataset: Flan-T5-XXL scored between 27.58 % and 56.10 %, GPT-3.5-Turbo scored between 47.19 % and 55.72 %, GPT-4 scored between 58.95 % and 70.59 %, MedLLaMA-13B scored between 13.44 % and 45.88 %, and Zephyr-7B-Beta scored between 30.64 % and 42.56 %. However, PubMedBERT achieved 89.86 % as the SOTA highest score.

– Rohanian et al. (2024) proposed instruction tuning for biomedical language processing. The authors have utilised two general LLMs: Llama2-Med-7B, and Llama2-Med-13B. These instruction-based models were trained on 200,000 instruction-focused samples. The results show that on the NCBI dataset, Llama2-MedTuned-7B scored 87.18 % and Llama2-MedTuned-13B scored 85.69 %. However, PubMedBERT achieved 89.86 % as the SOTA highest score.

  Regarding the BC5CDR-Disease dataset: Llama2-MedTuned-7b scored 83.92 % and Llama2-MedTuned-13b scored 85.46 %. However, $\text{BioBERT}_{\text{BASE}}$ v1.1 achieved 87.15 % as the SOTA highest score. For the BC2GM dataset, Llama2-MedTuned-7b scored 76.46 %, and Llama2-MedTuned-13b scored 79.12 %. However, $\text{BioBERT}_{\text{LARGE}}$ v1.1 achieved 85.01 % as the SOTA highest score. Unlike the three aforementioned datasets, Llama2-MedTuned-13b and Llama2-MedTuned-7b surpassed the SOTA highest score on BC5CDR-Chemical dataset, with F1-scores of 94.51 % and 93.88 %, respectively, while PubMedBERT achieved a lower F1-score of 93.50 %. Based on the reported studies, this is the only exception, and encoder-only models still hold the highest SOTA F1 score for most of the gold-standard BioNER datasets, despite having only 110 million parameters.

Finally, Tian et al. (2023) conducted an extensive literature survey on ChatGPT and LLMs in biomedicine and health, summarizing the performance of different LLMs on BioNER datasets. The authors concluded that LLMs struggle to surpass encoder-only fine-tuned models. The authors also noted that while ChatGPT and various LLMs recognise entities that sound plausible, these do not always match the gold-standard entities.

3.2.1.3 Encoder-Decoder Models

With the emergence of T5 trained on colossal clean crawled corpus (C4) and availability of biomedical datasets, Phan et al. (2021) were motivated to introduce a T5-based model tailored to biomedical domain and other encoder-decoder models are developed:

- SciFive: this is a sequence-to-sequence encoder-decoder architecture (Vaswani et al., 2017) based on the T5 framework (Raffel et al., 2020). Phan et al. (2021) trained SciFive on two biomedical datasets: unlabeled PubMed abstracts and PMC full-text articles. SciFive adapts the original structure and parameters of T5. The authors tested different variants of SciFive on various BioNLP tasks.

  For BioNER, the highest score achieved by a SciFive variant is 89.39 % on the NCBI disease dataset. However, the highest SOTA score remains 89.86 %, achieved by PubMedBERT. Regarding the BC5CDR-Disease dataset, the highest score of a SciFive variant is 87.62 %, surpassing the highest SOTA score of 87.15 %, also achieved by $\text{BioBERT}_{\text{BASE}}$ v1.1. A SciFive variant scored 94.76 % on the BC5CDR-Chemical dataset, surpassing the SOTA score of 93.50 %, achieved by PubMedBERT. For the BC4CHEMD dataset, a SciFive variant scored 92.36 %, while the highest SOTA score is 92.67 %, achieved by $\text{BioBERT}_{\text{LARGE}}$ v1.1. A similar pattern applies to the BC2GM dataset, where the highest score for a SciFive variant is 84.29 %, while the SOTA score of 85.01 % is also held by $\text{BioBERT}_{\text{LARGE}}$ v1.1.

- ClinicalT5: Lu et al. (2022) introduced ClinicalT5, a T5-based text-to-text transformer model pre-trained on clinical text. The authors trained different variants of ClinicalT5 using textual notes from MIMIC-III dataset, a publicly available health-related database (Johnson et al., 2016). ClinicalT5 has been evaluated across various clinical NLP and BioNLP tasks.

  For BioNER, the highest score achieved by a ClinicalT5 variant on the NCBI disease dataset is 87.92 %, while the highest SOTA score remains 89.86 %, achieved by PubMedBERT.

- BioBART: Yuan et al. (2022) introduced BioBART, a generative language model that adapts BART to the biomedical domain. BioBART was pre-trained on PubMed abstracts and evaluated on various clinical NLP and BioNLP tasks. However, BioBART has not been assessed on the commonly used BioNER datasets, as discussed in Section 2.2.1. Despite this, the fine-tuned datasets evaluated for NER did not surpass the SOTA results, as reported by the authors.

3.2.2 Choice of Language Models

We have presented different biomedical language PLMs in Sections 3.2.1.1 to 3.2.1.3. The encoder-based models continue to exhibit state-of-the-art BioNER F1 scores. Although our experiments were

conducted prior to the emergence of decoder-based models and LLMs, as noted in the preceding sections, the majority of the highest scores were still achieved by encoder-based models, with only occasional outperformance by decoder-only and encoder-decoder models.

Moreover, the usage of LLMs in biomedical and healthcare applications poses significant challenges and risks. For example, LLMs tend to hallucinate (Tian et al., 2023). The usage of LLMs by healthcare professionals in decision-making requires full verification of the generated information by LLMs. In addition, recent studies show that LLMs may amplify biases inherited from historical data (Shah et al., 2020; OpenAI, 2023). This is problematic in the biomedical field, where biased outputs could negatively impact the quality of patient care and lead to harmful consequences (Obermeyer et al., 2019; Sourlos et al., 2022). Furthermore, concerns around privacy persist, as some of the data used to train LLMs may contain sensitive personal information. As observed by Huang et al. (2022), generative pre-trained transformer models may inadvertently leak personal information. For instance, OpenAI has reported that GPT-4 has the capacity to identify individuals, along with their associated personal data such as phone numbers and geographic locations.

Beyond these technical concerns, the use of LLMs in medical contexts raises pressing legal and ethical issues (Sallam, 2023; Li et al., 2024). We argue that current evaluations of LLMs remain insufficient, particularly in contrast to traditional NLP tasks like NER and RE, which can be assessed through automatic metrics, such as F1 scores. In contrast, expert evaluations of LLM-generated free-text outputs are considered the gold standard, but these are labour-intensive and lack scalability. As an alternative, efforts have been made to develop less expert-dependent evaluation methods. One common approach involves converting tasks into multiple-choice questions, such as in MedQA, PubMedQA, and MedMCQA (Tian et al., 2023), where LLM performance is evaluated based on the accuracy of generated answers. However, this method is limited, as the predefined answer choices often fail to reflect the complexity of real-world biomedical queries. Another approach involves comparing LLM-generated responses against reference summaries or answers using automatic evaluation metrics. Lexical overlap metrics such as recall-oriented understudy for gisting evaluation (ROUGE) (Lin, 2004) and bilingual evaluation understudy score (BLEU) (Papineni et al., 2002) can be employed, alongside semantic similarity metrics like BERTscore (Zhang et al., 2020), BARTscore (Yuan et al., 2021), and GPT score (Fu et al., 2024).

In light of these considerations, it is crucial to continue refining encoder-based models, particularly in addressing class imbalance, to maintain the high standards of accuracy required in biomedical applications.

3.2.3 Vanilla Fine-tuning Approaches

Transfer learning via fine-tuning PLMs involves leveraging knowledge learnt from one task or domain to improve the performance of another related task or different domain. Fine-tuning involves taking a pre-trained model that has been trained on a large corpus of text data using unsupervised learning objectives and train it on a task-specific dataset using supervised learning (Devlin et al., 2019). This process allows the model to adapt its learned representations to better suit the requirements of a specific downstream task, such as text classification, token classification, or question answering. Fine-tuned PLMs for BioNER have emerged as powerful tools in biomedical text mining and NLP tasks.

With the power of transfer learning, general domain pre-trained models can be fine-tuned on biomedical textual data to achieve better performance in identifying and categorizing entities such as genes, proteins, diseases, drugs and other biomedical concepts (Lee et al., 2020b; Kanakarajan et al., 2021).

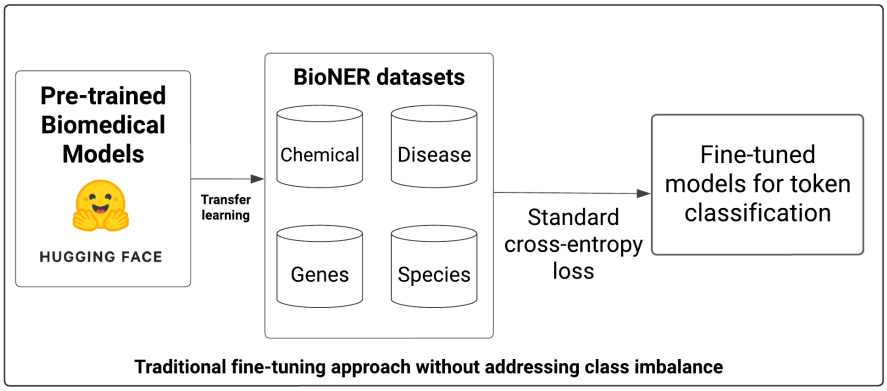

Figure 3.5: Vanilla fine-tuning approach on BioNER targeted datasets.

Table 2.1 presents the biomedical gold-standard datasets. These dataset contain unstructured text annotated by domain experts with entity labels indicating the presence and type of biomedical entities within the text. Examples of gold-standard datasets include the NCBI Disease Corpus, Linnaeus and the BioCreative datasets.

Generally, a loss function measures how well the model's predictions match the ground-truth values in the training data (Luo et al., 2023b). In other words, it quantifies the discrepancies between the model's predicted outputs and the actual target values. Thus, during learning, the model adjusts its parameters to minimize the loss, leading to more accurate predictions. In the context of NER, the loss function serves the main goal by measuring the error between the model's predicted entity labels and the true ones in the training data (Mosbach et al., 2021). Figure 3.5 depicts a vanilla fine-tuning approach for BioNER. The vanilla model is typically trained with standard loss functions. Hence, each token in the input text is treated equally, if all the training

examples have the same loss costs, regardless of their importance or rarity (Mobasher et al., 2023).

In the context of NER, standard loss functions can be categorised as follows:

- Token-level that operate at the level of individual tokens in the input text. This is suitable for tasks where the main objective is to classify each token independently, regardless of the relationship between tokens. Thus, they may struggle to handle entity boundaries.
- Entity-level that evaluate the model's performance at the level of entire named entities as the unit of evaluation. Since the main goal of NER tasks is to correctly identify and categorise entire named entities, in this thesis we evaluate our models using entity-F1 loss by calculating the F1 score at the entity level. This ensures accurate measurement of the overlap between predicted entity spans and the true spans in the dataset.

Before discussing the different standard loss functions, we highlight the main differences between the single-label and multi-label scenarios in the context of NER as presented in Section 2.1.1. In the single-label cases, each entity is classified into a single label. In other words, single-labelled entities do not belong to multiple categories simultaneously. Multi-labelled entities can be associated with multiple labels, reflecting the diverse characteristics that entities may exhibit. For instance, multi-labelled entities may include nested or overlapping entities (Sajid et al., 2023).
In a single-label scenario, cross-entropy loss denoted by CE is applied. For the multi-label scenario, binary cross-entropy loss denoted by BCE is utilized.
Equation 3.1 defines the CE loss based on the predicted probabilities and ground-truth labels, as follows:

$$CE := -\sum_{j=1}^{c} y_j \log p_j \tag{3.1}$$

where c denotes the total number of classes in the classification problem. y_j is the true label indicator for class j. Thus, y_j is 1 if class j is the correct class for the given sample, 0 otherwise. Similarly, p_j represent the predicted probability assigned to class j by the classification model.
Equation 3.2 is the binary cross-entropy loss function denoted by BCE. Since it is a binary classification, c is equal to 2. The BCE loss is computed by summing over both classes. For each class, it calculates the negative logarithm of the predicted probability for the true class label ($y_i \log p_j$) and the negative class ($1 - y_i$) log($1 - p_j$).

$$BCE := -\sum_{j=1}^{c} \left[y_j \log p_j + (1 - y_j) \log(1 - p_j) \right] \tag{3.2}$$

CE is used for problems with multiple classes, and its formulation extends to scenarios with more than two classes. BCE is only tailored to binary classification tasks with two classes, also known

as the positive and negative classes. Regarding multi-label classification, the absence or presence of each label is treated as a separate binary classification task. This means that for each label, the model predicts the probability of it being absent (0) or present (1) independently of the other labels.

3.2.4 Summary and Open Issues

Leveraging different pre-training strategies contributes to advancing BioNER capabilities and applications in biomedical research, healthcare and other related domains. Various BioPLMs have been proposed, for instance, continual or mixed-domain PLMs such as BioBERT and BlueBERT. In addition, domain-specific models such as PubMedBERT and BioELECTRA and compact lightweight BERTs like BioTinyBERT. Vanilla fine-tuned BioPLMs models for BioNER are state-of-the-art; however, they encounter various technical limitations in addressing the class imbalance problem:

- **Standard loss function bias:** the cross-entropy loss and binary cross-entropy loss are calculated based on the entire dataset during fine-tuning. This is problematic in imbalanced datasets, as the loss function can be dominated by the majority classes. Thus, the model focuses more on optimizing majority classes while neglecting minority classes. This is troublesome because biomedical gold-standard datasets mostly contain limited instances of under-represented entities magnifying data sparsity issues. Thus, vanilla fine-tuned models may not adequately address this issue, leading to insufficient learning of discriminative features for these rare entities.

- **Overfitting to majority classes:** vanilla fine-tuned models are prone to overfitting on the majority classes, especially when the training data is highly skewed towards specific entity types (Aghajanyan et al., 2021; Yuan et al., 2023; ValizadehAslani et al., 2022). The vanilla fine-tuned model may learn to memorize frequent patterns associated with the majority entities, leading to poor generalization and performance on minority classes in unseen data.

In conclusion, while vanilla fine-tuning has successfully adapted pre-trained language models for BioNER tasks, it falls short in addressing the class imbalance problem inherent in biomedical datasets, as presented in Table 2.2. In Section 3.3, we discuss the class imbalance issue in detail and review state-of-the-art approaches aimed at addressing it, while also highlighting their limitations.

3.3 Class Imbalance

The class imbalance problem typically occurs when there are more instances of certain classes than others. For example, consider a medical diagnosis system that classifies patients into one of several disease categories based on clinical data. In this scenario, the majority of the cases belong to common

conditions such as the common cold, which may have 5,000 instances, and influenza, with 1,000 instances. However, more severe or rare conditions, such as pneumonia, tuberculosis, or rare genetic disorders, are significantly under-represented, with only 500, 200, and 10 instances, respectively.
In this context, classes with the most data are called majority classes, while those with fewer examples constitute minority classes. For instance, domain experts may focus on annotating concepts related to specific aspects, such as symptoms or drugs used for therapy in full-text articles. Henning et al. (2023) reported that some rare entity types might have fewer than ten tokens across the corpus, leading to an extreme imbalance compared to the overall distribution of tokens (Johnson and Khoshgoftaar, 2019; Henning et al., 2023). This imbalance can result in biased classifiers, where models perform well on majority classes, but may struggle to classify minority ones.

Let the training dataset td consist of n token samples, and let c be the total number of distinct entity classes, where n_j represents the number of samples in class j. The total number of samples is denoted by:

$$n := \sum_{j=1}^{c} n_j \tag{3.3}$$

Class imbalance can be described by comparing the class frequencies n_j with the frequencies of other classes in the dataset, particularly the frequency of the majority class, denoted as $\max(n_1, n_2, \ldots, n_c)$. The imbalance for each class j is measured using the imbalance ratio (IR), which compares the number of instances of class j to the size of the most frequent class in the dataset. For a multi-class dataset, the imbalance IR_j for class j is defined as:

$$IR_j := \frac{\max(n_1, n_2, \ldots, n_c)}{n_j} \tag{3.4}$$

$$\text{for} \quad j \in \{1, 2, \ldots, c\}$$

where:

- $\max(n_1, n_2, \ldots, n_c)$ is the number of samples in the majority class,
- n_j is the number of samples in class j, and
- c is the total number of classes.

If $IR_j > 1$, then class j is considered a minority class. A higher imbalance ratio indicates a greater level of imbalance. Additionally, the class distribution can be expressed as a vector pr of relative frequencies:

$$pr := (pr_1, pr_2, \ldots, pr_c) \tag{3.5}$$

where each pr_j is the proportion of samples in class j with respect to the total class frequency n_k (i.e., the total number of samples in the dataset):

$$pr_j := \frac{n_j}{n_k} \tag{3.6}$$

In a perfectly balanced dataset, all the classes have an equal number of samples. Therefore, the relative frequencies pr_j for each class j will be equal, such that each class has a relative frequency denoted by $pr_j := \left(\frac{1}{c}\right)$.

3.3.1 State-of-the-Art in Class Imbalance

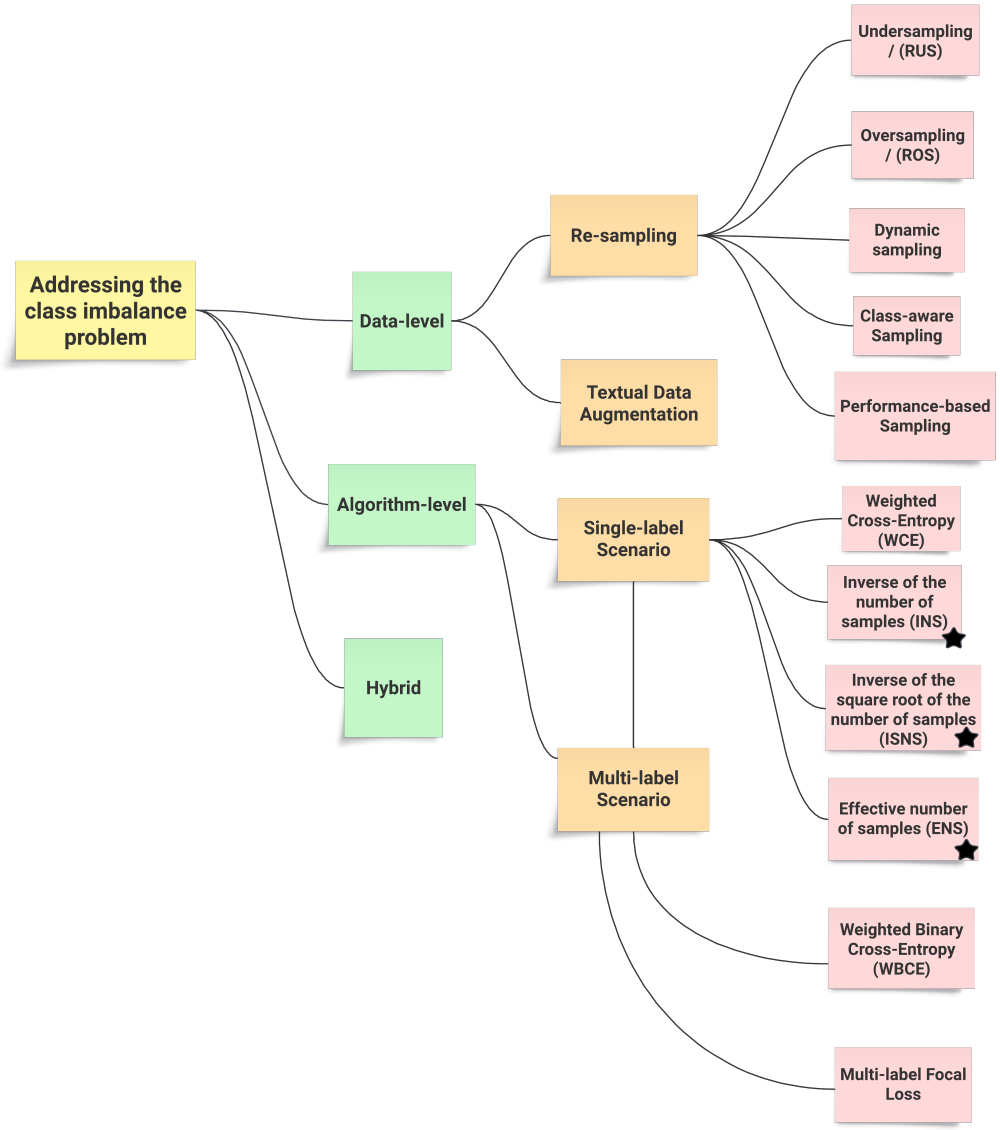

Figure 3.6: A hierarchical representation of the state-of-the-art approaches that address the class imbalance problem. The ⋆ specifies comparable weighting schemes to WeLT as discussed in Section 3.4.4.4.

Most of the NLP SOTA approaches that address the class imbalance are adapted from the computer vision tasks. In this section, we categorise the SOTA into three types: data-level, algorithm-level, and hybrid, as depicted in Figure 3.6.

3.3.1.1 Data-level Approaches

In this section, we focus on data-level approaches that address class imbalance within the context of NER tasks. We categorise these approaches into two types: re-sampling, and data augmentation. Several resampling approaches are adopted to tackle class imbalance that include the oversampling of minority instances, undersampling by removing major classes, or a combination of both.

Sampling approaches can be executed repeatedly during training and can also be done randomly, such as random oversampling (ROS) and random undersampling (RUS) (Mikolov et al., 2013). Masko and Hensman (2015) propose balancing the training data with ROS of image data. The experimental results show that applying ROS can be effective in addressing the class imbalance.

In the biomedical domain, Akkasi et al. (2018) investigated the class imbalance problem in biomedical datasets. The authors proposed a balanced undersampling approach for sequence data and enhanced classification performance by systematically removing negative samples from training data.

Class-aware sampling (CAS) addresses class imbalance during training by ensuring sufficient exposure to minority entities. CAS is achieved by dynamically adjusting the sampling ratios for different classes to increase instances of minority entities.

Shu et al. (2023) proposed a novel meta-model with class-aware sample weighting (CMW-Net). CMW-Net adaptively extracts an explicit sample weighing scheme directly from training data.

Performance-based sampling (PAS) adjusts the sampling strategy based on the model's performance in handling difficult cases. PAS aims to enhance the robustness and accuracy of the model by emphasizing challenging instances.

Strategies such as hard negative mining, misclassification-aware sampling, or uncertainty sampling fall under the umbrella of performance-based sampling. The authors proposed sentence resampling for NER based on the importance of each training sentence (Wang and Wang, 2022). The authors consider the count of entity tokens, the rareness of entity types and the density of tokens labelled as an entity as important factors for resampling functions.

Data augmentation aims to diversify the training dataset by generating new instances, achieved through introducing variations to existing data while preserving underlying patterns and semantics. In the current work context, we specifically focus on text augmentation. Textual data augmentation can be achieved through simple string-based manipulations such as synonym replacement, random insertion, deletion, or swap. Back translation is a textual data augmentation

method involving the creation of variations in sentence structure and paraphrasing while maintaining the original meaning.
Wei and Zou (2019) presented easy data augmentation (EDA), which utilizes dictionary-based synonym replacements, random swap, insertion and deletion. Juuti et al. (2020) generate new minority class instances using EDA. The authors also employ embedding-based synonym replacement to generate new minority instances for English binary text classification. Zhang et al. (2022) proposed attention-based text augmentation to address the class imbalance in long-tailed multi-label settings.

3.3.1.2 Algorithm-level Approaches

Cost-sensitive learning refers to the adaptation of predictive models to address the class imbalance by assigning varying costs or weights to classes. The goal of cost-sensitive learning is to mitigate the impact of under-represented classes allowing these predictive models to pay more attention to minority classes. Elkan (2001) proposed one of the earliest cost-sensitive approaches. Elkan introduced a factor that can be multiplied by a certain threshold, resulting in higher weights for the misclassification of minority classes.
Algorithmic approaches focus on designing new loss functions and adapting threshold adjustment. For example, focal loss (FL) is proposed to mitigate class imbalance in object detection by reformulating the standard CE in Equation 3.1. FL down-weights easily classified examples (Lin et al., 2017). Since FL is tailored to object detection tasks, there are various hard samples such as small objects, crowded scenes and noisy low-quality data. Thus, FL does not only solve the class imbalance problem but also classifies hard samples that pose challenges to correctly detect and localize objects. Nemoto et al. (2018) utilised FL for the image classification task for rare building changes. The results show that FL improves related to class imbalance and over-fitting. Cao et al. (2019) designed label-distribution-aware margin loss that optimizes the standard CE by minimizing a margin-based generalization bound.
Khan et al. (2018) introduced an effective cost-sensitive deep learning approach (CoSen CNN) that jointly learns class misclassification and network weight parameter costs during training. CoSen CNN is used to modify the output of the CNN's last layer by giving higher importance to samples with high costs. CoSen CNN is evaluated against the baseline CNN, various sampling and cost-sensitive approaches. Buda et al. (2017) adjusted CNN output thresholds to improve overall performance. The chosen threshold is based on dividing the network outputs for each class by the estimated prior probability, thus reducing the likelihood of misclassifying examples from the minority classes. This approach surpasses the baseline CNN for various image classification tasks.
In the context of NER, the weights are adjusted based on the distribution of entity types in the datasets. This adjustment aims to penalize misclassification in minority classes. The weighted cross-entropy loss function is a common practice in the development of cost-sensitive learning in NER.

As previously mentioned in Section 3.2.4, the fine-tuned models use the standard cross-entropy loss for single-label and multi-label scenarios as defined in Equations 3.1 and 3.2, respectively. The usage of such standard loss functions is problematic in the context of imbalanced datasets since these functions assume that all the classes contribute equally to the overall loss. This does not reflect the reality that certain classes have significantly fewer instances than others. Therefore, various hyperparameters or custom loss functions are adopted to address the class imbalance problem for single and multi-label scenarios.

Single-label Scenarios

In single-label multi-class classification scenarios, a common strategy to mitigate the effects of imbalanced class distributions involves the use of a modified loss function, namely the weighted cross-entropy loss[12] (WCE). In WCE, weights are assigned to enable the model to prioritise learning from under-represented classes. The WCE loss facilitates the assignment of distinct weights to each individual class. In the context of single-label multi-class classification, this entails assigning weights to each class independently, thereby tailoring the learning emphasis for each distinct class in the classification task.

Equation 3.7 defines the WCE loss function:

$$WCE := -\sum_{j=1}^{c} \alpha_j y_j \log p_j \tag{3.7}$$

where c represents the total number of classes in the classification problem, and y_j denotes true label for class j. The variable p_j refers to the predicted probability assigned to class j by the classification model. The weight assigned to each class is represented by (α_j), a positive scalar that influences the model's learning process. Higher values of α_j signal the model to prioritise learning from the corresponding class during training.

The assignment of weights α_j is typically approached through either manual specification or via equation-based methods.

Domain experts manually assign specific weights to each class, which makes this approach more subjective and context-dependent. For example, Tayyar Madabushi et al. (2019) fine-tuned BERT for sentence-level propaganda classification. The authors utilised WCE and manually set α_j values of four and one for the minority and majority classes, respectively. These values were determined through hyperparameter search, and the experiments demonstrated the importance of addressing the class imbalance problem.

[12] Cross-entropy Loss: https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html, last accessed: 01.08.2024.

In contrast, the equation-based assignment involves calculating instance-specific class weights, which are designed to reflect class frequencies. Several methods have been proposed to define instance-specific class weights, including (1) the inverse of the number of samples (INS), (2) the inverse of the square root of the number of samples (ISNS), and (3) the effective number of samples (ENS).
Equation 3.8 defines the weighted cross-entropy loss denoted as INS, where specific weights are defined based on the inverse of the actual occurrences of class j.

$$INS := -\sum_{j=1}^{c} \frac{1}{n_j}(y_j \log p_j) \tag{3.8}$$

where:

- INS is computed as the negative summation of the product of the inverse of class occurrences $\frac{1}{n_j}$,
- y_j is the ground truth indicator,
- the logarithm of the predicted probability $\log p_j$ for each class j, and
- j ranges from 1 to c

The term $\frac{1}{n_j}$ in Equation 3.8 introduces a higher penalty for classes with lower occurrences, making INS particularly sensitive to the accuracy of predictions for minority classes. In other words, INS imposes a strong penalty for misclassification of minority classes.

Equation 3.9 specifies the ISNS loss function. ISNS is a weighted cross-entropy loss that incorporates the scaling factor $\frac{1}{\sqrt{n_j}}$, which is based on the square root of class occurrences. This term adjusts the misclassification penalty for less frequent classes, making it more proportional to their occurrences.

$$ISNS := -\sum_{j=1}^{c} \frac{1}{\sqrt{n_j}}(y_j \log p_j) \tag{3.9}$$

The inclusion of the square root term provides a softer penalty for minority classes compared to INS, thereby offering a more balanced approach.
In summary, while both INS and ISNS share the objective of addressing class imbalance, they differ in their weighting mechanisms and their approach to balancing precision and recall:

- INS tends to improve recall by reducing the impact of false negatives but may lower precision due to reduced penalties for false positives.

- ISNS aims to balance precision and recall more effectively, mitigating the precision challenges typically observed with INS.

To this end, INS tends to prioritise recall at the expense of precision, whereas ISNS offers a more refined approach that seeks a balance between these two performance metrics.

Equation 3.10 defines the ENS loss function. ENS is a weighted cross-entropy loss function that specifies class weights based on the inverse of the effective number of samples $\frac{1-\beta}{1-\beta^{n_j}}$.

$$ENS := -\sum_{j=1}^{c} \frac{1-\beta}{1-\beta^{n_j}} (y_j \log p_j) \tag{3.10}$$

where $\beta \in [0, 1]$ is a hyperparameter and n_j represents the frequency of class j. The weighting factor $\frac{1-\beta}{1-\beta^{n_j}}$ is designed to adjust the contribution of each class to the overall loss. By introducing the hyperparameter β, researchers can tailor the weighting strategy to the specific requirements of the classification task (Cui et al., 2019).
ENS serves as an interpolation between the INS and ISNS weighting schemes. The following observations illustrate the relationship between ENS, INS, and ISNS:

- When β is set to 1, ENS behaves similarly to INS.
- When β is set to 0.5, ENS resembles ISNS.

Suri (2022) fine-tuned various transformer-based pre-trained models, including BERT, DistilBERT, ALBERT, and RoBERTa, for a patronising and condescending language detection task. Suri encountered challenges related to the class imbalance problem, thus, the authors explored different weighting schemes, including INS, ISNS, and ENS. The experimental results demonstrated the effectiveness of these schemes in addressing class imbalance. The results reveal that ENS outperforms other weighting scheme methods (Suri, 2022). Similarly, Li and Xiao (2020) developed a hybrid model combining two BERT models and a feature-based logistic regression model for propaganda techniques classification. Due to the skewed class distribution in the propaganda dataset, they modified the BERT cost function by employing weighted cross-entropy loss based on the reciprocal frequency of the classes. The proposed cost-weighted learning approaches effectively mitigated the class imbalance problem. Divyanth et al. (2022) proposed DeepARRNet, a deep learning model trained on a pea root image dataset to detect the rare disease "Aphanomyces root rot". The dataset exhibited class imbalance, and the authors employed two commonly used weighting schemes, INS and ISNS. The findings indicated that both weighting schemes outperformed the vanilla model, with INS yielding better results than ISNS.
In summary, all three weighting schemes share the common goal of addressing class imbalance in classification tasks, but they differ in their specific weighting formulations:

- INS assigns weights as the inverse of class frequencies, as specified in Equation 3.8. This approach enhances recall by reducing the effect of false negatives, though potentially at the expense of precision.

- ISNS incorporates a square root scaling of class frequencies, as defined in Equation 3.9. This method aims to balance precision and recall more effectively by introducing a softer penalty for majority classes.

- ENS introduces the hyperparameter β to modulate class weights, as defined in Equation 3.10, allowing for a more nuanced adjustment of class contributions. The influence of majority classes depends on the chosen value of β.

The selection of INS, ISNS, or ENS should be based on the degree of class imbalance in the dataset as defined in Equation 3.4, the desired trade-off between precision and recall, and the level of control required over the weighting mechanism.

Multi-Label Scenarios

In multi-label scenarios, each instance can belong to several classes simultaneously and the task is to predict the presence or absence of each class independently. Thus, weighted binary cross-entropy (WBCE) can be tailored to multi-label scenarios by treating each class prediction as a separate binary classification problem. WBCE computes the weighted sum of binary cross-entropy losses for both classes: one for the positive class ($y_i \, log \, p_j$), and one for the negative class (($1 - y_i$) log($1 - p_j$)).

$$WBCE := -\sum_{j=1}^{c} \alpha_j \Big[y_j \log p_j + (1 - y_j) \log(1 - p_j) \Big] \tag{3.11}$$

Equation 3.11 defines the WBCE loss function. WBCE's loss is calculated based on the binary cross-entropy between the predicted probabilities and the true labels, with the option to apply class-specific weights (α_j). WBCE is applied to binary classification tasks when there are only two classes. The weights α_j allow adjusting the importance of each class in the overall loss and mostly the weights are manually assigned.

Equation 3.12 denotes the Focal Loss function for multi-label classification $FL_{MultiLabel}$.

$$FL_{MultiLabel} := -\sum_{j=1}^{c} \Big[y_j (1 - p_j)^{\beta} \log p_j + (1 - y_j) \, {p_j}^{\beta} \log(1 - p_j) \Big] \tag{3.12}$$

$FL_{MultiLabel}$'s loss defines a hyperparameter $\beta \in [0, 5]$. This hyperparameter controls the rate at which easy examples are down-weighted relative to hard examples. This equation allows addressing class

imbalance and emphasizing hard-to-classify examples through the focal parameter β. Higher values of β lead to more aggressive down-weighting of easy examples, while lower values place less emphasis. Recent work addresses class imbalance when applying BERT for sentence classification. For example, Tayyar Madabushi et al. (2019) applied cost-weighting for a binary classification problem on which the exact weight is related to the dissimilarity of training, development and test datasets.

3.3.1.3 Hybrid Approaches

Hybrid methods integrate one or more approaches, levering the strengths of each approach. For instance, Huang et al. (2016) introduced the quintuplet sampling method, which generates discriminative representations using the large margin local embedding (LMLE) and a novel triple-header hinge loss function. LMLE effectively learns representations from imbalanced data and has achieved state-of-the-art results on benchmark image datasets. Building on this, Huang et al. (2020) proposed an improved version known as cluster-based large-margin local embedding (CLMLE), which combines LMLE with a k-nearest cluster algorithm. CLMLE outperforms existing methods in highly imbalanced face recognition and attribute prediction tasks.
Pouyanfar et al. (2018) proposed a dynamic sampling method for imbalanced image data, which involves oversampling minority classes and undersampling majority classes. This dynamic sampling approach surpasses traditional hybrids of oversampling and undersampling techniques. Ando and Huang (2017) developed a deep oversampling framework that extends synthetic oversampling techniques to the deep feature space obtained from conventional neural networks. Buda et al. (2017) compared RUS and ROS across various imbalanced image datasets. Dong et al. (2019) addressed class imbalance in computer vision applications by combining hard sample mining with a novel loss function.
We report several recent hybrid approaches as follows:

- Yang et al. (2020b) introduced a hybrid siamese CNN extremely imbalanced multi-label text classification.
- Yang et al. (2022) developed a hybrid sampling approach that combines oversampling and undersampling strategies to enhance data preprocessing effects.
- Elyan et al. (2021) designed a hybrid ensemble classifier framework that applies density-based undersampling and cost-effective methods for imbalanced data. This framework combines undersampling of negative class samples with oversampling to alleviate class imbalance.
- Groccia et al. (2023) proposed a cost-sensitive and data-sampling approach to early prediction of cardiovascular event risk. The results demonstrated that integrating cost-sensitive models with over-sampling and under-sampling techniques is effective.

- Singh et al. (2023) introduced a method called batch-balanced loss (BBFL) that addresses class imbalance in disease classification datasets. BBFL applies batch-balancing to equalize model learning across class samples and incorporates FL as a custom loss function to emphasize hard samples in the learning gradient.

3.3.2 Summary and Open Issues

Class imbalance presents a significant challenge that may adversely affect overall model performance. In biomedical texts, biomedical entities are frequently under-represented, which can lead to biased models. To address this issue, several approaches have been developed, including data-level, algorithm-level, and hybrid methods. Each of these approaches carries its own set of trade-offs:

- Data-Level approaches: these methods involve manipulating the dataset to mitigate class imbalance. While they are valuable, data-level approaches alone may not fully address issues associated with under-represented entities. Notable limitations include:
  - overfitting: generating synthetic instances for minority classes without considering their quality and diversity can lead to overfitting. This results in reduced generalisation performance on unseen data, as the model may learn false correlations and noise (Gesi and Ahmed, 2024),
  - information loss: instances of the majority class, which constitute the bulk of the training data, contain valuable linguistic patterns. Removing these instances may limit the model's ability to capture the full spectrum of linguistic variations within the majority class, thereby affecting its performance on real-world text data,
  - PLM's historical memory: PLMs exhibit a robust historical memory of linguistic patterns derived from extensive datasets. However, class imbalance in fine-tuning data may present challenges, as PLMs might prioritise learning from majority classes and consequently overlook less frequent entities (Zhu et al., 2020), and
  - impact of data de-duplication: over-sampling may degrade model quality through data duplication, which increases memory requirements and computational costs during training (Lee et al., 2022).
- Algorithm-Level approaches: these methods involve modifying the training algorithm or introducing custom loss functions to handle class imbalance. Although techniques like INS, ISNS, and ENS provide several solutions, they come with trade-offs:
  - risk of overfitting: weighting schemes such as INS are highly sensitive to class imbalance, as they assign higher weights to minority classes. While this can help the model to focus on less frequent classes, it may lead to overfitting on these classes. As a result, the model

may prioritize minimizing loss for minority classes at the expense of the overall dataset performance,

  - limited sensitivity to class imbalance: the ISNS provides a softer penalty compared to INS, but it may still under-weight minority classes, particularly in datasets with many classes or skewed distributions, and
  - complexity in hyperparameter tuning: the ENS involves a hyperparameter β that controls weighting based on the effective number of samples. This requires careful experimentation and optimization, as an inappropriate β value can lead to underfitting or overfitting, affecting the balance between emphasis on minority classes and overall class distribution.

- Hybrid Approaches: these combine multiple techniques from both data-level and algorithm-level methods, aiming to leverage their strengths. However, hybrid approaches introduce additional trade-offs:
  - increased complexity: combining multiple techniques can complicate the management of interactions and fine-tuning, making the process challenging and time-consuming,
  - high computational resources: hybrid methods often require more computational resources due to the need for training multiple models and performing ensemble techniques, which can be infeasible in resource-constrained environments, and
  - interpretation complexity: the involvement of multiple techniques makes interpreting the resulting model's behaviour more complex. Moreover, understanding how each approach contributes to the overall performance of the model often requires advanced analysis.

In summary, while state-of-the-art methods for mitigating class imbalance, including data-level, algorithm-level, and hybrid approaches, have made notable progress, substantial room for improvement remains. These approaches often face trade-offs, such as overfitting, underfitting, computational complexity, and sensitivity to noisy or imbalanced data.
In Section 3.4, we introduce the proposed WeLT approach. WeLT addresses the aforementioned limitations by offering a simple, flexible, and effective solution for cost-sensitive fine-tuning in real-world scenarios.

3.4 A Cost-sensitive Fine-tuning Approach

In contrast to state-of-the-art methods that rely on fixed weighting schemes or manually tuned hyperparameters, WeLT offers a more adaptive, and data-driven solution. WeLT addresses class imbalance by adjusting the weight of each class based on its complement relative frequency within

the dataset. The WeLT approach consists of two main steps: first, the weighted loss is computed for each class, proportional to its representation in the training data; second, the class weights are normalised to ensure balanced training.
Unlike the vanilla fine-tuning approach discussed in Section 3.2.3, WeLT represents a cost-sensitive fine-tuning strategy specifically designed to tackle the class imbalance problem, as illustrated in Figure 3.7.

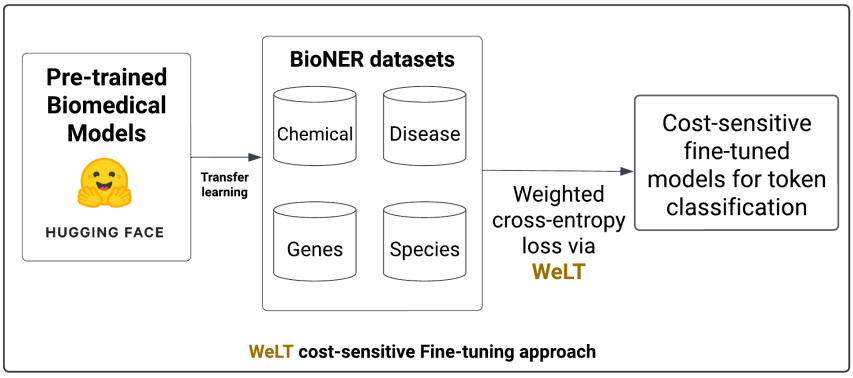

Figure 3.7: WeLT: A class-balanced re-weighting loss function for fine-tuning BioNER classifiers (Mobasher et al., 2023).

We present the WeLT approach in Section 3.4.1 and provide examples of re-scaled weights applied to real-world data. Section 3.4.4 details the experimental setup used to evaluate WeLT for BioNER, with results discussed in Section 3.4.5. Our findings demonstrate that addressing class imbalance leads to superior performance, as WeLT consistently outperforms all vanilla fine-tuned models. Furthermore, WeLT shows advantages over other existing weighting schemes in most experiments, with a comprehensive error analysis in Section 3.4.6. Section 3.6 examines the impact of incorporating balanced biomedical entities using WeLT in BioNEL. Results indicate that leveraging these entities in BioNEL improves performance compared to vanilla models. Finally, Section 3.7 provides an overall summary, highlighting the benefits of WeLT and discussing potential extensions.

3.4.1 WeLT Fine-tuning Approach

WeLT offers a flexible and adaptive solution by dynamically adjusting class weights based on dataset characteristics. It provides a softer penalty for the misclassification of minority classes via the normalisation of re-scaled class weights. This normalisation prevents overfitting to minority classes while addressing class imbalance effectively. WeLT aims to enhance the model's generalisation and reliability in real-world scenarios.

We define WeLT as the weighted cross-entropy loss function in Equation 3.13. WeLT adjusts the class weights based on the normalised complement of each class's relative frequency. Consequently, majority classes receive lower re-scaled weights, while minority classes are assigned higher re-scaled weights.

$$WeLT := -\sum_{j=1}^{c} \sigma\left(1 - \frac{n_j}{\sum_{k=1}^{c} n_k}\right) y_j \log p_j \tag{3.13}$$

Where Equation 3.14 defines the Softmax function as part of Equation 3.13:

$$\sigma\left(1 - \frac{n_j}{\sum_{k=1}^{c} n_k}\right) := \frac{e^{1 - \frac{n_j}{\sum_{k=1}^{c} n_k}}}{\sum_{l=1}^{c} e^{1 - \frac{n_l}{\sum_{k=1}^{c} n_k}}} \tag{3.14}$$

where:

- c is the total number of classes i.e., the cardinality or the size of $\mathcal{E}$, where $c = |\mathcal{E}|$, j is the index representing each class,
- σ denotes the softmax function, which is applied to the computed weights to normalise them into a probability distribution, as defined in Equation 3.14,
- n_j is the number of samples in the dataset belonging to class j,
- $1 - \frac{n_j}{\sum_{k=1}^{c} n_k}$ computes the complement of the relative frequency of class j compared to the overall frequency of all classes, and
- y_j is a binary indicator of whether the true label is class j. p_j is the predicted probability that the input belongs to class j.

WeLT integrates the softmax-normalised weights based on the complement of each class's relative frequency into the weighted cross-entropy loss function. For under-represented classes, the weighting factor is higher. This ensures that minority classes have a greater influence on the overall loss computation, while majority classes have lower weighting factors, reducing their impact.
In early experiments, applying re-scaled class weights without normalisation led to no significant improvements. This was attributed to strong penalties applied to majority classes, causing the model to be overly sensitive to minority classes. To resolve this issue, we normalised the re-scaled weights, ensuring that the sum of the weights remains balanced across all classes. The normalised complement of relative frequency is thus a more effective strategy, reducing overfitting and enabling the model to generalise well across all classes.

As highlighted in Section 2.2.1, based on the context of our work with the IOB tagging scheme, **O** is the majority class while **B** and **I** are the minority classes. Figure 3.8 illustrates the technical implementation of WeLT cost-sensitive fine-tuning approach.

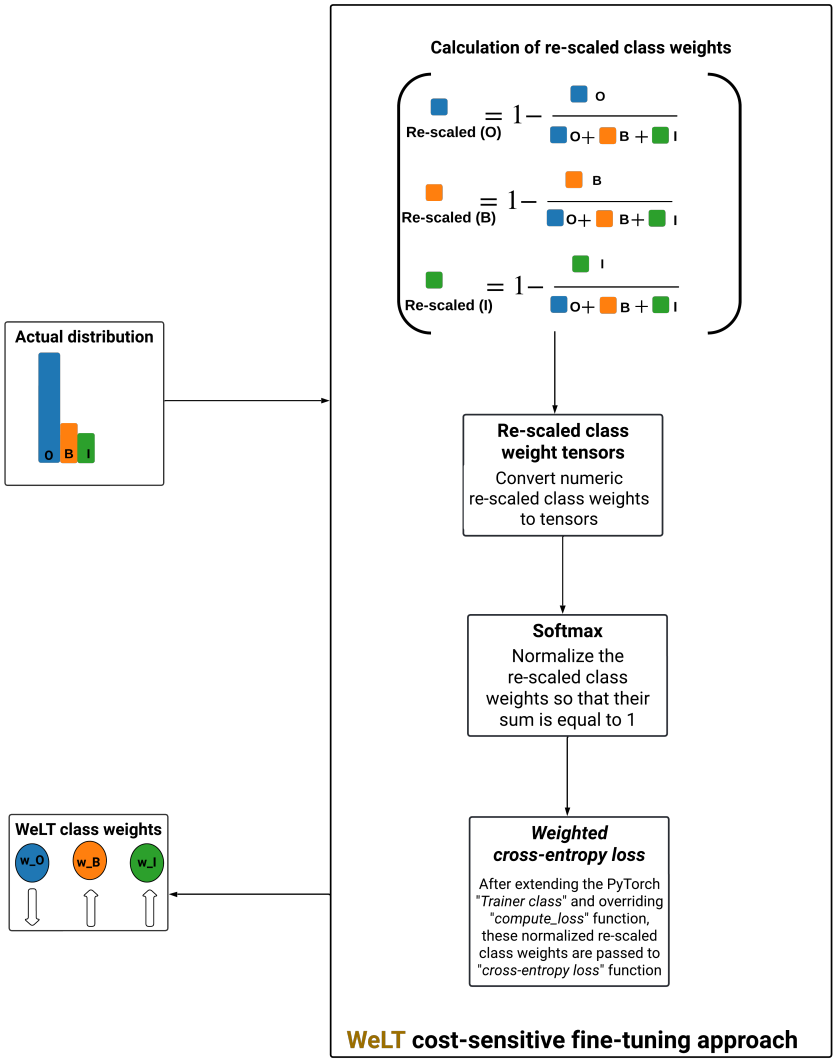

Figure 3.8: An example illustrating the calculation of normalized re-scaled class weights in WeLT.

Once the rescaled class weights are computed, they are converted to tensors and then afterwards are normalized as defined in Equation 3.13. To implement our customized WeLT Trainer, we extended the Trainer class. The Trainer class is utilised for training pyTorch models via Hugging Face Transformers[13] and override the `compute_loss` function that uses standard cross-entropy loss and define the weighted cross-entropy loss. Finally, these weights are passed into the WeLT

[13] Trainer Class: https://huggingface.co/docs/transformers/main/main_classes/trainer, last accessed: 01.08.2024.

Trainer's `compute_loss` function to ensure the model penalizes misclassifications more heavily for under-represented classes, leading to better overall performance on imbalanced datasets.

3.4.2 WeLT's Application Example

We present a generic method for calculating the re-scaled class weights for the IOB tags without requiring additional hyperparameter search factors, as defined in Equation 3.13. We illustrate the calculation of WeLT's loss function based on the NCBI dataset. Let td_{NCBI} be the training data with three classes, thus $c = 3$. Let n_o, n_b, and n_i represent the number of instances per class for the O, B, and I classes, respectively. The three-class instances are as follows:

- $n_o = 4{,}262{,}718$
- $n_b = 389{,}892$
- $n_i = 1{,}215{,}981$

Hence, the total classes frequency is denoted by $n_k = 5{,}868{,}591$. The tailored WeLT equation for the NCBI dataset is defined as:

$$\text{WeLT} := -\sum_{j=1}^{3} \sigma\left(1 - \frac{n_j}{\sum_{k=1}^{3} n_k}\right) y_j \log p_j$$

We illustrate the complement relative class frequencies compared to the average frequency of all classes, expressed as rc_o, rc_b, and rc_i for c_o, c_b, and c_i, respectively:

- $rc_o := 1 - \frac{n_o}{n_k} = 1 - \frac{4{,}262{,}718}{5{,}868{,}591} \approx 0.274$
- $rc_b := 1 - \frac{n_b}{n_k} = 1 - \frac{389{,}892}{5{,}868{,}591} \approx 0.934$
- $rc_i := 1 - \frac{n_i}{n_k} = 1 - \frac{1{,}215{,}981}{5{,}868{,}591} \approx 0.793$

The softmax-normalised values and the exponentials of the given values are as follows:

- $e^{0.274} \approx 1.315$
- $e^{0.934} \approx 2.542$
- $e^{0.793} \approx 2.209$

The sum of these exponentials is $1.315 + 2.542 + 2.209 \approx 6.066$. Let α_o, α_b, and α_i be the softmax-normalised values for classes c_o, c_b, and c_i, respectively. They are as follows:

- $\alpha_o \approx \frac{1.315}{6.066} \approx 0.217$
- $\alpha_b \approx \frac{2.542}{6.066} \approx 0.419$
- $\alpha_i \approx \frac{2.209}{6.066} \approx 0.364$

3.4.3 WeLT vs. Others

Vanilla fine-tuned trainers use cross-entropy loss and binary cross-entropy loss, as defined in Equations 3.1 and 3.2, respectively, to minimise the training error by assuming that individual samples and classes are equally important (i.e., all the class frequencies are sufficiently balanced). Since biomedical gold standard training datasets are highly imbalanced, we use the weighted cross-entropy loss. WeLT's normalised re-scaled class weights are passed to the weighted cross-entropy loss function, as specified in Equation 3.13, after extending the class trainer and overriding the `compute_loss` function (Paszke et al., 2017).
Subsequently, the models are fine-tuned using WeLT with the same exact training cost as the vanilla fine-tuned approach. Despite the great efforts of existing weighting schemes, they still have limitations, as mentioned in Section 3.3.2.
To bridge this research gap, WeLT seeks to offer tailored and complementary solutions, including the following:

- adaptivity to various class imbalances: WeLT presents a tailored approach that considers both class imbalance and the dataset's specific characteristics. WeLT offers fair weight adjustments that are computed equally for all classes in the dataset. Thus, allowing for dynamic adaptation to different class distributions. This adaptability ensures improved performance across various datasets and imbalance scenarios,

- better weighting scheme: WeLT integrates the softmax-normalised weights based on the complement of each class's relative frequency. Instead of relying solely on existing class-specific weighting schemes such as INS and ISNS, WeLT leverages the overall dataset's distribution to tailor the weighting scheme dynamically,

- balanced emphasis on classes: WeLT finds a compromise between mitigating class imbalance and maintaining performance across all classes. This is achieved by effectively prioritising minority classes while minimising the impact on majority ones via the softmax normalisation

step. Thus, WeLT ensures that all classes receive adequate attention during training, reflecting the real-world dataset's overall class distribution, and

- a simple yet effective solution: WeLT presents a simplified weighting scheme by directly incorporating softmax normalised re-scaled class weights without the additional need for hyperparameters. This saves additional costs associated with determining the best hyperparameter values, such as the β in ENS.

In summary, WeLT provides an easy adaptable approach that can accommodate datasets with diverse levels of imbalance, as discussed later in the following section. This adaptability is crucial in real-world scenarios where biomedical datasets exhibit high degrees of imbalance across different classes. Additionally, WeLT offers a simple weighting scheme, as it does not require additional calculations and hyperparameter tuning for each individual class as illustrated in Figure 3.8.
In the following section, we conduct several experiments to evaluate WeLT's performance and compare it against the vanilla trainer and other weighting schemes.

3.4.4 Evaluating WeLT on BioNER

We demonstrate the performance of WeLT's loss function through various experiments on eight biomedical gold-standard datasets focusing on the BioNER task. We evaluate WeLT on both mixed-domain and domain-specific BERT and ELECTRA models. We compare WeLT to their corresponding vanilla fine-tuning approach and three existing weighting schemes. We assess the behaviour of WeLT when being fine-tuned while dealing with different dataset sizes and a variety of class distributions. In addition to the experimental analysis, we further share the implementation details and evaluation metrics.

3.4.4.1 Evaluation Datasets and Metrics

As mentioned in Section 2.2.2, we have fine-tuned eight gold-standard datasets including various entity types such as disease, chemical, genes and species. In addition, we added the latest BioRED dataset after further pre-processing steps as follows: (1) filtering of the human-annotated chemical and disease entities, (2) and the conversion of BioC XML format (Comeau et al., 2013) to the IOB tagging scheme to be consistent with datasets in the format of BioBERT-PyTorch using bconv.[14]
For a fair comparison, we have evaluated WeLT's fine-tuning approach with the same evaluation script as BioBERT. We report the entity-level micro-averaged precision, recall and F1 scores as highlighted in Section 2.2.2. We additionally evaluate the annotation quality for species entities on the Linnaeus dataset. Hence, we used two sequence labelling metrics: seqeval, and FairEval.

[14]bconv: Python library for converting between BioNLP formats: https://github.com/lfurrer/bconv, last accessed: 01.08.2024.

3.4.4.2 Baselines

Due to the nature of our work that investigates the impact of addressing the class imbalance before fine-tuning, we do not compete with the state-of-the-art BioNER baselines. However, we compare the vanilla fine-tuning approach to WeLT and three existing weighting schemes using the same hyperparameters. We report the hyperparameter settings to reproduce our results in the Appendix (see Tables 1- 8).

3.4.4.3 Experimental and Implementation Settings

In this section, we report the comparable weighting schemes in Section 3.4.4.4. Our experiments include six different fine-tuning approaches and they are compared to WeLT's approach. We investigate the impact of addressing the class imbalance problem on general-domain and domain-specific pre-trained language models as highlighted later in Section 3.4.4.5 on the eight biomedical gold-standard datasets. All the experiments were carried out using a single Tesla P40 GPUs with 24GB memory.

3.4.4.4 Comparable Weighting Schemes

We choose the comparable weighting schemes for the single-label scenario, as discussed in Section 3.3.1.2. Therefore, we compare WeLT with the vanilla fine-tuned approach. In addition, we evaluate WeLT alongside other weighting schemes: INS, ISNS, and ENS, as defined in Equations 3.8, 3.9, and 3.10, respectively. Regarding the ENS approach, we use different values for β, representing a lower bound, median, and upper bound, as follows: 0.3, 0.5, and 0.9, respectively.

3.4.4.5 General and Domain-specific Language Models

We investigate the effectiveness of addressing the class imbalance problem using both general and domain-specific pre-trained language models as outlined in 3.2.1. Moreover, we evaluated the performance of different transformer architectures such as BERT and ELECTRA.
We adapted the BioBERT (Lee et al., 2020b) PyTorch NER code to develop WeLT. In our experiments, we used the following five pre-trained model variants: BioBERT, BlueBERT, PubMedBERT, SciBERT, and BioELECTRA.

3.4.4.6 Fine-tuning and Hyper-parameter Settings

For a fair comparison, we used the same hyperparameters for fine-tuning the BioNER models. For more details on the hyperparameters, see the Appendix (Tables 1-8).

3.4.5 Results and Discussion

We present the results of seven fine-tuning experiments, which include:

- INS, as defined in Equation 3.8.
- ISNS, as defined in Equation 3.9.
- ENS, as defined in Equation 3.10, applying three values for β: $\beta = 0.3$, $\beta = 0.5$, and $\beta = 0.9$.
- The vanilla fine-tuning approach, as defined in Equation 3.1.
- The WeLT fine-tuning approach, as defined in Equation 3.13 and illustrated in Figure 3.8.

Thus, we have 280 experimental results, as presented in Tables (3.1 to 3.8). We extensively report the results based on the following three criteria:

- **Class imbalance percentage**: regarding the class distribution percentage, as presented in Table 2.2, the Linnaeus dataset is the most highly skewed dataset, and NCBI is the least imbalanced one. WeLT achieves the highest score for the experiments related to Linnaeus, as presented in Table 3.1, except for the fine-tuned BioBERT, achieving the second-best score.

Model	Metrics	INS	ISNS	ENS ($\theta = 0.3$)	ENS ($\theta = 0.5$)	ENS ($\theta = 0.9$)	Vanilla	WeLT(ours)
BioBERT	P	**92.52**	90.10	89.32	91.81	92.17	91.81	91.21
	R	78.57	83.87	**86.39**	85.34	86.25	85.34	86.25
	F1	84.98	86.88	87.83	88.46	**89.11**	88.46	88.66
PubMedBERT	P	**88.49**	85.47	85.31	85.63	85.15	85.63	86.98
	R	79.41	77.59	76.62	79.48	**80.87**	79.48	80.66
	F1	**83.70**	81.34	80.73	82.44	82.96	82.44	**83.70**
BlueBERT	P	91.23	91.10	90.73	91.15	90.24	90.97	**91.35**
	R	50.87	64.34	65.59	64.41	64.54	85.83	**86.25**
	F1	65.32	75.41	76.14	75.47	75.26	88.33	**88.72**
SciBERT	P	88.43	91.41	91.02	90.51	90.65	90.51	**92.44**
	R	46.96	62.38	65.80	64.61	66.36	64.61	**66.57**
	F1	61.34	74.16	76.38	75.40	76.63	75.40	**77.40**
BioELECTRA	P	79.07	82.38	80.56	82.82	82.39	82.82	**84.15**
	R	70.41	75.71	79.83	81.08	81.99	81.08	**82.62**
	F1	74.49	78.90	80.19	81.94	82.19	81.94	**83.38**

Table 3.1: The Linnaeus fine-tuning scores comparing WeLT with three weighting schemes and the vanilla fine-tuning approach. Precision (P), Recall (R) and F1-score (F1) are the evaluation metrics. The best scores are shown in bold, and the second-best ones are underlined.

Similarly, WeLT achieves the best score for all experiments using NCBI, except for BlueBERT, as presented in Table 3.2.

- **Size of training datasets**: according to the statistics presented in Table 2.1, the BioRED dataset is the smallest dataset.

Model	Metrics	INS	ISNS	ENS ($\theta = 0.3$)	ENS ($\theta = 0.5$)	ENS ($\theta = 0.9$)	Vanilla	WeLT(ours)
BioBERT	P	85.75	85.41	86.74	85.23	85.33	86.12	**86.87**
	R	87.81	88.43	88.64	87.81	87.91	87.91	**88.95**
	F1	86.77	86.89	87.68	86.50	86.60	87.01	**87.90**
PubMedBERT	P	80.73	81.36	79.70	79.68	81.58	79.68	**82.45**
	R	77.70	**80.52**	79.79	80.10	80.31	80.10	79.79
	F1	79.19	80.94	79.75	79.89	80.94	79.89	**81.10**
BlueBERT	P	**86.76**	86.47	86.52	86.17	86.36	86.17	86.40
	R	88.75	89.27	90.31	88.95	**91.04**	88.95	90.00
	F1	87.74	87.85	88.37	87.54	**88.64**	87.54	88.16
SciBERT	P	86.38	85.77	84.96	85.95	**86.73**	85.95	86.34
	R	88.54	89.16	88.33	89.27	88.54	89.27	**89.58**
	F1	87.44	87.43	86.61	87.58	87.62	87.58	**87.93**
BioELECTRA	P	86.55	87.26	85.74	85.65	85.65	85.65	**87.66**
	R	83.85	87.81	88.33	88.33	88.33	88.33	**88.85**
	F1	85.18	87.53	87.01	86.97	86.97	86.97	**88.25**

Table 3.2: The NCBI fine-tuning scores comparing WeLT with three weighting schemes and the vanilla fine-tuning approach. Precision (P), Recall (R) and F1-score (F1) are the evaluation metrics. The best scores are shown in bold, and the second-best ones are underlined.

WeLT achieves the best performance in fine-tuning experiments, except for PubMedBERT on BioRED-Disease, as presented in Table 3.3. A similar trend is observed for BioRED-Chemical in Table 3.4, except for BioBERT and BlueBERT.

Model	Metrics	INS	ISNS	ENS ($\theta = 0.3$)	ENS ($\theta = 0.5$)	ENS ($\theta = 0.9$)	Vanilla	WeLT(ours)
BioBERT	P	84.38	**84.46**	83.91	81.52	84.03	81.52	83.79
	R	84.57	85.66	86.21	84.46	84.68	84.46	**86.54**
	F1	84.48	85.06	85.05	82.96	84.35	82.96	**85.14**
PubMedBERT	P	58.57	66.77	**71.77**	68.92	69.79	68.92	67.95
	R	43.32	64.87	**68.70**	65.75	67.50	65.75	67.50
	F1	49.81	65.81	**70.20**	67.30	68.63	67.30	67.72
BlueBERT	P	65.56	68.88	67.16	65.10	66.56	65.10	**69.32**
	R	56.67	64.44	68.27	66.95	**68.38**	66.95	66.52
	F1	60.79	66.59	67.71	66.01	67.45	66.01	**67.89**
SciBERT	P	68.98	**73.75**	71.91	71.68	72.33	69.29	72.34
	R	60.83	66.41	64.98	**68.70**	67.50	68.16	67.83
	F1	64.65	69.89	68.27	**70.16**	69.83	68.72	70.01
BioELECTRA	P	83.88	**85.54**	83.97	84.02	83.97	84.02	84.95
	R	85.44	87.41	88.29	88.07	88.29	88.07	**89.60**
	F1	84.66	86.47	86.08	86.00	86.08	86.00	**87.22**

Table 3.3: The BioRed-Disease fine-tuning scores comparing WeLT with three weighting schemes and the vanilla fine-tuning approach. Precision (P), Recall (R) and F1-score (F1) are the evaluation metrics. The best scores are shown in bold, and the second-best ones are underlined.

On the other hand, BC4CHEMD is the largest dataset. WeLT achieves the second-best score in all experiments, as presented in Table 3.5, except for BioBERT.

Model	Metrics	INS	ISNS	ENS ($\theta = 0.3$)	ENS ($\theta = 0.5$)	ENS ($\theta = 0.9$)	Vanilla	WeLT(ours)
BioBERT	P	87.93	88.73	**90.56**	88.37	89.54	85.00	88.57
	R	80.77	84.11	87.18	**87.31**	86.91	84.77	86.91
	F1	84.20	86.36	**88.84**	87.84	88.21	84.89	87.73
PubMedBERT	P	89.03	88.30	88.72	89.55	89.04	89.55	**90.57**
	R	82.37	**87.71**	86.11	85.84	85.71	85.84	85.98
	F	85.57	88.01	87.39	87.66	87.34	87.66	**88.21**
BlueBERT	P	86.63	87.18	87.29	**89.05**	88.36	86.42	88.80
	R	86.51	88.11	**90.78**	90.12	88.25	90.12	88.91
	F1	86.57	87.64	89.00	**89.58**	88.30	88.23	88.85
SciBERT	P	73.86	80.88	79.84	82.37	81.60	**82.53**	81.48
	R	49.79	58.74	**67.69**	66.75	63.95	64.35	67.55
	F1	59.48	68.05	73.26	73.74	71.70	72.31	**73.86**
BioELECTRA	P	**89.64**	86.27	88.11	61.55	85.97	61.55	89.43
	R	77.43	85.58	84.11	53.00	**86.78**	53.00	84.77
	F1	83.09	85.92	86.06	56.95	86.37	56.95	**87.04**

Table 3.4: The BioRed-Chemical fine-tuning scores comparing WeLT with three weighting schemes and the vanilla fine-tuning approach. Precision (P), Recall (R) and F1-score (F1) are the evaluation metrics. The best scores are shown in bold, and the second-best ones are underlined.

Model	Metrics	INS	ISNS	ENS ($\theta = 0.3$)	ENS ($\theta = 0.5$)	ENS ($\theta = 0.9$)	Vanilla	WeLT(ours)
BioBERT	P	**92.49**	91.97	91.84	91.78	91.48	91.78	91.70
	R	88.14	89.80	90.54	90.21	**90.93**	90.21	90.45
	F1	90.26	90.88	91.18	90.99	**91.20**	90.99	91.07
PubMedBERT	P	**91.69**	81.36	91.21	90.75	90.06	90.75	91.36
	R	85.59	80.52	88.63	88.43	**88.87**	88.43	88.38
	F1	88.54	80.94	**89.90**	89.57	89.46	89.57	89.84
BlueBERT	P	**89.55**	89.21	89.07	88.67	88.88	88.67	89.22
	R	82.81	84.62	85.68	**85.85**	**85.85**	**85.85**	85.46
	F1	86.05	86.85	**87.34**	87.24	**87.34**	87.24	87.30
SciBERT	P	**81.17**	80.18	80.05	80.20	80.90	79.71	79.71
	R	68.13	72.88	**75.03**	74.43	73.67	74.35	74.99
	F1	74.08	76.36	**77.46**	77.21	77.12	76.93	77.28
BioELECTRA	P	**93.19**	92.71	92.87	92.57	92.70	92.57	93.01
	R	89.80	91.02	91.70	91.85	**92.00**	91.85	91.61
	F1	91.46	91.86	92.28	92.21	**92.35**	92.21	92.30

Table 3.5: The BC4Chem fine-tuning scores comparing WeLT with three weighting schemes and the vanilla fine-tuning approach. Precision (P), Recall (R) and F1-score (F1) are the evaluation metrics. The best scores are shown in bold, and the second-best ones are underlined.

- **Pre-training approach**: WeLT achieves the best score in the BC5CDR-Chemical experiments, as presented in Table 3.6, except for fine-tuning SciBERT, where it achieves the second-best score for PubMedBERT.

  The BC5CDR-Disease experiments, as presented in Table 3.7, indicate that WeLT achieves the best score in all experiments, except for fine-tuning SciBERT, and the second-best score for BioBERT.

Model	Metrics	INS	ISNS	ENS (θ = 0.3)	ENS (θ = 0.5)	ENS (θ = 0.9)	Vanilla	WeLT(ours)
BioBERT	P	**93.39**	92.59	92.56	93.09	92.79	92.71	92.83
	R	90.64	92.94	92.72	92.88	93.31	92.92	**93.63**
	F1	91.99	92.77	92.64	92.99	93.05	92.82	**93.23**
PubMedBERT	P	**93.39**	92.59	92.32	92.83	92.79	79.68	93.31
	R	90.64	**92.94**	89.74	90.99	89.93	80.10	91.19
	F1	91.99	**92.77**	91.01	91.90	91.34	79.89	92.24
BlueBERT	P	**89.45**	88.10	86.72	86.28	86.98	86.28	87.68
	R	73.53	79.62	80.94	**81.65**	81.16	**81.65**	80.68
	F1	80.71	83.65	83.73	83.90	83.97	83.90	**84.04**
SciBERT	P	87.95	89.28	88.82	**90.03**	89.51	87.01	89.22
	R	79.62	81.89	82.95	83.93	83.67	**84.62**	83.06
	F1	83.58	85.43	85.78	**86.88**	86.49	85.80	86.03
BioELECTRA	P	**95.11**	94.66	94.28	94.00	94.28	94.00	94.07
	R	91.92	93.92	94.33	94.26	94.33	94.26	**94.65**
	F1	93.49	94.29	94.30	94.13	94.30	94.13	**94.36**

Table 3.6: The BC5CDR-Chemical fine-tuning scores comparing WeLT with three weighting schemes and the vanilla fine-tuning approach. Precision (P), Recall (R) and F1-score (F1) are the evaluation metrics. The best scores are shown in bold, and the second-best ones are underlined.

Model	Metrics	INS	ISNS	ENS (θ = 0.3)	ENS (θ = 0.5)	ENS (θ = 0.9)	Vanilla	WeLT(ours)
BioBERT	P	85.13	84.14	85.14	84.70	84.91	84.70	**85.61**
	R	85.03	**86.25**	86.18	86.12	85.37	86.12	85.53
	F1	85.08	85.18	**85.66**	85.40	85.14	85.40	85.57
PubMedBERT	P	78.05	79.05	80.22	79.32	79.99	79.32	80.67
	R	74.68	77.28	77.41	**78.05**	77.89	**78.05**	77.28
	F1	76.33	78.15	78.79	78.68	78.92	78.68	**78.94**
BlueBERT	P	77.13	77.39	76.97	77.00	75.72	77.00	**78.12**
	R	70.32	75.15	76.76	**77.19**	76.92	**77.19**	76.67
	F1	73.57	76.26	76.86	77.09	76.31	77.09	**77.38**
SciBERT	P	**79.45**	78.50	**79.45**	78.49	79.04	78.49	79.19
	R	69.75	74.95	76.58	76.74	**77.35**	76.74	76.55
	F1	74.28	76.68	77.99	77.60	**78.19**	77.60	77.85
BioELECTRA	P	86.35	86.97	85.83	85.15	85.62	85.15	**87.58**
	R	86.55	87.20	87.81	87.74	**89.01**	87.74	87.68
	F1	86.45	87.08	86.81	86.42	87.28	86.42	**87.63**

Table 3.7: The BC5CDR-Disease fine-tuning scores comparing WeLT with three weighting schemes and the vanilla fine-tuning approach. Precision (P), Recall (R) and F1-score (F1) are the evaluation metrics. The best scores are shown in bold, and the second-best ones are underlined.

Finally, the BC2GM experiments, as presented in Table 3.8, indicate that WeLT achieved the best scores, except when fine-tuning SciBERT, where it achieved the second-best score.

Model	Metrics	INS	ISNS	ENS (θ = 0.3)	ENS (θ = 0.5)	ENS (θ = 0.9)	Vanilla	WeLT(ours)
BioBERT	P	83.09	82.89	82.95	82.62	82.57	82.62	**83.34**
	R	82.43	82.51	83.35	83.44	**83.47**	83.44	83.30
	F1	82.76	82.70	83.15	83.03	83.02	83.03	**83.32**
PubMedBERT	P	**84.72**	84.32	74.27	73.74	73.66	83.47	83.99
	R	83.46	84.60	73.96	74.30	73.80	84.90	**85.48**
	F1	84.08	84.46	74.11	74.02	73.73	84.18	**84.73**
BlueBERT	P	83.66	83.63	83.96	83.92	83.36	83.92	**84.56**
	R	82.67	83.60	84.42	84.14	**84.45**	84.14	83.93
	F1	83.16	83.61	84.19	84.03	83.90	84.03	**84.24**
SciBERT	P	72.31	72.96	73.15	72.55	**73.61**	72.98	73.05
	R	71.60	73.09	75.08	75.35	**75.77**	74.86	75.35
	F1	71.95	73.02	74.10	73.92	**74.68**	73.90	74.18
BioELECTRA	P	**83.89**	83.28	83.34	83.25	83.47	83.25	83.73
	R	83.58	84.77	85.13	84.79	85.05	84.79	**85.29**
	F1	83.74	84.02	84.23	84.01	84.25	84.01	**84.50**

Table 3.8: The BC2GM-Gene fine-tuning scores comparing WeLT with three weighting schemes and the vanilla fine-tuning approach. Precision (P), Recall (R) and F1-score (F1) are the evaluation metrics. The best scores are shown in bold, and the second-best ones are underlined.

We highlight the special patterns in the experimental results and provide insights into the successful and unsuccessful cases related to the performance of WeLT:

- Fine-tuning WeLT on the largest dataset, BC4CHEMD, achieved the second-best score in all experiments except for fine-tuning BioBERT. Based on our observations, ENS variants demonstrated the best performance. We believe that considering the overall class distribution for calculating the re-scaled weights may degrade performance. Further investigations should incorporate data size as an additional factor. Despite the superior performance of ENS variants, we note that adding extra hyperparameters in ENS is problematic and costly due to the unknown appropriate β factor.

- WeLT achieved the best fine-tuning scores for Linnaeus as the highly skewed dataset in all experiments, except for fine-tuned BioBERT. We believe that the calculation of new re-scaled weights positively impacts performance. Other weighting schemes focus solely on the number of class samples, rather than the overall class distribution.

- Fine-tuning BioELECTRA using WeLT resulted in the best scores except for BC4CHEMD. BioELECTRA is a biomedical version of ELECTRA, which employs a more efficient pre-training strategy known as "replaced token detection." Unlike BERT models, which mask out a small subset of tokens, ELECTRA learns from all input tokens. It uses an additional neural network designed to trick the model by replacing random tokens with fake tokens.

- Regarding fine-tuning PubMedBERT as a domain-specific pre-trained language model, WeLT achieved the best score for the following four datasets: Linnaeus, BioRED-Chemical, BC5CDR-Disease, and BC2GM. ENS variants performed the best for the BioRED-Disease dataset. For the other three datasets, WeLT achieved the second-best score. We believe that WeLT and various ENS variants should be considered when fine-tuning PubMedBERT. The same considerations apply to SciBERT experiments.

3.4.6 Error Analysis

Despite the positive impact of addressing class imbalance on overall performance and sequence labelling evaluation, as presented in Table 3.9, we still observed various types of BioNER mismatches during the error analysis.

As a proof-of-concept, we evaluated the tagging quality outputs of each fine-tuning approach on the Linnaeus dataset. We report F1-scores using seqeval with strict mode and FairEval with fair mode. WeLT achieved the best score for fine-tuning BlueBERT, SciBERT, and BioELECTRA models, and the second-best score for BioBERT and PubMedBERT.

Model	Metrics	INS	ISNS	ENS ($\theta = 0.3$)	ENS ($\theta = 0.5$)	ENS ($\theta = 0.9$)	Vanilla	WeLT
BioBERT	Seqeval	85.01	86.88	87.86	88.49	**89.15**	88.49	88.70
	FairEval	86.76	88.37	89.16	89.72	**90.08**	89.72	89.92
PubMedBERT	Seqeval	82.04	84.89	86.79	**88.79**	86.98	84.89	86.45
	FairEval	84.36	86.62	88.17	**89.19**	88.17	86.62	88.18
BlueBERT	Seqeval	65.56	75.42	76.14	76.14	76.14	88.33	**88.73**
	FairEval	66.33	76.25	76.52	76.52	76.52	89.58	**89.74**
SciBERT	Seqeval	61.35	74.16	76.39	75.41	76.63	75.41	**77.40**
	FairEval	63.02	74.83	76.84	76.06	76.85	76.06	**77.55**
BioELECTRA	Seqeval	74.49	78.91	80.20	81.95	82.20	81.95	**83.38**
	FairEval	78.27	81.32	82.21	84.06	84.33	84.06	**85.64**

Table 3.9: Sequence labelling evaluation F1-scores for species entities in Linnaeus using seqeval with strict mode and FairEval with fair mode. The best scores are in bold and the second-best scores are underlined.

The observed mismatches occur due to the following three types of errors:

- type-1: An entity predicted by the NER model but not annotated in the gold-standard datasets. For instance, “S” is detected by BioPLMs as an abbreviation for “Sulphur”, but it was not annotated by human experts in the BC5CDR gold standard dataset,
- type-2: An entity annotated in the gold-standard datasets but not predicted by the NER model. The main issue behind such misclassification is abbreviated entities. For example, “PAN”, an abbreviation for “Peroxyacetyl nitrate”, is not recognised as a chemical entity, and

- type-3: An entity correctly predicted but with overlapping span errors. For example, BioPLMs recognise two chemical entities separately, such as "amphotericin B-" and "sodium deoxycholate", while the gold-standard annotation is "amphotericin B-sodium deoxycholate".

We believe that the first two types of mismatch errors require knowledge enrichment and the use of active learning approaches to address semantic annotation issues. The third type of error can be mitigated by enhancing the post-processing method to better merge tokens that are part of a recognised entity.

3.5 Study Limitations

The scope of this study was limited to investigating WeLT's approach on a single downstream task, BioNER, within a domain-specific context. Additionally, we did not explore the behaviour of WeLT on other downstream tasks, such as entity linking and relation extraction. Furthermore, we focused exclusively on English biomedical medium-sized datasets due to the limited availability of gold-standard datasets. This includes the Biomedical Language Understanding Evaluation benchmark (BLUE) (Peng et al., 2019), which contains only two biomedical and one clinical NER datasets. In addition, WeLT's weighting scheme primarily focuses on class frequencies within the dataset, but it does not specifically address the rarity of tokens, which may affect the handling of infrequent tokens.

Our results do not exhibit strong statistical significance, with the highest F1-score improvement achieved by WeLT being approximately 1.19 % and the lowest improvement around 0.02 %. This level of improvement aligns with trends observed in previous BioNER research (Li et al., 2020; Shi et al., 2022; Archana and Prakash, 2024; Nemoto et al., 2024). For instance, PubMedBERT, a domain-specific PLM trained from scratch, demonstrates a highest F1-score improvement of 1.7 % and a lowest improvement of 0.48 %. Similarly, BioBERT with continual pre-training shows an F1-score improvement of 0.62 %. These comparisons suggest that the observed improvements are within a typical range for state-of-the-art models in this domain.

Notwithstanding these limitations, the study suggests that addressing class imbalance during fine-tuning offers advantages over vanilla fine-tuning. WeLT is a simple yet effective approach with no additional training costs, unlike the *ENS* method, which requires tuning three different β values for fair comparison. We believe WeLT can be easily applied to general domain, highly skewed datasets, as its cost-sensitive fine-tuning approach promotes immediate integration into any tailored information extraction pipeline.

3.6 Impact of Recognised WeLT Entities on BioNEL

Given the limitations of WeLT's study discussed above, we collaborated with Pedro Ruas and Francisco M. Couto during a secondment at the LASIGE Faculty of Sciences, University of Lisbon, to investigate the impact of WeLT-recognised entities on the BioNEL task. We explore the effectiveness of handling class imbalance in BioNER and its impact on BioNEL, using a strict evaluation script that requires both a matching boundary and correct entity type for accurate recognition.

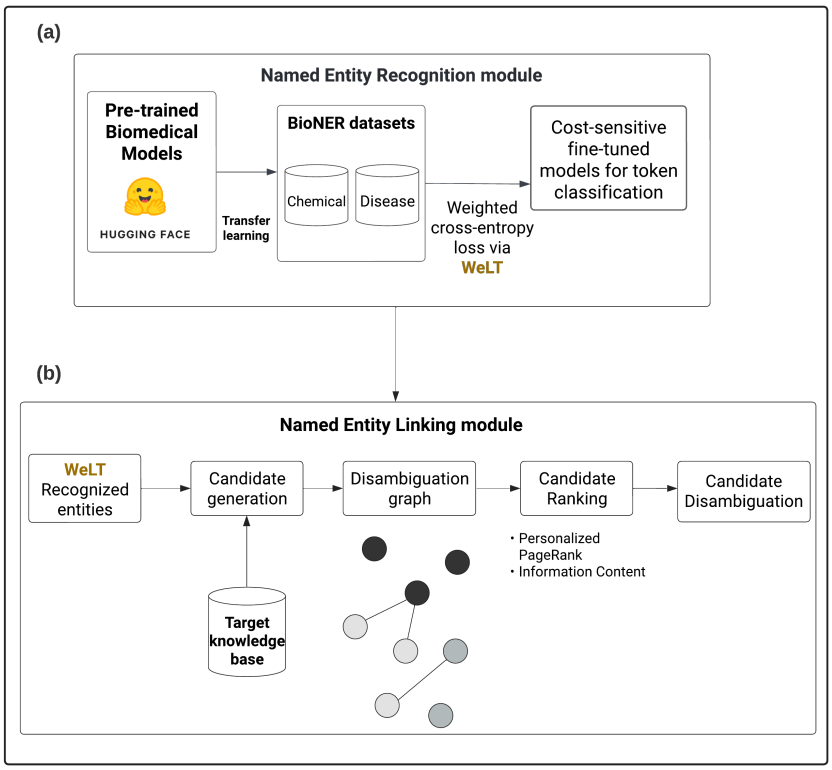

Figure 3.9: Joint BioNER and BioNEL pipeline. Block (a) highlights the WeLT fine-tuning approach, while block (b) shows BioNEL using REEL-NILINKER to link recognised chemical and disease entities.

We investigate the impact of WeLT-recognised entities on BioNEL by evaluating the overall performance of BioNER and BioNEL using the BioCreative evaluation script.[15] We compare WeLT's performance with vanilla fine-tuning approaches. The experimental results demonstrate that WeLT's handling of class imbalance outperforms vanilla models in BioNER. Additionally, the entities recognised by WeLT enhance BioNEL performance in most cases.

[15] BioCreative evaluation script: `https://ftp.ncbi.nlm.nih.gov/pub/lu/BC7-NLM-Chem-track`, last accessed: 01.08.2024.

Beyond overall performance assessment, we evaluate annotation quality for chemical and disease entities. Sequence labelling using seqeval[16] and FairEval[17] indicates that WeLT improves tagging quality.

3.6.1 Utilising WeLT-Recognised Entities for BioNEL

Building on WeLT's success in addressing class imbalance for BioNER (Mobasher et al., 2023), we aimed to extend its impact to BioNEL. Although BioNEL models typically use ground-truth recognised entities, we hypothesise that WeLT's improvements in BioNER can reduce entity misclassification, thereby enhancing BioNEL performance. To explore this, we adopted REEL-NILINKER, proposed by Pedro Ruas and Francisco M. Couto, which partially links biomedical entities to knowledge base concepts such as CTD-Chemical and MEDIC (Ruas and Couto, 2022). Figure 3.9 illustrates the joint extraction pipeline for BioNER and BioNEL using WeLT, adapted from REEL-NILINKER.

3.6.2 Experiments Overview

In this section, we present the ground-truth datasets, preprocessing steps, implementation settings, and evaluation metrics.

3.6.2.1 Evaluation Data

The following gold-standard datasets were used for training and evaluating both the vanilla and WeLT models: BC5CDR (Chemical and Disease), BioRED, and the NCBI Disease corpus. The knowledge bases used are MEDIC (Davis et al., 2020)[18] and CTD-Chemical Vocabulary (Davis et al., 2020)[19]

We conducted fine-tuning experiments using mixed-domain and domain-specific PLMs. Specifically, we fine-tuned BioBERT on NCBI and BC5CDR-Disease, and PubMedBERT (Gu et al., 2022) on BC5CDR-Chemical and BioRED.

3.6.2.2 Baselines

Given that our work investigates the impact of addressing class imbalance before fine-tuning, we do not compete with state-of-the-art BioNER baselines. Instead, we compare the vanilla fine-tuning approach with WeLT.

[16] seqeval evaluation script: https://github.com/chakki-works/seqeval, last accessed: 01.08.2024.
[17] FairEval evaluation script: https://github.com/katrinortmann/FairEval, last accessed: 01.08.2024.
[18] MEDIC version: 2022-06-30.
[19] CTD-Chemical vocabulary version: 2022-06-30.

3.6.2.3 Experimental and Implementation Settings

We used the BioBERT-PyTorch[20] implementation for vanilla fine-tuning and WeLT[21] for the WeLT cost-sensitive fine-tuning approach. For the BioRED dataset, we filtered out chemical and disease entities.

We utilised the official evaluation script of BioCreative VII Track 2 for BioNER (span evaluation) and BioNEL (identifier evaluation). Strict and approximate evaluation models were used for both vanilla and WeLT fine-tuned models. Besides assessing the overall performance of WeLT and traditional models, we also evaluated the annotation quality for chemical and disease entities using seqeval and FairEval metrics adapted for WeLT.

3.6.2.4 Fine-tuning and Hyper-parameter Settings

For a fair comparison, we used identical hyperparameters for fine-tuning BioNER models, as detailed in the Appendix (see Table 9). We fine-tuned BioBERT on both NCBI and BC5CDR-Disease, and PubMedBERT on BC5CDR-Chemical and BioRED (Chemical and Disease).

3.6.3 Results and Discussion

The results of the BioNER evaluation are presented in Table 3.10.

			Strict		**Approx.**	
Type	**Dataset**	**Metrics**	Vanilla	WeLT	Vanilla	WeLT
Disease	BC5CDR	*P*	82.14	**82.45**	91.66	**91.96**
		R	80.84	**81.28**	92.14	**92.78**
		F1	81.49	**81.86**	91.90	**92.37**
	NCBI Disease	*P*	85.10	**85.67**	91.94	**92.53**
		R	88.72	**89.04**	95.74	**95.85**
		F1	86.87	**87.32**	93.80	**94.16**
	BioRED	*P*	84.41	**85.44**	94.06	**94.71**
		R	**86.80**	86.37	95.09	**95.42**
		F1	85.59	**85.90**	94.57	**95.07**
Chemical	BC5CDR	*P*	**93.53**	93.33	**97.11**	96.54
		R	87.62	**89.01**	90.62	**91.68**
		F1	90.48	**91.12**	93.75	**94.04**
	BioRED	*P*	87.93	**88.41**	91.01	**91.72**
		R	90.85	**92.04**	93.90	**95.89**
		F1	89.37	**90.19**	92.43	**93.76**

Table 3.10: Strict and approximate evaluation results of WeLT against the corresponding vanilla fine-tuned trainer for BioNER. The metrics are Precision (P), Recall (R), and F1-score (F1). The best scores are in bold.

[20] BioBERT-PyTorch code: https://github.com/dmis-lab/biobert-pytorch, last accessed: 01.08.2024.
[21] WeLT code: https://github.com/mobashgr/WeLT, last accessed: 01.08.2024.

The models using WeLT marginally surpassed the vanilla trainer, as expressed by the F1-score on five gold-standard datasets for strict and approximate criteria. Our results demonstrate that WeLT achieves higher recall while maintaining high precision for all experiments except for the BioRED disease dataset and the BC5CDR chemical dataset, where there is a degradation in recall performance and precision score, respectively.

We assessed the tagging quality outputs from the vanilla and WeLT models on three datasets as proof of concept, reporting only F1 scores. Additionally, we evaluated the overall performance of BioNER and BioNEL on the previously mentioned five datasets. The F1-score results of sequence labelling evaluation for BC5CDR (chemical and disease entities) and BioRED (chemical entities) are presented in Table 3.11. The results show better sequence labelling quality from WeLT on all three datasets, with the least improvement being 0.59 % and the highest improvement being 1.3 %.

Label	Dataset	Metrics	Vanilla	WeLT
Disease	BC5CDR	*seqeval*	83.04	**84.34**
		FairEval	87.16	**87.45**
Chemical	BC5CDR	*seqeval*	90.53	**91.33**
		FairEval	91.67	**92.35**
	BioRED	*seqeval*	89.93	**90.52**
		FairEval	90.94	**91.71**

Table 3.11: Sequence labelling evaluation F1-score for disease and chemical entities using seqeval with strict mode and FairEval with fair mode. The best scores are in bold.

			Strict		**Approx.**	
Type	**Dataset**	**Metrics**	Vanilla	WeLT	Vanilla	WeLT
Disease	BC5CDR	*P*	74.83	**75.00**	75.75	**75.86**
		R	**77.87**	**77.87**	**80.76**	80.65
		F1	76.32	**76.41**	76.62	**76.66**
	NCBI Disease	*P*	**64.90**	64.31	**67.69**	67.40
		R	72.36	**73.29**	78.24	**78.80**
		F1	68.43	**68.51**	71.03	**71.25**
	BioRED	*P*	71.97	**73.00**	74.54	**76.27**
		R	**72.38**	71.51	**77.63**	76.96
		F1	72.17	**72.25**	74.39	**74.94**
Chemical	BC5CDR	*P*	**86.12**	85.91	86.59	**86.68**
		R	86.48	**87.03**	87.16	**87.96**
		F1	86.30	**86.46**	85.92	**86.40**
	BioRED	*P*	**74.60**	73.47	**72.58**	70.23
		R	**83.33**	81.08	**78.16**	75.52
		F1	**78.72**	77.09	**73.74**	71.36

Table 3.12: Strict and approximate evaluation results of WeLT against the corresponding vanilla fine-tuned trainer for BioNEL. The metrics are Precision (P), Recall (R), and F1-score (F1). The best scores are in bold.

The results of the BioNEL evaluation are presented in Table 3.12. The recognised entities from WeLT slightly improved the BioNEL results for four datasets. For instance, in the BC5CDR document (3323599), part of the recognised disease entities from WeLT were "focal segmental glomerular sclerosis" and "FSGS". This improved the BioNEL by linking them to the correct identifier "MESH: D005923". We also observed that the incorrectly recognised entities by the vanilla fine-tuning approach led to incorrect normalisation. For example, incorrectly recognised entities from the vanilla fine-tuning approach ("glomerular sclerosis" and "FSGS") led to incorrect entity linking to "MESH: D007674" and "MESH: C565831", respectively.

3.6.4 Study Limitations

The scope of this study was limited to evaluating the impact of recognised entities from WeLT and comparing them with the vanilla ones used as input for BioNEL. Our investigation focused only on chemical and disease entities. For the BioNEL task, our collaborators used recent versions of vocabularies that do not fully encompass annotations present in the BC5CDR and NCBI datasets. In addition, our collaborators used the reported version of the MEDIC and CTD-Chemical vocabularies, and older versions are not available.

Despite these limitations, the study suggests that recognised entities from WeLT have a positive impact on BioNEL. Thus, addressing the class imbalance not only enhances BioNER's performance but also demonstrates advantages in BioNEL. We believe that WeLT has been evaluated on various evaluation scripts, including entity-level F1 score adapted from BioBERT using the default seqeval mode. Additionally, the BioCreative's evaluation script uses strict and relaxed modes in which a correct entity has the right span and type. Thus, we suggest that the proposed joint BioNER via WeLT and BioNEL using REEL-NILINKER can be adapted to various domains for both tasks with a class imbalance problem.

3.7 Summary and Discussion

In summary, we addressed the class imbalance challenges in BioNER, including rare entities and data annotation difficulties. Despite advancements in BioPLMs and fine-tuning techniques, we highlighted the limitations of traditional fine-tuning methods, particularly their bias towards majority classes in Section 3.2.3. The detailed trade-offs of vanilla models, such as the use of standard loss functions and the risk of overfitting to majority classes, were identified in Section 3.2.4. In Section 3.3.2, we reviewed different strategies for addressing class imbalance, including data-level, algorithm-level, and hybrid approaches, underlining their respective trade-offs and the complexities of mitigating class imbalance in BioNER using BioPLMs.

Given these trade-offs and the limitations of traditional methods, WeLT offers a novel approach to addressing class imbalance in BioNER by dynamically adjusting class weights based on their complement relative frequency in the dataset.
We believe that WeLT addresses several limitations of traditional approaches, offering a more robust and effective BioNER pipeline for the following reasons:

- WeLT adjusts class weights dynamically based on the normalised complement of each class's relative frequency, allowing the loss function to be more sensitive to minority classes. Unlike data-level approaches, such as oversampling or undersampling, which can lead to overfitting or information loss, WeLT focuses on learning from minority classes without introducing noise or reducing model quality. The method penalises misclassification of majority classes and assigns greater importance to rare ones, effectively reducing bias towards majority classes and enhancing the model's ability to predict instances from minority classes, which represent rare events in real-world scenarios.

- WeLT mitigates overfitting by preventing the model from memorising frequent patterns associated with majority entities. By assigning higher weights to minority classes, it encourages the model to learn discriminative features for rare entities, leading to better generalisation performance on unseen data.

- WeLT enhances the fine-tuning process by incorporating class imbalance directly into the training objective, ensuring the model is trained to prioritise correct classification across both majority and minority classes, resulting in more balanced and accurate predictions.

- Unlike hyperparameter-based approaches that combine multiple techniques, WeLT simplifies the process by directly modifying the fine-tuning procedure. This reduces computational complexity and avoids conflicts between different methods, making it easier to implement and interpret.

- WeLT offers an adaptive and dynamic solution, unlike traditional fixed-weighting schemes or hyperparameters. This adaptability enables WeLT to handle varying degrees of class imbalance across different datasets, domains, and tasks, resulting in improved performance.

In conclusion, WeLT overcomes the limitations of traditional methods by improving model performance, reducing bias, and ensuring equitable learning across all classes. We evaluated WeLT using five different BioPLMs, including general-domain and domain-specific pre-trained models. Our experiments encompassed **280** runs (eight datasets, five BioPLMs, and seven fine-tuning approaches). We comprehensively assessed WeLT's performance compared to other fine-tuning methods and conducted thorough error analyses. While we focused on BioNER, we also explored the impact of WeLT-recognised entities on the BioNEL task, demonstrating the clear

advantages of WeLT over existing weighting schemes and traditional fine-tuning methods.

Although our primary focus has been on addressing class imbalance in the biomedical domain for BioNER and BioNEL tasks, we are inspired to apply WeLT to broader domains, particularly for joint named entity recognition and relation extraction, by tackling the significantly imbalanced negative sampling problem discussed in Chapter 4.

4 Span-based Joint Named Entity and Relation Extraction Using WeLT

Biomedical gold-standard datasets are naturally highly skewed; as discussed in Chapter 3. Thus, vanilla fine-tuned models using BioPLMs are often biased and tend to misclassify named entities. To address this issue, WeLT is proposed as a cost-sensitive fine-tuning approach. We evaluated WeLT's performance on flat BioNER (i.e., where entity spans are assumed non-overlapping, and the entities in the text do not exhibit hierarchical or embedded structures) and BioNEL. The results demonstrate the effectiveness of this approach in addressing the class imbalance problem. Consequently, we were motivated to explore the performance of WeLT on other downstream tasks, such as RE and joint entity and relation extraction (JNERE). Furthermore, we sought to apply WeLT to a broader range of applications beyond the biomedical domain.
As discussed in Section 2.1.3, one of the JNERE models is a span-based approach. Span-based approaches are inherently well-suited for handling overlapping entities (often nested). We highlighted the differences between various types of NER in Section 2.1.1. Here, we briefly recap the concept: overlapping entities occur when two or more named entities share part of the same token span but are not strictly contained within one another.

Example 4.1. *(Overlapping Entities)*
Given the sentence "British Prime Minister Keir Starmer visited London," there are two named entities:

- *"British Prime Minister" as a position entity.*
- *"Keir Starmer" as a person entity.*

Here, "British Prime Minister" represents the position held by "Keir Starmer", resulting in a direct overlap between the title and the individual. This overlap does not imply that either entity is fully contained within the other; rather, both entities coexist within the same span. Thus, "Keir Starmer" is recognised as a specific instance of the title "British Prime Minister."

Conversely, nested entities occur when one named entity is entirely contained within another. Unlike overlapping entities, nested entities exhibit a clear hierarchical relationship, where the inner entity exists within the boundaries of the outer entity.

Example 4.2. *(Nested Entities)*
Given the sentence "The Prime Minister of the United Kingdom, Keir Starmer, spoke today", there are three named entities:

- *"Prime Minister of the United Kingdom" is a position entity.*
- *"United Kingdom" is a location entity.*
- *"Keir Starmer" is a person entity.*

The entity "United Kingdom" is nested within the entity "Prime Minister of the United Kingdom", as the former is entirely contained within the span of the latter.

Since JNERE directly predicts entity spans, it can capture the full extent of overlapping entities (often nested) without relying on sequential labelling schemes like IOB or BILOU tags. By jointly modelling entity spans and relations, span-based approaches are better suited to identifying nested entities (Li et al., 2021) and their relationships by leveraging the contextual information provided by both entities and relations. Therefore, we focus on span-based JNERE models that address overlapping entities (often nested). We argue that JNERE models offer several advantages over traditional separate pipelines due to:

- **End-to-End Coherence**: JNERE pipeline supports seamless end-to-end coherence in information extraction by integrating NER and RE tasks. This approach allows the pipeline to directly extract entities and their corresponding relationships from the text without requiring separate processing stages or intermediate representations. The integration of NER and RE tasks promotes the incorporation of both semantic and syntactic information. Constraints on entity types and relation patterns are jointly enforced during training and inference, enhancing the overall quality of information extraction. As a result, JNERE benefits from shared contextual information, where recognised entities provide valuable context for relation extraction and vice versa.
- **Reduced Error Propagation and Improved Efficiency**: JNERE mitigates error propagation by allowing errors to be corrected jointly, in contrast to separate pipelines, where errors from NER may propagate to RE. For instance, if NER misclassifies entities, RE may extract incorrect relationships based on incorrectly predicted entity boundaries. Additionally, joint learning contributes to computational efficiency gains compared to separate sequential pipelines. The joint model leverages shared computations, potentially reducing inference time and resource consumption.

Span-based Entity and Relation Transformer (SpERT) is one of the state-of-the-art methods in span-based JNERE (Eberts and Ulges, 2020). Despite significant advancements in span-based JNERE,

SpERT introduces an additional imbalance problem by sampling too many negative entities and relations during training. This issue arises due to SpERT's negative sampling strategy, which does not consider the dataset's class distribution.

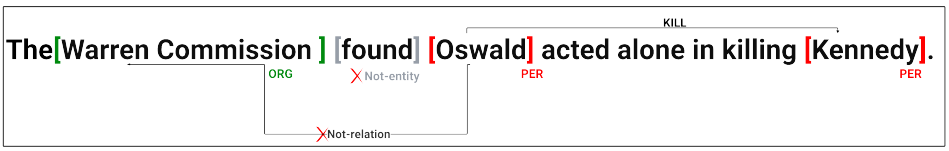

Figure 4.1: An example of span-based joint extraction. "PER" and "ORG" are two pre-defined entity types, and "KILL" is a pre-defined relation type. × Not-entity and × Not-relation denote non-entity and non-relation, respectively. This snippet is part of the CoNLL04 dataset (Roth and Yih, 2004).

Example 4.3. *(Negative samples added by SpERT)*
Figure 4.1 illustrates an example of span-based joint extraction. SpERT recognises the following entities as positive samples:

- *"Warren Commission" is an organisation entity.*
- *"Oswald" is a person entity.*
- *"Kennedy" is a person entity.*

Later in Section 4.2.4, we discuss SpERT's negative sampling strategies in detail. Briefly, SpERT ***adds random negative samples*** *within the sentence as non-entity samples, such as "found". Similarly, for relations, SpERT extracts the relation tuple (Kennedy, KILL, Oswald) and generates non-relation samples, such as (Oswald, Not-relation, Warren Commission).*

Paradoxically, SpERT demonstrates that strong within-sentence negative samples are crucial for model training. However, two distinct issues arise during SpERT's training process:

- **Imbalanced data distribution**: class frequencies within the datasets are skewed, resulting in majority and minority classes. In this study, we evaluate our model on the biomedical ADE dataset and the general domain CoNLL04 corpus. ADE is a relatively balanced dataset, as presented in Table 2.4, whereas CoNLL04 is a skewed dataset, as presented in Table 2.5.
- **Grossly imbalanced negative samples**: SpERT introduces negative samples, including (a) "non-entity" (i.e., none entities) and (b) "non-relation" (i.e., none relations). We elaborate on this issue in Section 4.3.

To address the aforementioned issues, we propose a modified joint training loss function using WeLT to balance the disparity between positive and negative entities and relations. SpERT employs shallow classifiers for both NER and RE tasks (Eberts and Ulges, 2020). By addressing the

imbalanced data distribution and negative sample issues, while retaining critical negative examples, we aim to enhance model performance. Hence, we introduce WeLT-SpERT, which narrows the gap between the class distribution of standard imbalanced datasets and SpERT's negative samples by incorporating a re-scaled joint loss function for entities and relations. Experimental results on the CoNLL04 and ADE datasets demonstrate that WeLT-SpERT variants have marginal improvements over original span-based baselines, with extensive analyses validating the effectiveness of our approach.

In summary, our contributions in this chapter are as follows:

- We present cost-sensitive span and relation classification approaches to address the data distribution gap between positive and negative samples in SpERT, proposing novel joint loss functions using WeLT.

- We investigate several WeLT loss functions to assess their impact on overall performance:
  - Applying only cost-sensitive span classification, referred to as "SpERT-NER".
  - Applying only cost-sensitive relation classification using `weight` and `pos_weight` parameters in the binary cross-entropy loss function, referred to as "SpERT-RE" and "SpERT-pos-weight".
  - Combining both cost-sensitive span and relation classification, referred to as "SpERT-NERE".

- We conduct extensive experiments to evaluate WeLT-SpERT variants on two publicly available datasets, CoNLL04 and ADE. Our results demonstrate the modest performance of WeLT-SpERT variants compared to SpERT for both NER and JNERE tasks.

- We release the code[1] and share the hyperparameters necessary to reproduce our research results.

Structure. In Section 4.1, we provide an overview of related work on JNERE models, with a primary focus on span-based approaches. Subsequently, in Section 4.2, we discuss the SpERT model in detail, outlining the main trade-offs of this approach and identifying research gaps. Building upon this background, we present cost-sensitive SpERT using WeLT in Section 4.3. In Section 4.4, we describe the experimental settings and present the results. Finally, in Section 4.5, we provide a summary and discussion.

[1]WeLT-SpERT code: `https://github.com/mobashgr/WeLT-SpERT`, last accessed: 03.09.2024.

4.1 Related Work

A paradigm shift has occurred in the field of JNERE, moving from token tagging schemes to span classification. Span-based approaches are preferred as they handle overlapping entities (often nested) within a sentence more effectively. Unlike token tagging schemes, which often struggle with these complexities, span-based models better utilise the global features of named entities, including dependencies between entities, and ensure semantic and syntactic coherence between these entities (Bin Ji, 2023). For instance, the relationship between two entities, such as a person's name and their occupation, can provide additional cues to accurately classify both entities and their relations. Typically, span-based JNERE approaches consider spans and span pairs as potential entities and relations, respectively. Consequently, a span-level classification model performs an exhaustive search over all spans, enabling it to manage complex structures where entities and relations may overlap or be nested (Eberts and Ulges, 2020).

Span-based models excel in distinguishing between different levels of nesting and accurately extracting entities within their contexts, leading to more precise extraction of nested entities compared to traditional frameworks. These approaches are effective at handling overlapping entities without the need for complex post-processing steps, unlike sequential labelling schemes (Yu et al., 2022).

Text spans are continuous segments of text, with their length restricted by a threshold ε (Dixit and Al-Onaizan, 2019; Luan et al., 2019). For a sentence with n tokens, let all possible spans be denoted by $s = (a, b)$, where a and b are the indices of the span's start and end tokens, respectively.

Example 4.4. *(Span Enumeration)*
Given the sentence "Breast cancer is treated using chemotherapy.", the span length is expressed as l_s, and $\varepsilon = 3$. We denote all possible spans as follows:

$$spans = \begin{cases} l_s = 1 & \text{"Breast", "cancer", "is", "treated", "using", "chemotherapy", "."} \\ l_s = 2 & \text{"Breast cancer", "cancer is", "is treated", "treated using", "using chemotherapy", "chemotherapy ."} \\ l_s = 3 & \text{"Breast cancer is", "cancer is treated", "is treated using", "treated using chemotherapy", "using chemotherapy."} \end{cases}$$

Thus, for the given text and the ε value, we formulate its spans as follows:

$$s = [t_a, t_{a+1}, \ldots, t_b]$$

$$\text{subject to } 1 \leq a \leq b \leq n \text{ and } (b - a + 1) \leq \varepsilon$$

It is worth mentioning that N-grams are contiguous sequences of n tokens from a given text in which the length of the n-grams is fixed. For instance, a 2-gram (bigram) always consists of exactly two consecutive items. While both N-grams and text spans involve continuous text segments, N-grams

are a specific type of text span with a fixed length, whereas text spans can have varying lengths up to a specified threshold.

Example 4.5. *(Difference between N-grams and text spans)*
Given the sentence "Natural language processing", the 2-grams (bigrams) and the possible text spans for $\varepsilon = 3$ *are:*

- *"Natural language".*
- *"language processing".*
- $l_s = 1$*: "Natural", "language", "processing".*
- $l_s = 2$*: "Natural language", "language processing".*
- $l_s = 3$*: "Natural language processing".*

As illustrated in Example 4.5, N-grams are a specific type of text span with a fixed length, while text spans offer more flexibility in segment length within a defined range.
In Section 2.1.3, we discussed three JNERE paradigms. Here, we focus specifically on span-based approaches. Dixit and Al-Onaizan (2019) introduced the use of PLMs in span-based joint models using embeddings from language models. Zhong and Chen (2021) proposed a span-based joint model using a lighter version of BERT, known as ALBERT. Ji et al. (2020) presented a span-based joint extraction framework with attention-based semantic representations, employing multi-label perception attention to enrich span representations. Yu et al. (2020) introduced a span-based JNERE model by decomposing the problem into multiple labelling tasks. The authors tagged all head entities and then extracted the tail entities and relations, using bidirectional long short-term memory (Bi-LSTM) to predict span boundaries. Shen et al. (2021) proposed the trigger-sense memory flow framework, incorporating a memory module to retain learned category representations in NER and RE tasks. The authors designed a multi-level memory flow attention mechanism to enhance bidirectional interaction between entity recognition and relation extraction. Wei et al. (2021) proposed a Bi-LSTM model that captures bidirectional semantic dependencies by assigning different weights to various parts-of-speech features. In addition, the attention mechanism is used for entity and relation extraction. Ye et al. (2022) introduced packed levitated markers, which consider the interrelation between spans and span pairs by packing markers in the encoder using a neighbourhood-based strategy to model entity boundary information more effectively. Zhu et al. presented a span-based JNERE model with multi-level lexical attention on context features (ER-LAC). ER-LAC uses multi-granularity lexical features to enhance the span semantic representation, employing a transformer classifier to capture internal connections between span pairs and improve relational classification performance.

Despite achieving state-of-the-art results for entity and relation extraction, span-based JNERE models depend heavily on the quality of entity span enumerations. Most models enumerate numerous inaccurate entity spans, known as negative samples, which lead to severe class imbalance problems and high computational complexity. This often results in significant false-positive errors during inference.

Given a sentence with n tokens, the total number of possible spans, denoted as ps, is given by: $ps = \frac{\varepsilon(2n - \varepsilon + 1)}{2}$.

For a span i ($1 \leq i \leq ps$), its start and end tokens are indexed by $start(i) = a$ and $end(i) = b$, respectively, where $1 \leq a \leq b \leq n$. Additionally, the span length l_s must satisfy the constraint $(b - a + 1) \leq \varepsilon$. For example, if a sentence has 50 tokens and the span length threshold $\varepsilon = 10$, then a total of 455 spans will be generated. Let's assume that this sentence has two positive entity samples and the maximum random negative entity samples are set to 100. In that case, there are 100 negative samples, i.e., $min(453, 100)$. Consequently, the pairwise combination of spans generates 4950 negative relation samples, highlighting the severe disparity between positive and negative samples. Previous span-based models have not adequately addressed the impact of negative samples on model performance.

Few studies have attempted to address the imbalance problem in span-based JNERE models:

- Ji et al. (2020) adopted a sampling strategy that assigns a higher weight to the binary cross-entropy loss of relation classification and a lower weight to the cross-entropy loss of span classification. They used fixed scaling factors of 0.4 for span classification and 0.6 for relation classification to maintain a more balanced data distribution.

- Tang et al. (2022) proposed a boundary assembling model to address the imbalance caused by numerous inaccurate entity spans. The proposed model integrates boundary detection, span classification, and relation extraction into an end-to-end framework, demonstrating superior performance compared to state-of-the-art models. They used a weighted sum objective with hyperparameters in the range [0, 1] for these tasks.

- Bin Ji (2023) introduced a two-phase paradigm that classifies entities and relations in the first phase and predicts their types in the second phase. They enhanced this model with global features, combining entity types and distances to reduce the gap between negative entities and pre-defined ones.

To the best of our knowledge, no model that uses weight re-scaling without hyperparameter tuning has yet been developed to address the class imbalance problems in span-based JNER.

4.2 SpERT Approach

In this section, we present the SpERT model, which comprises the three following modules as illustrated in Figure 4.2: *span classification*, *span filtering*, and *relation classification*. Given an input sentence, the fine-tuned BERT generates token and contextualised vector representations. The "span classification" layer enumerates all possible spans by combining the outputs from fine-tuned BERT and applying width constraints to characterise these spans. Subsequently, the spans are classified and filtered, preparing them for the final layer, which performs relation classification and completes the JNERE task. To train the classifier efficiently, SpERT uses negative samples and employs a joint loss function, as further explained in the following sections.

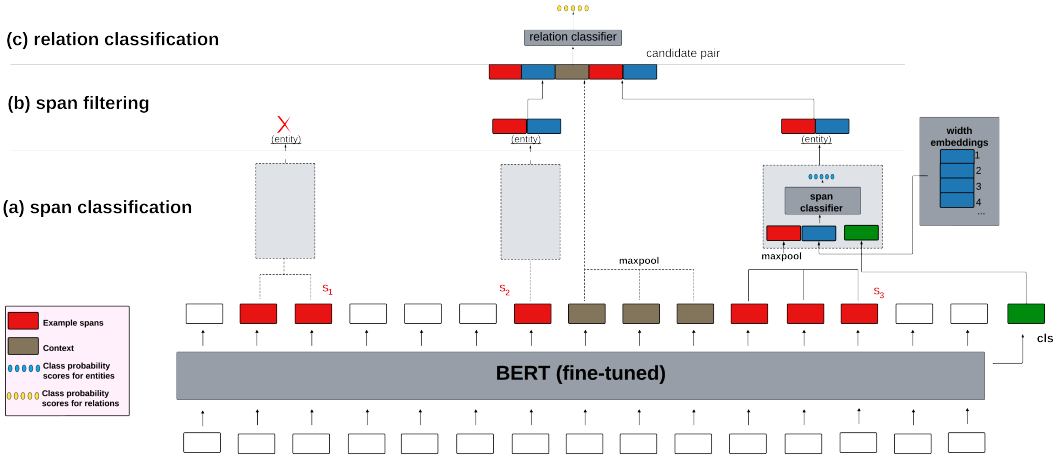

Figure 4.2: Architecture of SpERT (Eberts and Ulges, 2020). SpERT processes a token sequence through fine-tuned BERT and performs the JNERE task in three stages: (a) spans within the sentence are classified into entity types (shown for three samples in the red block: s_1, s_2, and s_3), (b) non-entity spans added due to negative sampling are filtered (e.g., s_1), and (c) entity pairs (e.g., s_2 and s_3) are combined with their context (brown blocks) and classified into relations.

Figure 4.2 provides a bottom-up view of the SpERT model. The bottom layer shows the vector representation layer that uses fine-tuned BERT to extract contextual information (as illustrated in the brown blocks). The input sentence undergoes byte pair encoding (BPE) (Sennrich et al., 2016), splitting it into a sequence of n tokens. BPE encoding breaks uncommon words (e.g., "Surprisingly") into common subwords (e.g., "Surprising" and "ly"), thus constraining the vocabulary and effectively handling out-of-vocabulary (OOV) and rare words.

The BPE tokens are processed through BERT, resulting in a sequence of embeddings ES of length $n + 1$:

$$ES := [e_1, e_2, \ldots, e_n, e_{[CLS]}]$$

where $e_{[CLS]}$ is the special classifier token representing the overall sentence context. Each embedding vector $e_j \in \mathbb{R}^{d_l}$, where d_l denotes the embedding dimension.

4.2.1 Span Classification

SpERT's span classifier takes a candidate span as input. Let $s := (e_j, e_{j+1}, \ldots, e_{j+k})$ denote a span, where e_j specifies the j-th token embedding. The span classifier maps the span s to the predefined set of entity types $\mathcal{E}$. It maps the span s to a class from $\mathcal{E} \cup \{none\}$, where *none* denotes spans that do not encompass entities. The model detects entities among all token subsequences or spans within the input sentence. Each candidate span is classified into an entity type or labelled as "none" if it does not constitute an entity.

SpERT's span classifier consists of three parts, as shown in "step (a)" in Figure 4.2:

1. The span classifier takes the BERT embeddings of the span and combines them using a fusion function f via max-pooling:

   $$f(e_j, e_{j+1}, \ldots, e_{j+k})$$

   This aggregates the representations of individual tokens in a sequence to produce a single contextualised sentence representation.

2. For each span width $k + 1$, there is a dedicated width embedding w_{k+1} lookup, as shown in Figure 4.2 (illustrated in the blue embedding matrix). The width embedding encodes information about the length of the span being considered. For instance, the span width of a person's name might differ from that of an organisation's name. Including width embeddings helps the model capture characteristics and learn to associate certain span widths with specific entity types. Moreover, width embeddings provide prior knowledge about the likelihood of different span widths representing valid entities. For example, very long spans are less likely to represent entities and may indicate noise or irrelevant text. By incorporating this prior knowledge, SpERT can make more informed decisions during span classification.

   Equation 4.1 describes the span representation:

   $$e(s) := f(e_j, e_{j+1}, \ldots, e_{j+k}) \circ w_{k+1} \tag{4.1}$$

   where $e(s)$ specifies the embedding of span s, f denotes the fusion function that combines embeddings $(e_j, e_{j+1}, \ldots, e_{j+k})$ of tokens within the span, w_{k+1} denotes the width embedding obtained from a dedicated embedding matrix for span width $k + 1$, and $\circ$ denotes concatenation.

3. The classifier token in Figure 4.2 (illustrated in the green block) represents the overall context of the sentence. For instance, contextual keywords such as "passing by" or "heading to" can strongly indicate the entity class "location". Thus, the final input to the span classifier is:

$$x^s := e(s) \circ e_{[CLS]} \tag{4.2}$$

where x^s specifies the final input to the span classifier, $e(s)$ denotes the span representation obtained from the fusion of token embeddings within the span and the width embedding, and $e_{[CLS]}$ denotes the classifier token representing the overall sentence context.

The final input is fed into a Softmax classifier to predict the posterior probability distribution over each entity class (including "none") as defined in Equation 4.3:

$$\hat{y}^s := \sigma(W^s \cdot x^s + b^s) \tag{4.3}$$

where:

- $\hat{y}^s$ specifies the predicted posterior probability distribution over each entity class (including the "none" class) for the span s,
- σ is the Softmax activation function that computes the probabilities of each class, ensuring that the probabilities sum to 1,
- W^s expresses the weight matrix of the Softmax classifier,
- x^s denotes the input vector representing the span obtained from the concatenation of the span representation $e(s)$ as specified in Equation 4.2 and the classifier token $e_{[CLS]}$,
- b^s specifies the bias vector of the Softmax classifier.

4.2.2 Span Filtering

"Step (b)" in Figure 4.2 illustrates the span filtering process. By examining the highest-scored class, the span classifier's output (as specified in Equation 4.3) determines which class each span belongs to. The spans assigned to the "none" class are filtered out, resulting in a set of spans $\mathcal{S}$ that are considered entities belonging to the set of predefined categories $\mathcal{E}$. The authors stated that spans longer than ten tokens are pre-filtered to limit the computational cost of span classification (Eberts and Ulges, 2020).

4.2.3 Relation Classification

Finally, once the "none" entities are filtered out, the relation classifier processes each candidate pair. For instance, this applies to "s_2" and "s_3" in Figure 4.2.

The input to SpERT's relation classifier consists of two components:

1. The two entity candidates (e.g., s_1 and s_2) after being fused with BERT embeddings using Equation 4.1. For example, this results in $e(s_1)$ and $e(s_2)$.

2. The localised context between the two entity candidates. In other words, the span ranges from the end of the first entity to the start of the second one, as depicted by the "brown blocks" in Figure 4.2. The BERT embeddings in this context are combined by max-pooling to obtain the context representation. For instance, $c(s_1, s_2)$ denotes the context representation. The authors set $c(s_1, s_2) = 0$ in cases where s_1 and s_2 overlap.

Both input representations are concatenated and passed through a single-layer classifier, which outputs scores indicating the likelihood of each relation being present between the two entities. Since relations may be asymmetric, both (s_1, s_2) and (s_2, s_1) pairs need to be classified. Thus, two input representations x_1^r and x_2^r are generated:

$$x_1^r := e(s_1) \circ c(s_1, s_2) \circ e(s_2)$$
$$x_2^r := e(s_2) \circ c(s_1, s_2) \circ e(s_1)$$

Both x_1^r and x_2^r are passed through a single-layer classifier:

$$\hat{y}_{1/2}^r := \gamma\left(W^r \cdot x_{1/2}^r + b^r\right) \tag{4.4}$$

where:

- γ denotes a sigmoid function. A high response in the sigmoid layer indicates that the corresponding relation holds between s_1 and s_2, and

- W^r is the weight matrix for the relation classification layer, and b^r is the bias term for the relation classification layer.

Given a confidence threshold α, any relation with a score $\geq \alpha$ is considered activated. The authors set the relation filtering threshold to 0.4 (Eberts and Ulges, 2020). If no relation is activated, the sentence is assumed to express no known relation between the two entities.

4.2.4 Negative Sampling Strategy

Negative sampling is vital for providing the model with both positive and negative examples during training. This enables the model to learn to distinguish between true entities and relations versus non-entities and non-relations effectively, thus reducing false positives.
SpERT's negative sampling is performed on each sentence d_i in the training dataset in a single BERT pass. The authors set a fixed number of negative samples randomly from sentence d_i to be labelled as "none". The negative samples are combined with positive ones existing in the dataset td, including (a) candidate spans and (b) candidate entity pairs. Let n_e represent a fixed number of random non-entity spans as negative samples, and let n_r denote the negative relation samples from positive entity pairs.
The training samples are applied in learning the span and relation classifiers, and negative samples n_e and n_r are selected as follows:

- For the span classifier: SpERT utilises all labelled entities $\mathcal{S}^{gt}$ as positive entity samples, plus a fixed number n_e of random non-entity spans as negative samples, such as those "found" in Figure 4.1.
- For the relation classifier: SpERT uses ground truth relations as positive samples and picks n_r negative samples from those entity pairs $S^{gt} \times S^{gt}$ that are not labelled with any relation. For instance, (Oswald, Not-relation, Warren Commission) as depicted in Figure 4.1.

Eberts and Ulges found that the optimal value for both n_e and n_r is 100.

4.2.5 Joint Loss for JNERE

SpERT applies a supervised training strategy with sentences annotated with named entities and relations. The joint loss function for entity and relation classification is:

$$\mathcal{L} := \mathcal{L}^s + \mathcal{L}^r \tag{4.5}$$

where:

- $\mathcal{L}^s$ is the loss of the span classifier using the cross-entropy loss function, and
- $\mathcal{L}^r$ is the loss of the relation classifier using the binary cross-entropy loss.

$\mathcal{L}^s$ and $\mathcal{L}^r$ are averaged over each batch's samples. The authors explicitly mentioned that "no class weights are applied". Each sentence is run only once through BERT in a single pass. Hence, multiple positive and negative samples pass through a single shallow linear layer for the entity and relation classifiers respectively, which speeds up the training process.

4.3 Cost-Sensitive SpERT Using WeLT

SpERT is one of the state-of-the-art models for span-based JNERE approaches. Eberts and Ulges proposed a robust negative sampling strategy, as discussed above. The authors set fixed non-entity spans as negative samples in the ADE and CoNLL04 datasets, resulting in the following outcomes:

- Figures 4.4 and 4.6 depict the class distributions and negative samples ("non-entities") in the ADE and CoNLL04 datasets, respectively.
- Similarly, additional negative samples ("non-relations") were added due to SpERT's negative sampling strategy, as shown in Figures 4.5 and 4.8 for the ADE and CoNLL04 datasets, respectively.

These negative samples are crucial for SpERT's training, as evidenced by the ablation studies conducted by the authors. However, we argue that this strong negative sampling strategy negatively impacts the overall performance. To address this issue, we propose a cost-sensitive version of SpERT, named WeLT-SpERT, which utilises WeLT's loss function to balance span and relation classification, as illustrated in Figure 4.3.

We emphasise the key differences in WeLT-SpERT's span and relation classifiers, along with the modified joint loss function.

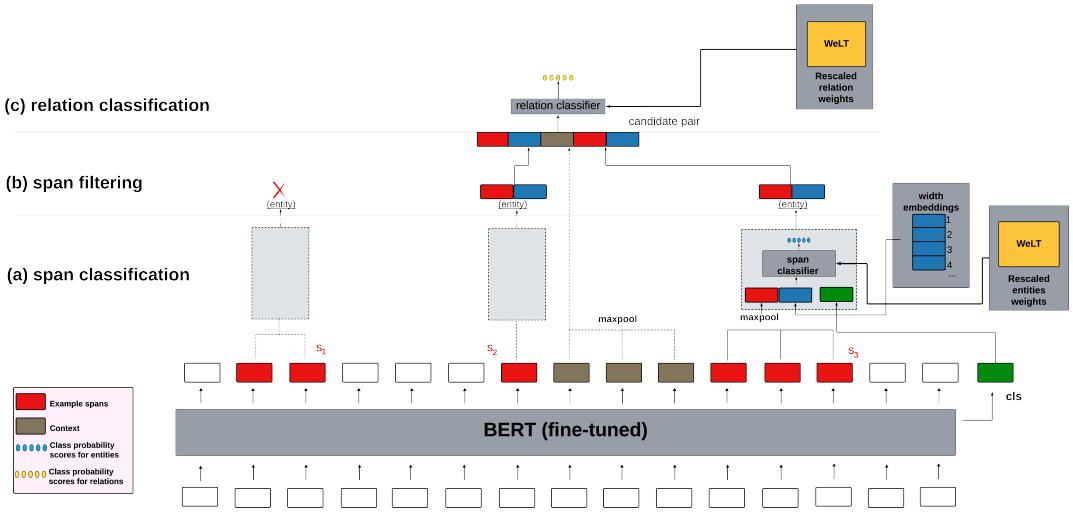

Figure 4.3: Overview of the WeLT-SpERT model for JNERE. The WeLT components represent our contributions to the span and relation classifiers. Image adopted from (Eberts and Ulges, 2020).

4.3.1 WeLT Span Classifier

For the span classification step, the input sentences are tokenised and processed through a fine-tuned BERT model to obtain the contextualised token embeddings, as discussed in Section 4.2. For each

possible subsequence of tokens within the input sequence, a span representation is generated by concatenating the embeddings of the start and end tokens of the span, along with a learned width embedding representing the length of the span in a similar fashion to that defined in Equation 4.1. The span representation is then passed through the span classifier, which outputs a probability distribution over the possible entity classes, including none, as defined in Equations 4.2 and 4.3.

To train the span classifier, SpERT uses a standard cross-entropy loss function. In contrast, SpERT-NER employs a weighted cross-entropy loss to mitigate class imbalance for span classification:

$$\mathcal{L}^{s}_{\text{SpERT-NER}} := -\frac{1}{ns}\sum_{i=1}^{ns}\sum_{j=1}^{c}\sigma\left(1-\frac{n_j}{\sum_{k=1}^{c} n_k}\right) y_{i,j}\log \hat{y}^{s}_{i,j} \tag{4.6}$$

where:

- ns: is the number of spans,
- c: is the number of entity classes and none entities,
- n_j: is the number of instances that belong to class j,
- $\sigma\left(1-\frac{n_j}{\sum_{k=1}^{c} n_k}\right)$: is the rescaled weight for class j using the WeLT approach via the Softmax function σ,
- $y_{i,j}$: is the binary indicator (0 or 1) indicating if class label j is the correct classification for sample i,
- $\hat{y}^{s}_{i}$: is the predicted probability of span i being of class j, as calculated in Equation 4.3.

By incorporating class-specific weights, the WeLT span classifier becomes more sensitive to minority classes and considers the distribution of none entities, improving detection and classification performance for these classes.

4.3.2 WeLT Relation Classifier

Once the entity spans are classified, they are paired to form potential relations. For each pair of spans, a relation representation is constructed by concatenating their respective embeddings. Then, this concatenated representation is fed into one of the WeLT relation classifiers, and final relations are determined based on rescaled weight scores.
To train the relation classifier, SpERT uses a standard binary cross-entropy loss function, while SpERT-RE and SpERT-pos-weight employ a weighted binary cross-entropy loss to address class

imbalance.

First, we demonstrate the calculation of rescaled relation weights denoted by w_r, and then present the two WeLT relation classifiers' losses:

$$w_r := \sum_{j=1}^{r} \frac{n_{\text{head}_j} + n_{\text{tail}_j}}{\sum_{i=1}^{e} n_i} \cdot \frac{nr}{\sum_{j=1}^{r} n_j} \tag{4.7}$$

where:

- r: is the total number of relation classes,
- n_{head_j}: is the number of instances where the entity appears as the head of the relation,
- n_{tail_j}: is the number of instances where the entity appears as the tail of the relation,
- $\cdot$: specifies a multiplication operation,
- $\sum_{i=1}^{e} n_i$: is the total number of entities,
- nr: is the number of instances of the relation class indexed by j, and
- $\sum_{j=1}^{r} n_j$: is the total number of relation instances.

This weight w_r considers both the frequency of entity pairs involved in the relations and the frequency of the relation class itself. The idea is to combine the contribution from the two entities' arguments with the relative frequency of the relation class to derive a comprehensive weight for each relation. For example, to calculate the w_r of a work relationship: if a person works at an organisation, we sum the total frequencies of the person entity and the organisation entity over the total entity frequencies, multiplied by the frequency of the work relationship over the total relation frequencies.

Finally, we normalise the weights w_r for all relation classes so that they sum to 1, maintaining a probabilistic interpretation via the Softmax function σ:

$$\sigma(w_r) := \frac{e^{w_r}}{\sum_{k=1}^{r} e^{w_k}}$$

We present two weighted binary cross-entropy loss[2] using the `pos-weight` and `weight` parameters.

The `pos_weight` parameter applies a weighting factor to the positive class in the loss function, thus adjusting the contribution of the positive class in binary classification. The `weight` parameter

[2]Binary cross-entropy losses: https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html, last accessed: 03.09.2024.

assigns individual weights to each sample.

We first define the WeLT weighted binary cross-entropy loss using `pos_weight` denoted by $\mathcal{L}^r_{\text{SpERT-pos-weight}}$:

$$\mathcal{L}^r_{\text{SpERT-pos-weight}} := -\frac{1}{ns}\sum_{i=1}^{ns}(\, y_i \log\,(\hat{y}^r_i) \cdot \sigma(w_r) + (1 - y_i)\log(1 - \hat{y}^r_i\,)) \tag{4.8}$$

We present the WeLT weighted binary cross-entropy loss using `weight` denoted by $\mathcal{L}^r_{\text{SpERT-RE}}$:

$$\mathcal{L}^r_{\text{SpERT-RE}} := -\frac{1}{ns}\sum_{i=1}^{ns}\sigma(w_r) \cdot (\, y_i \log\,(\hat{y}^r_i) + (1 - y_i)\log(1 - \hat{y}^r_i\,)) \tag{4.9}$$

where:

- ns: number of spans,
- $\sigma(w_r)$: is the normalised rescaled relation weights as calculated in Equation 4.7,
- y_i: is the true label for the i-th sample (1 for positive relation, 0 for negative), and
- $\hat{y}^r_i$: is the predicted probability of the relation for the i-th sample as calculated in Equation 4.4.

4.3.3 WeLT-SpERT's Dummy Example

We provide a step-by-step example to demonstrate the calculation of re-scaled class weights used in the span and relation classifiers. A dummy dataset is employed to illustrate the process. The dummy dataset has four entities denoted by $\mathcal{E} = \{e_1, e_2, e_3, e_4\}$ and three relations denoted by $\mathcal{R} = \{r_1, r_2, r_3\}$. First, we present the steps for the calculations of entity class weights, followed by the relation class weights.

4.3.3.1 Calculating Entity Class Weights

We specify the frequencies of the entities as follows:

- $e_1 = 100$,
- $e_2 = 4940$,

- $e_3 = 333$,
- $e_4 = 500$.

The weight w_{e_j} of each entity $e_j \in \mathcal{E}$ is computed via Equation 4.6, hence:

$$w_{e_j} := 1 - \frac{n_j}{\sum_{k=1}^{c} n_k}$$

The total entity frequency is: $\sum_{k=1}^{c} n_k = 100 + 4940 + 333 + 500 = 5873$. Thus, the complement of each entity's relative frequency is as follows:

$$w_{e_1} = 1 - \frac{100}{5873} \approx 0.9830$$

$$w_{e_2} = 1 - \frac{4940}{5873} \approx 0.1587$$

$$w_{e_3} = 1 - \frac{333}{5873} \approx 0.9433$$

$$w_{e_4} = 1 - \frac{500}{5873} \approx 0.9149$$

Finally, these entity weights are normalised via Softmax. For this, the sum of the exponentiated weights is:

$$\sum_{k=1}^{c} e^{w_{e_k}} \approx 8.9086$$

The normalised class weights are as follows:

$$\sigma(w_{e_1}) = \frac{2.6726}{8.9086} \approx 0.2998$$

$$\sigma(w_{e_2}) = \frac{1.1719}{8.9086} \approx 0.1315$$

$$\sigma(w_{e_3}) = \frac{2.5681}{8.9086} \approx 0.2882$$

$$\sigma(w_{e_4}) = \frac{2.4960}{8.9086} \approx 0.2805$$

4.3.3.2 Calculating Relation Class Weights

We specify the frequencies of the relations as follows:

- $r_1 = 100$,
- $r_2 = 20$,
- $r_3 = 300$.

We present the entity arguments for each relation:

- (e_3, e_4, r_1),
- (e_1, e_2, r_2),
- (e_3, e_3, r_3).

The weight w_r of each relation $r_j \in \mathcal{R}$ is computed via Equation 4.3.2, hence:

$$w_r = \frac{n_{\text{head}} + n_{\text{tail}}}{\sum_{i=1}^{e} n_i} \cdot \frac{n_r}{\sum_{j=1}^{r} n_j}$$

The total relation frequency is: $\sum_{j=1}^{\mathcal{R}} n_j = 100 + 20 + 300 = 420$. Thus, the relation weights are as follows:

$$w_{r_1} = \left(\frac{333 + 500}{5873}\right) \cdot \left(\frac{100}{420}\right) \approx 0.0338$$

$$w_{r_2} = \left(\frac{100 + 4940}{5873}\right) \cdot \left(\frac{20}{420}\right) \approx 0.0409$$

$$w_{r_3} = \left(\frac{333 + 333}{5873}\right) \cdot \left(\frac{300}{420}\right) \approx 0.0810$$

Finally, these relation weights are normalised via Softmax. The sum of the exponentiated weights is:

$$\sum_{k=1}^{r} e^{w_{r_k}} \approx 3.1604$$

The normalised relation weights are as follows:

$$\sigma(w_{r_1}) = \frac{1.0344}{3.1604} \approx 0.3272$$

$$\sigma(w_{r_2}) = \frac{1.0417}{3.1604} \approx 0.3295$$

$$\sigma(w_{r_3}) = \frac{1.0843}{3.1604} \approx 0.3433$$

Using the dummy dataset, we have demonstrated the process of calculating and normalising weights for entities and relations using the Softmax function. These weights are part of the span and relation classifier loss functions, as discussed in the following section.

4.3.4 WeLT Joint Loss Functions

In the previous section, we introduced cost-sensitive WeLT span and relation classifiers. The aim is to examine the effect of (1) balancing entity classes, (2) relation classes, and (3) both combined.
We propose four variations of the WeLT-SpERT models with customised loss functions to handle the class imbalance. These loss functions assign higher weights to minority classes, ensuring better performance across underrepresented categories.
Below, we outline four joint loss functions, each consisting of two core components: one for entity classification and one for relation extraction. Each variant employs its distinct joint loss function, described as follows:

- Variant 1: also known as "SpERT-NER" using the WeLT's span classifier (Equation 4.6) and SpERT's relation classifier (Equation 4.4), the joint loss function is defined as:

$$\mathcal{L}_{\text{SpERT-NER}} := \mathcal{L}^{s}_{\text{SpERT-NER}} + \mathcal{L}^{r} \tag{4.10}$$

- Variant 2: also known as "SpERT-NERE" using the WeLT's span classifier (Equation 4.6) and the WeLT's relation classifier (Equation 4.9), the joint loss function is defined as:

$$\mathcal{L}_{\text{SpERT-NERE}} := \mathcal{L}^{s}_{\text{SpERT-NER}} + \mathcal{L}^{r}_{\text{SpERT-RE}} \tag{4.11}$$

- Variant 3: also known as "SpERT-pos-weight" using SpERT's span classifier (Equation 4.3) and the WeLT's relation classifier with the `pos_weight` parameter (Equation 4.8), the joint loss function is defined as:

$$\mathcal{L}_{\text{SpERT-pos-weight}} := \mathcal{L}^{s} + \mathcal{L}^{r}_{\text{SpERT-pos-weight}} \tag{4.12}$$

- Variant 4: also known as "SpERT-RE" using SpERT's span classifier (Equation 4.3) and the WeLT's relation classifier with the `weight` parameter (Equation 4.9), the joint loss function is defined as:

$$\mathcal{L}_{\text{SpERT-RE}} := \mathcal{L}^{s} + \mathcal{L}^{r}_{\text{SpERT-RE}} \tag{4.13}$$

The WeLT-SpERT incorporates a weighted loss training mechanism that adjusts the loss function based on the frequency of classes, including none samples, thereby ensuring that the minority classes receive more focus during training. Hence, this strategy is absent in the original SpERT model. In the following section, we evaluate the four WeLT-SpERT variants and compare them with the original SpERT model.

To summarise, the primary difference between SpERT and the WeLT-SpERT variants lies in the handling of class imbalance, which is prevalent in real-world datasets for JNERE, including the none entities and relations added by SpERT.

4.4 Evaluating WeLT-SpERT

In this section, we conduct extensive experiments to investigate the impact of mitigating class imbalance using four different WeLT joint loss functions, as presented in Section 4.3.4, and compare them to SpERT. All experiments were performed using a single Tesla P40 GPU with 24 GB of memory. The hyperparameters used are reported in the Appendix (see Table 10), following SpERT's experimental settings for a fair comparison.

The evaluation is conducted on two publicly available datasets: (a) ADE and (b) CoNLL04, as discussed in Section 2.2.3. The statistical class distributions of entities and relations for ADE and CoNLL04 are presented in Table 2.4 and Table 2.5, respectively. The authors used the training and development datasets for training SpERT.[3]

Figure 4.4 depicts the distribution of ADE's two predefined entities and the non-entities added by SpERT. Figure 4.5 shows the frequency of the sole relation in the ADE relations distribution, including the predefined category and the non-relations added by SpERT.

Additionally, Figure 4.6 illustrates the distribution of the four predefined entities in CoNLL04, along with the non-entities added by SpERT. Figure 4.7 illustrates the occurrences of each relation argument type such as "Live_in" relationship with 421 instance that has people and location as entity arguments. Figure 4.8 displays the distribution of the five predefined relations and the non-relations added by SpERT.

In this section, we focus only on evaluating the WeLT-SpERT variants against the original SpERT model. However, in Chapter 7, we provide an extensive comparison to other state-of-the-art models. Hence, our primary baseline in this context is SpERT.

We evaluate the WeLT-SpERT variant models on both entity recognition and relation extraction using the same evaluation strategy as SpERT to ensure a fair comparison:

[3]SpERT code: https://github.com/lavis-nlp/spert?tab=readme-ov-file#reproduction-of-experimental-results, last accessed: 03.09.2024.

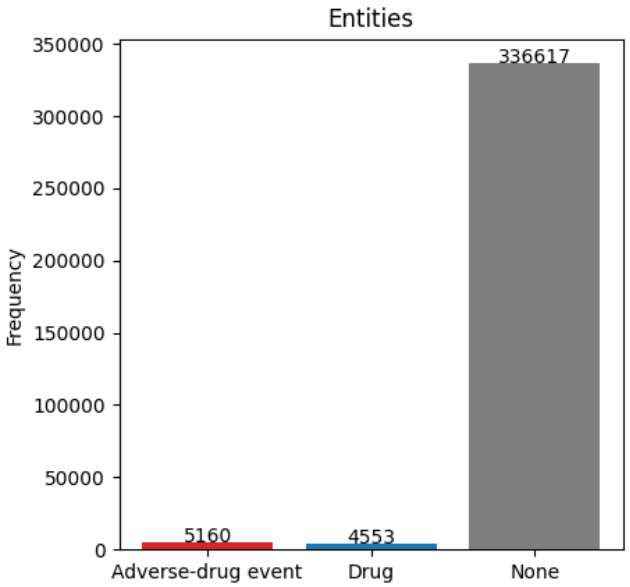

Figure 4.4: Frequency of entities in the ADE training dataset. "Adverse-drug event" and "Drug" are predefined entity types. "None" represents non-entities added by SpERT.

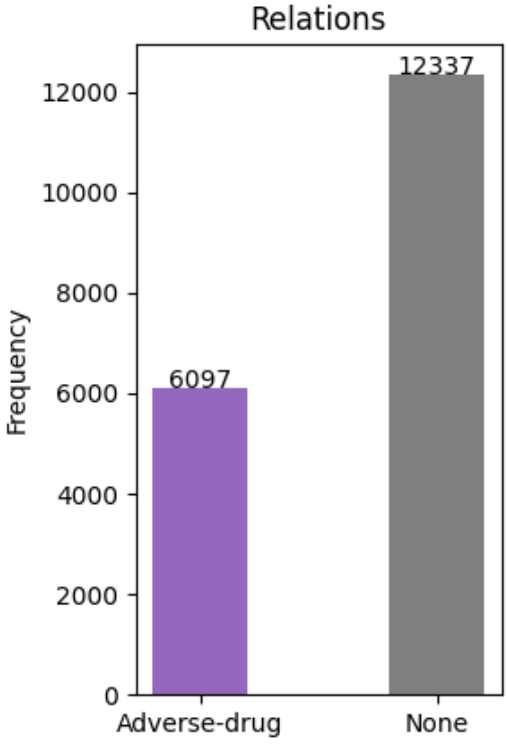

Figure 4.5: Frequency of relations in the ADE training dataset. "Adverse-drug" is the predefined relation type representing the Adverse-drug events. "None" represents non-relations added by SpERT.

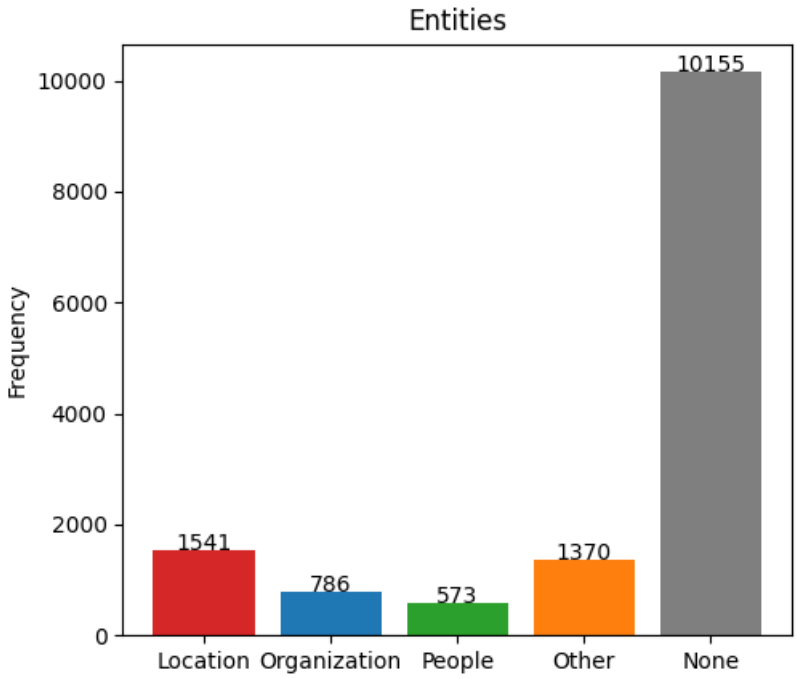

Figure 4.6: Frequency of entities in the CoNLL04 training and development dataset. "Location", "Organization", "People", and "Other" are predefined entity types. "None" represents non-entities added by SpERT.

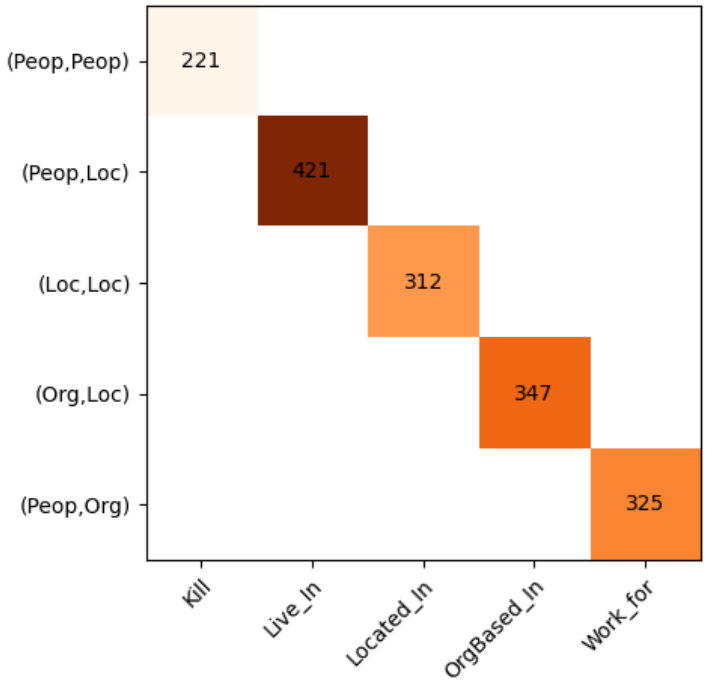

Figure 4.7: Occurrences of each relation and its corresponding entity arguments in CoNLL04.

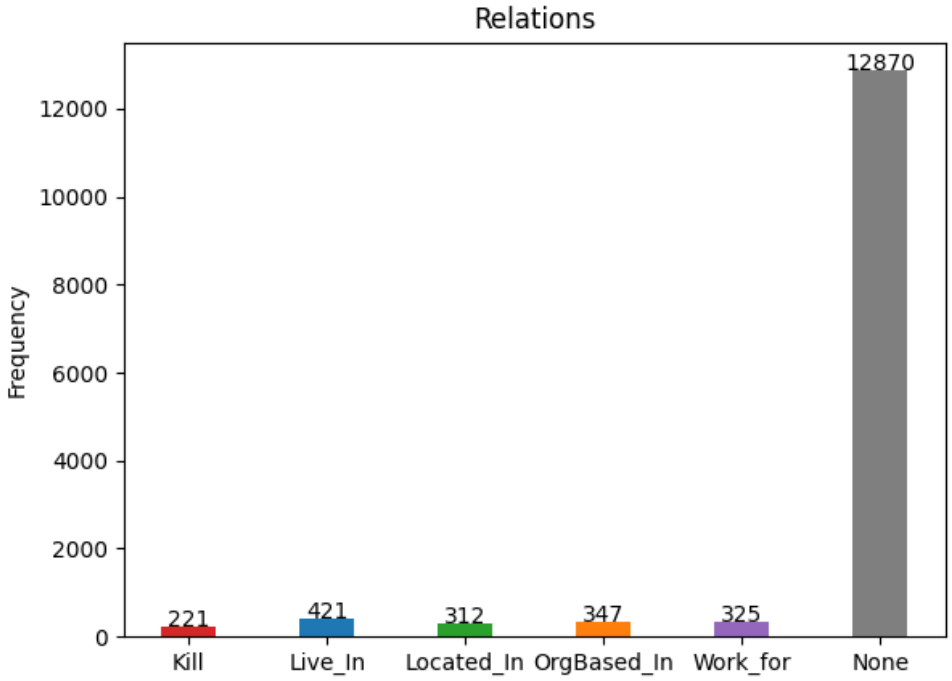

Figure 4.8: Frequency of relations in the CoNLL04 training and development dataset. "Kill", "Live_in", "Located_In", "OrgBased_In", and "Work_for" are predefined relation types. "None" represents non-relations added by SpERT.

- A correct entity is considered only if the predicted span and type match the ground truth. For example, suppose the ground truth contains the entity `(London, Location)` with the span `[0, 6]`. If the model predicts `(London, Organisation)` or `(London, Location)` but with a span of `[1, 7]`, the entity would be considered incorrect. The predicted span `(London, Location)` with the exact span `[0, 6]` is the only valid prediction.

- A correct relation is considered valid if it has the correct type and both associated entities are correctly identified (i.e., following the first criterion). For example, if the ground truth relation is `((London, Location), Located_In, (UK, Location))`, the model must predict the correct relation type `Located_In` and identify both entities `(London, Location)` and `(UK, Location)` with their exact spans and types. If one of the entities is incorrectly identified or has a mismatch in span or type, the predicted relation will not be considered correct.

We report macro-averaged precision, recall, and F1 scores for both the CoNLL04 and ADE datasets, as discussed in Section 2.2.4. For ADE, the F1-score is averaged over 10-fold cross-validation. Additionally, we report micro-averaged precision, recall, and F1 scores for the CoNLL04 dataset, as stated in Section 2.2.2.

4.4.1 Evaluation Results

We report the average results over five runs for each dataset. Table 4.1 presents the macro-average scores for both datasets. This table also encompasses two types of evaluations related to the ADE dataset, one version containing 120 instances of relations with overlapping entities and another version without overlapping entities.

		Entity			Relation		
Dataset	Model	Precision	Recall	F1	Precision	Recall	F1
CoNLL04	SpERT Eberts and Ulges (2020)	85.78	86.84	86.25	74.75	71.52	72.87
	SpERT-NER	84.92	87.10	85.95	71.19	73.02	71.89
	SpERT-NERE	86.07	**89.46**	**87.70**	64.26	**74.44**	68.77
	SpERT-pos-weight	85.37	87.65	86.49	**76.65**	70.99	**73.56**
	SpERT-RE	**87.49**	87.16	87.28	72.79	72.87	72.72
ADE	SpERT Eberts and Ulges (2020)	88.99	89.59	89.28	77.77	79.96	78.84
	SpERT-NER	90.11	**93.53**	91.79	80.75	**88.67**	84.53
	SpERT-NERE	**92.13**	92.60	**92.37**	**83.82**	86.60	**85.19**
	SpERT-pos-weight	91.18	92.70	91.93	82.00	86.19	84.04
	SpERT-RE	90.50	93.22	91.84	81.28	87.57	84.31
ADE (without overlapping)	SpERT Eberts and Ulges (2020)	89.26	89.26	89.25	78.09	80.43	79.24
	SpERT-NER	92.15	92.26	92.20	83.49	85.61	84.54
	SpERT-NERE	91.47	91.92	91.69	83.31	85.90	84.58
	SpERT-pos-weight	**92.28**	92.22	92.25	**84.60**	86.04	85.31
	SpERT-RE	91.99	**92.66**	**92.32**	84.35	**87.45**	**85.87**

Table 4.1: Macro-average F1-scores comparison between SpERT and the proposed WeLT-SpERT variants on the CoNLL04 and ADE datasets. The best scores are shown in bold, and the second-best ones are underlined.

We outline notable patterns observed in the macro-averaged results, as described below:

- **Performance on CoNLL04**:
  - For NER, SpERT-NERE achieves the best recall (89.46 %) and the best F1-score (87.70 %), indicating that it outperforms both the baseline SpERT (86.25 %) and the other WeLT-SpERT variant models in terms of overall performance. SpERT-RE obtains the highest precision (87.49 %) and competitive F1-score (87.28 %).
  - For RE, SpERT-pos-weight achieves the best precision (76.65 %) and F1-score (73.56 %), marginally outperforming SpERT (72.87 %) and SpERT-RE (72.72 %). This shows that the pos-weight variant offers an improvement in relation extraction.

- **Performance on ADE**:
  - For NER, SpERT-NERE performs the best in terms of both precision (92.13 %) and F1-score (92.37 %), showing modest improvement over the baseline SpERT (89.28 %). This indicates the advantages of the NERE variant in handling the ADE dataset.
  - For RE, SpERT-NERE also performs the best in terms of F1-score (85.19 %), with the highest precision (83.82 %), which surpasses SpERT (78.84 %). The model demonstrates a robust capability in relation extraction for this dataset.
- **Performance on ADE (without overlapping)**:
  - For NER, SpERT-pos-weight achieves the highest precision (92.28 %) and an almost identical F1-score (92.25 %) to SpERT-RE (92.32 %), indicating competitive performance across these models, while SpERT lags behind (89.25 %).
  - For RE, SpERT-RE excels with the best precision (84.35 %) and F1-score (85.87 %), surpassing (79.24 %) and the other WeLT-SpERT variants. This shows that SpERT-RE is particularly effective in this specific dataset configuration.
- **Key Observations**:
  - SpERT-NERE performs the best in the CoNLL04 dataset for NER, while SpERT-pos-weight excels in RE.
  - SpERT-NERE is particularly strong for entity extraction in both the ADE and ADE (without overlapping) datasets, showing improvements in precision and F1-scores.
  - SpERT-RE offers strong performance for both entity and relation extraction in the ADE (without overlapping) dataset, achieving the highest scores overall.
  - SpERT, as the baseline, generally performs lower than all the proposed variants, showing that each WeLT-SpERT variant contributes to performance improvement across both entity and relation extraction tasks.

The proposed models (WeLT-SpERT variants) exhibit modest performance compared to the baseline model. Each variant shows particular strengths in different datasets and tasks, with SpERT-NERE excelling in entity extraction and SpERT-RE performing best in relation extraction. In Table 4.2, we present the CoNLL04's micro-average F1 score results.

We identify specific trends related to the micro-averaged outcomes, as outlined below:

- **Performance on Entity Extraction**:
  - SpERT achieves an F1-score of 88.94 %, with a recall of 89.64 %, serving as the baseline.

Dataset	Model	Entity			Relation		
		Precision	Recall	F1	Precision	Recall	F1
CoNLL04	SpERT Eberts and Ulges (2020)	88.25	89.64	88.94	73.04	70.00	71.47
	SpERT-NER	87.22	89.81	88.49	69.20	71.33	70.25
	SpERT-NERE	88.34	**91.29**	**89.79**	63.56	**72.75**	67.85
	SpERT-pos-weight	87.64	89.99	88.80	**75.38**	69.67	**72.41**
	SpERT-RE	**89.59**	90.08	89.83	71.73	71.56	71.65

Table 4.2: Micro-average F1-scores comparison between SpERT and the proposed WeLT-SpERT variants. The best scores are highlighted in bold, while the second-best are underlined.

  - SpERT-NERE demonstrates the highest recall (91.29 %) and the best F1-score (89.79 %), slightly surpassing the baseline and all other proposed variants. This indicates that SpERT-NERE is particularly effective in entity extraction, achieving positive performance overall.
  - SpERT-RE shows the highest precision (89.59 %) and a competitive F1-score (89.83 %), placing it just behind SpERT-NERE in overall entity extraction performance.

- **Performance on Relation Extraction**:
  - SpERT achieves an F1-score of 71.47 %, providing the baseline for comparison.
  - SpERT-pos-weight performs best with an F1-score of 72.41 % and the highest precision (75.38 %). This shows that the pos-weight variant offers modest improvements in relation extraction over SpERT.
  - SpERT-RE also shows positive performance with an F1-score of 71.65 % as the second-best F1 score, achieving a good balance between precision (71.73 %) and recall (71.56 %).

- **Key Observations**:
  - SpERT-NERE outperforms all other models in entity extraction, with the highest recall and F1-score, making it particularly effective in handling this task.
  - SpERT-pos-weight demonstrates the best performance in relation extraction, with the highest precision and F1-score, outperforming SpERT and other variants.
  - The SpERT-RE variant offers a balanced performance across both entity and relation extraction, achieving strong results for both tasks.

Overall, the evaluation of the WeLT-SpERT variants against the baseline SpERT model shows modest improvements in both NER and RE across the CoNLL04 and ADE datasets. The SpERT-NERE model achieves the highest F1 scores for both tasks, underscoring the advantages of addressing the class imbalance. There are marginal improvements in precision and recall, particularly in relation extraction for the ADE dataset, where gains range from 1 % to 3 %.

4.4.2 Error Analysis

We highlight descriptions of incorrect predictions from the SpERT-NERE to delineate future directions for improvements in Table 4.3. Among the predicted results of the proposed model on the CoNLL04 test set, we randomly sampled 100 error instances and categorised them into multiple predefined error categories as follows:

- **Entity recognition errors** occur when our model fails to correctly identify named entities, leading to false positives (i.e., misclassification of a non-entity as an entity) or false negatives (i.e., failure to identify a valid entity).
- **Span-level errors** arise when our model correctly recognises an entity, but the span boundary is incorrect (i.e., the start and end points), resulting in partial recognition errors.
- **Relation extraction errors** occur when our model incorrectly predicts a relationship between identified entities, including false positives (i.e., misclassification of a non-relation as a relation) or false negatives (i.e., missing a valid relationship).
- **Joint training errors** stem from the interaction between both tasks (entity recognition and relation extraction), where errors may propagate. Typically, a misclassified entity or span error can lead to incorrect relation predictions.

We provide an overview of some example error cases from Table 4.3:

- (a) NER misclassification: in this example, our model failed to classify the entity "Judith C.Toth" as a people entity, although it correctly predicted the relationship.
- (b) Incorrect NE span: the model incorrectly identifies "Ernest Tidwell" instead of "G.Ernest Tidwell". This type of span-level error arises from incorrect delineation of multi-token entity boundaries.
- (c) Incorrect relation: the model incorrectly predicts a "Located_In" relation between "Sabine Pass" and "Port Arthur".
- (d) Logical error: the model incorrectly assigns a "Live_In" relation between "Eduard A. Shevardnadze" and "China" instead of the correct relation with "Soviet".
- (e) Lack of syntactic information: the model incorrectly predicts an additional "Live_In" relation between "Gerald Baliles" and "New Hampshire".
- (f) Lack of knowledge: the entity "Organization of the Oppressed on Earth" is not recognised as an organisation entity.

(a) NER misclassification	
Sentence	[Judith C.Toth]$_{PEOP}$ says she returned for a fourth term in [Maryland]$_{LOC}$'s [House of Delegates]$_{ORG}$ because she couldn't find a better job.
Ground-Truth	[Judith C.Toth]$_{PEOP}$ ([House of Delegates]$_{ORG}$,OrgBased_In,[Maryland]$_{LOC}$)
Prediction	[Judith C.Toth] ([House of Delegates]$_{ORG}$,OrgBased_In,[Maryland]$_{LOC}$)
(b) Incorrect NE span	
Sentence	The "poison pill," ruled illegal in November by [U.S.]$_{LOC}$ District [G.Ernest Tidwell]$_{PEOP}$, would become effective after a shareholder had acquired 10 percent of the outstanding stock.
Ground-Truth	([G.Ernest Tidwell]$_{PEOP}$,Live_In,[U.S.]$_{LOC}$)
Prediction	([Ernest Tidwell]$_{PEOP}$,Live_In,[U.S.]$_{LOC}$)
(c) Incorrect relation	
Sentence	[Port Arthur]$_{LOC}$ Mayor [Malcolm Grant]$_{PEOP}$ asked the 800 residents of [Sabine Pass]$_{LOC}$ to evacuate the coastal community just west of the [Louisiana] $_{LOC}$ line, citing the likelihood of high water closing the only highway between the town and [Port Arthur]$_{LOC}$.
Ground-Truth	([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$) ([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$)
Prediction	([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$) ([Sabine Pass]$_{LOC}$,Located_In,[Port Arthur]$_{LOC}$) ([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$)
(d) Logical error	
Sentence	[Soviet]$_{LOC}$ Foreign [Eduard A.Shevardnadze]$_{PEOP}$ is to visit [China]$_{LOC}$ next month to pave the way for the first Chinese - Soviet summit in 30 years, Chinese television reported Monday.
Ground-Truth	([Eduard A.Shevardnadze]$_{PEOP}$,Live_In,[Soviet]$_{LOC}$)
Prediction	([Eduard A.Shevardnadze]$_{PEOP}$,Live_In,[Soviet]$_{LOC}$) ([Eduard A.Shevardnadze]$_{PEOP}$,Live_In,[China]$_{LOC}$)
(e) Lack of syntactic information	
Sentence	"He is the same easy - going, soft - spoken, self - effacing man we knew as governor of [New Hampshire]$_{LOC}$" , said [Virginia]$_{LOC}$ Gov. [Gerald Baliles]$_{PEOP}$, a Democrat.
Ground-Truth	([Gerald Baliles]$_{PEOP}$,Live_In,[Virginia]$_{LOC}$)[New Hampshire]$_{LOC}$
Prediction	([Gerald Baliles]$_{PEOP}$,Live_In,[Virginia]$_{LOC}$) ([Gerald Baliles]$_{PEOP}$,Live_In,[New Hampshire]$_{LOC}$)
(f) Lack of knowledge	
Sentence	Text of the statement issued by the [Organization of the Oppressed on Earth]$_{ORG}$ claiming [U.S.]$_{LOC}$ Marine Lt.[William R.Higgins]$_{PEOP}$ was hanged.
Ground-Truth	[Organization of the Oppressed on Earth]$_{ORG}$ ([William R.Higgins]$_{PEOP}$,Live_In,[U.S.]$_{LOC}$)
Prediction	[Organization of the Oppressed on Earth] ([William R.Higgins]$_{PEOP}$,Live_In,[U.S.]$_{LOC}$)
(g) Propagated error	
Sentence	An art exhibit at the [Hakawati Theatre]$_{ORG}$ in Arab east [Jerusalem]$_{LOC}$ was a series of portraits of Palestinians killed in the rebellion.
Ground-Truth	([Hakawati Theatre]$_{ORG}$,OrgBased_In,[Jerusalem]$_{LOC}$)
Prediction	([Hakawati Theatre]$_{LOC}$,Located_In,[Jerusalem]$_{LOC}$)

Table 4.3: Common error cases of *SpERT-NERE on the CoNLL04's test set*. The red colour expresses error cases.

- (g) Propagated error: the entity "Hakawati Theatre" is misclassified as a location rather than an organisation, leading to an incorrect relation prediction.

In summary, these error cases highlight various challenges in the JNERE task, including span-level accuracy, contextual understanding, syntactic parsing, logical consistency, and the incorporation of domain-specific knowledge. Addressing these issues requires a multifaceted approach that includes improvements in contextual embeddings, syntactic parsing, and knowledge integration to enhance the overall performance of the WeLT-SpERT model.

4.5 Summary and Discussion

The WeLT-SpERT models are cost-sensitive span-based JNERE models designed to address the challenge of overlapping entities. We compared the baseline SpERT model with four WeLT-SpERT variants, focusing on macro-averaged and micro-averaged precision, recall, and F1 scores across the CoNLL04 and ADE datasets, including a non-overlapping ADE variant.
Key patterns observed in the experimental results for both datasets include:

- **CoNLL04 Dataset**: WeLT-SpERT variants demonstrated modest improvements over the baseline SpERT model, except for SpERT-NER. Notable findings regarding F1 scores are:
  - SpERT-NER: rescaling entity classes using the WeLT joint loss function resulted in lower NER and RE performance. For micro-averaged NER, the F1 score was 88.49 %, slightly lower than SpERT. Despite a higher recall (89.81 %), precision dropped to 87.22 %, causing a decline in overall performance. The RE F1 score (70.25 %) also decreased, primarily due to a reduction in precision (69.20 %), although recall increased (71.33 %). The macro-averaged F1 scores followed similar patterns, with lower scores for both NER (85.95 %) and RE (71.89 %).
  - SpERT-RE: rescaling relation classes using the `weight` parameter led to higher F1 scores in most cases. The best micro-averaged NER F1 score was 89.83 %, slightly surpassing SpERT's 88.94 %. RE performance also improved, with an F1 score of 71.65 %. However, macro-averaged scores showed a minor reduction in RE F1 score (72.72 %).
  - SpERT-pos-weight: rescaling relation classes using the `pos_weight` parameter improved RE performance. The micro-averaged NER F1 score was 88.80 %, slightly lower than SpERT, but the RE F1 score (72.41 %) marginally exceeded SpERT's 71.47 %. Macro-averaged scores showed slight improvements in NER (86.49 %) and RE (73.56 %) F1 scores.

  - SpERT-NERE: rescaling both entity and relation classes resulted in mixed outcomes. The micro-averaged NER F1 score (89.79 %) was the second-best, but the RE F1 score (67.85 %) decreased compared to SpERT. Macro-averaged NER F1 (87.70 %) showed the best improvement, while the RE F1 score (68.77 %) declined.

- **ADE Dataset with Overlapping Entities**: observations regarding F1 scores include:
  - SpERT-NER: achieved an improvement in NER F1 score (91.79 %), marginally surpassing SpERT (89.28 %). The RE F1 score was 84.53 %, reflecting an enhancement in relation extraction.
  - SpERT-NERE: exhibited the highest NER F1 score (92.37 %) and RE F1 score of 85.19 %, suggesting that integrating relation extraction enhances overall performance.
  - SpERT-pos-weight and SpERT-RE: both variants achieved competitive NER F1 scores (91.93 % and 91.84 %, respectively), and maintained strong RE performance (84.04 % and 84.31 %, respectively).

- **ADE Dataset without Overlapping Entities**: similar patterns emerged regarding F1 scores:
  - SpERT-NER: achieved an improved NER F1 score (92.20 %), slightly surpassing SpERT (89.25 %). The RE F1 score was 84.54 %, compared to SpERT's 79.24 %.
  - SpERT-NERE: achieved RE F1 score of 84.58 % confirming the benefit of combining entity and relation extraction.
  - SpERT-RE: demonstrated the highest NER F1 score (92.32 %) and the best RE F1 score (85.87 %).
  - SpERT-pos-weight: achieved NER F1 score of 92.25 % and RE F1 score of 85.31 %.

In summary, the WeLT-SpERT variants show modest improvements over the baseline SpERT model, particularly in handling class imbalance in NER and RE tasks. The SpERT-NER variant primarily enhances recall, reflecting improved entity detection. Balancing relation classes using SpERT-RE improves precision, reducing false positives in relation extraction. The `pos_weight` parameter in SpERT-pos-weight leads to balanced gains in both precision and F1 scores for relations. Finally, balancing both classifiers in SpERT-NERE results in the highest overall F1 scores, demonstrating that simultaneous balancing of NER and RE is an effective strategy for improving overall model performance.

5 Attention Weight Mechanism JNERE Using WeLT

SpERT is considered one of the state-of-the-art models for JNERE approaches, as discussed in Chapter 4. However, we addressed its key limitations related to the "non-entities and relations" added through SpERT's strong negative sampling strategy. While the negative sampling approach plays an important role in SpERT's training, we argue that it negatively impacts the overall performance of JNERE. This is primarily due to the class imbalance among predefined categories and the data distribution gap between these categories and non-entities caused by SpERT's sampling strategy. To address this deficiency, we incorporated WeLT's loss function for the span classifier and introduced a novel weighting scheme for the relation classifier, as detailed in Section 4.3.

We are aware that there are various key limitations related to SpERT other than the strong negative sampling strategy, as outlined in Section 4.2.4. Some of these trade-offs are specifically related to the span classifier:

- **Lack of boundary supervision**: SpERT's classifier does not provide explicit boundary supervision for entity spans. Instead, it relies on width embeddings learned through back-propagation to determine span lengths, as highlighted in Section 4.2. This approach can lead to incorrect span extractions that are semantically similar to the correct ones, ultimately degrading performance. For example, if the correct entity is "geometric estimation problem", SpERT might extract both "geometric estimation problem" and "selection of geometric estimation problem", which negatively impacts model accuracy (Jianquan Ouyang, 2022).

- **Sole BERT encoding dependency**: SpERT's span classifier is based on a fully connected layer, which makes it heavily reliant on BERT encodings. Consequently, SpERT's architecture struggles with handling complex datasets that involve relations between diverse entity types. As discussed in Section 2.2.3, for example, SciERC's relations connect pairs of entities that can vary significantly in type, such as methods, tasks, materials, or metrics. This diversity adds complexity to relation extraction, as the model must accurately distinguish between different argument types. Moreover, scientific texts often include specialized terminologies,

further complicating the identification of accurate relationships between different entity types.

To this end, Jianquan Ouyang (2022) proposed a refined version of the SpERT model, titled "Attention and Span-based Entity and Relation Transformer" (ASpERT). ASpERT addressed SpERT's span classifier issues as follows:

- **Attention weight mechanism**: ASpERT introduces attention mechanisms that enhance boundary supervision by utilizing attention weights. ASpERT more accurately determines the start and end of entity spans, addressing the boundary supervision issue that SpERT faces.
- **Enhanced span filtering**: ASpERT's filtering process is refined to better classify spans into predefined entity types or non-entities. The authors utilized a "Multi-layer perceptron" (MLP) to output the probabilities of each entity class, selecting the highest predicted probability.

Typically, ASpERT addresses the main issues of SpERT by providing improved boundary supervision through better attention mechanisms as illustrated in Figure 5.1.

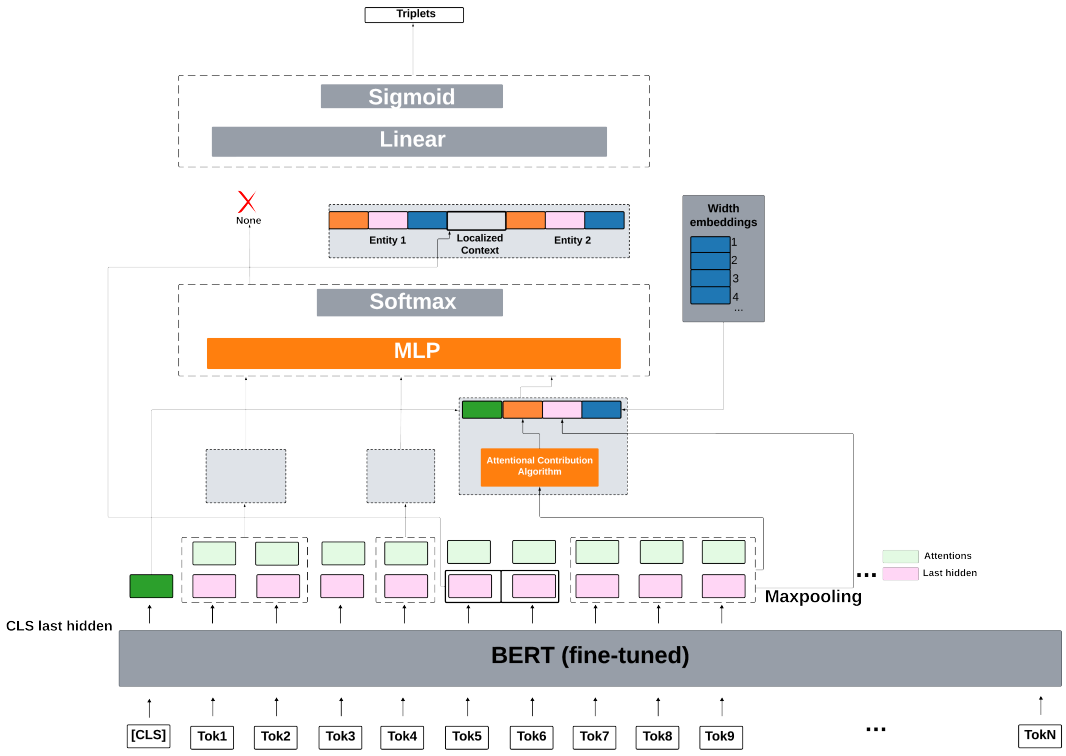

Figure 5.1: Architecture of ASpERT (Jianquan Ouyang, 2022). ASpERT is built upon the SpERT model (Eberts and Ulges, 2020), with key enhancements in the span classifier, as shown in the orange blocks, including (a) the attentional contribution algorithm and (b) the MLP.

ASpERT reduces the dependency on BERT encoding, and enhancing the integration of span and contextual information for more accurate entity and relation extraction. Despite ASpERT's

superior performance, it still adopts the negative sampling strategy and inherits the significantly imbalanced negative samples from SpERT. The data distribution between predefined categories and non-entities and relations remains highly skewed. As stated in Section 4.2.5, random non-entity spans and non-relations are added as negative samples per sentence. Given the modest improvements of WeLT-SpERT variants over the original SpERT, we are motivated to investigate the impact of addressing the class imbalance problem using WeLT, while incorporating ASpERT's approach for JNERE.

To this end, we propose a modified joint training loss using WeLT to balance the gap between positive and negative entities and relations. We present WeLT-ASpERT, which utilizes an enhanced ASpERT span classifier with an attention-weighting mechanism and improved span filtering. Although ASpERT enhances the classification of spans into entity types or non-entities, we believe there is still room for improvement, as the strong negative sampling strategy degrades ASpERT's performance. Experimental results on the CoNLL04 and ADE datasets demonstrate that WeLT-ASpERT models marginally outperform the span-based baselines, including SpERT and ASpERT. We conduct extensive analyses that validate the effectiveness of the proposed approach.

In summary, our contributions in this chapter are as follows:

- We present cost-sensitive attention and span-based entity and relation classification approaches to address the data distribution gap between positive and negative samples in ASpERT, and we propose a novel joint loss function using WeLT.
- We investigate several WeLT loss functions on the overall performance:
  - Applying cost-sensitive span-based attention classification, referred to as "ASpERT-NER".
  - Applying cost-sensitive span-based attention for relation classification using `weight` parameter in the binary cross-entropy loss function, titled "ASpERT-RE".
  - Combining both cost-sensitive span-based attention and relation classification, named "ASpERT-NERE".
- We conduct extensive experiments to evaluate WeLT-ASpERT variants on CoNLL04 and ADE datasets. Our results demonstrate the modest performance of the WeLT-ASpERT variants over SpERT and ASpERT for NER and JNERE tasks.
- We publicly release the code[1] and share hyperparameters to reproduce our research results.

[1]WeLT-ASpERT code: https://github.com/mobashgr/WeLT-ASpERT, last accessed: 06.09.2024.

Structure. In Section 5.1, we provide a detailed illustration of the ASpERT model and discuss its drawbacks. Section 5.2 introduces our cost-sensitive ASpERT models and the proposed WeLT joint loss functions. In Section 5.3, we present the experimental settings and results. Finally, in Section 5.4, we provide a summary and discussion.

5.1 JNERE with Attention Weight Mechanism

ASpERT addresses SpERT's limitations by incorporating advanced techniques for both entity and relation extraction. The core advancements lie in the integration of BERT-based embeddings with a novel approach for span and relation classification.

5.1.1 Span Classifier

ASpERT's span classifier builds upon SpERT's, as discussed in Section 4.2.1, but is designed to improve the identification and classification of entities in a given text. Key innovations include the use of the multi-layer perceptron and the integration of attentional contributions from BERT's attention heads, whereas SpERT relies solely on a linear layer for span classification. ASpERT's span classifier consists of the following components:

- **Token embeddings**: derived from BERT, these embeddings provide contextual information for each token in the input sequence.
- **Max-pooling layer**: applied to the embeddings of tokens within each candidate span to generate a fixed-size span embedding.
- **Width embedding**: encodes the length of the candidate span, providing additional contextual cues.
- **Attentional contribution algorithm**: aggregates attention information across all layers and heads of BERT, enriching the token embeddings with enhanced contextual information.
- **MLP**: replaces the simpler linear classifier used in SpERT, allowing the model to capture more complex patterns in the span embeddings.
- **Softmax layer**: classifies the spans into predefined entity categories.

Following the same strategy as SpERT's span classifier, candidate spans are generated from the token embeddings. Each span s is represented by max-pooling the embeddings of its constituent tokens. For a span s with tokens $[e_j, e_{j+1}, \ldots, e_{j+k}]$, the span embedding $e(s)$ is computed as follows:

$$e(s) := f_l(e_j, e_{j+1}, \ldots, e_{j+k}) \circ w_k \tag{5.1}$$

where:

- $f_l(e_j, e_{j+1}, \ldots, e_{j+k})$ specifies the token embeddings for the span combined using a max-pooling fusion function f_l,
- w_k is the width embedding that encodes the span's length, as illustrated in the blue embedding matrix in Figure 5.1, and
- $\circ$ denotes the concatenation operator.

ASpERT proposes an attentional contribution degree algorithm. The mask score $mask_s$ is defined as a vector whose dimensionality corresponds to the number of tokens in the input sentence d_i. If the sentence contains n tokens, then the mask score can be represented as follows:

$$mask_s \in \mathbb{R}^n$$

where n is the total number of tokens in the sentence. Each entry in the vector $mask_s(t_i)$ corresponds to the score associated with the token t_i in the sentence. Consequently, the dimensionality of the mask score vector reflects the number of tokens, thereby allowing for element-wise operations during subsequent processing in the model.
First, the mask score $mask_s$ for entity and non-entity spans is acquired via Algorithm 5.1. This algorithm calculates the mask score for each entity in the sentence. Entities within the span s are assigned a mask score of $-\infty$, effectively excluding them from further consideration, while entities outside the span are assigned a score of 0.

Algorithm 5.1 Mask score of entity sample

Input: Sentence: d_i ; Entity span: $s := (e_{j+1}, \ldots, e_{j+k})$
Output: The mask score of the entity sample, $mask_s$

```
for all t_i such that t_i ∈ d_i do                ▷ Loop at each token in the sentence
    if t_i ∈ s then                               ▷ Check if the token is part of the entity span
        mask_s(t_i) ← −∞   ▷ Assign −∞ to the mask score if the token is part of the entity span
    else
        mask_s(t_i) ← 0    ▷ Assign 0 to the mask score if the token is not part of the entity span
    end if
end for
    return mask_s                                 ▷ Return the final mask score
```

The time complexity of Algorithm 5.1, which computes the mask score of an entity sample, is $O(n)$, where n is the number of tokens in the input sentence d_i. This complexity arises from the single

loop that iterates through each token in the sentence, performing constant-time operations for each token. The same applies for Algorithm 5.2 that is discussed below.

Algorithm 5.2 then computes the attentional contribution degree $f_a(s)$ for an entity sample based on the attention scores obtained from the BERT model. This algorithm filters the attention head information with low attention to the candidate span based on a contribution threshold Θ. Jianquan Ouyang (2022) set the contribution threshold to 0.5.

The attentional contribution degree $f_a(s)$ is obtained by mean-pooling the attention head information from the token dimensions of context and the entity's token dimension.

Algorithm 5.2 Attentional Contribution Degree

Input: Entity span: $s := (e_{j+1}, \ldots, e_{j+k})$; Mask score of the entity span: $mask_s$; BERT model pre-trained with domain-specific datasets: M_s; Contribution degree threshold: Θ; Mean value from the token dimension of the context and the token dimension of the entity, $MeanPooling$.

Output: The attentional contribution degree, $f_a(s)$;

$A^{s'} \leftarrow M_s$[attentions] ▷ Extract attention weights from BERT model M_s
$A^s \leftarrow A^{s'} + mask_s$ ▷ Add mask score $mask_s$ to attention weights
$A_{temp} \leftarrow A^s$ ▷ Store modified attention weights in A_{temp} for further processing
for all a, b such that $a \in A^s, b \in A_{temp}$ **do** ▷ Iterate over all elements in A^s and A_{temp}
    if $a > \Theta$ **then** ▷ Check if attention value exceeds threshold Θ
        $b \leftarrow 1$ ▷ Set corresponding value in A_{temp} to 1 if threshold is exceeded
    else
        $b \leftarrow 0$ ▷ Set corresponding value in A_{temp} to 0 if threshold is not exceeded
    end if
end for
$A^s \leftarrow A^s \cdot A_{temp}$ ▷ Element-wise multiplication of A^s with updated A_{temp}
$f_a(s) \leftarrow MeanPooling(A^s)$ ▷ Compute attentional contribution degree using mean pooling
return $f_a(s)$ ▷ Return final attentional contribution degree $f_a(s)$

Thus, ASpERT's span representation shows improvements over SpERT's span representation (see Equation 4.1):

$$e(s) := f_l(e_j, e_{j+1}, \ldots, e_{j+k}) \circ f_a(e_j, e_{j+1}, \ldots, e_{j+k}) \circ w_k \tag{5.2}$$

Thus, ASpERT's final input to the novel span classifier is defined as follows:

$$x^s := e(s) \circ e_{[CLS]} \tag{5.3}$$

where:

- x^s specifies the final input to the span classifier,
- $e(s)$ denotes the span representation obtained from the fusion of token embeddings within the span, the width embedding, and the attentional contribution degree, and

- $e_{[CLS]}$ denotes the classifier token representing the overall sentence context.

Finally, the concatenated embedding x^s is passed through an MLP and a Softmax layer for classification. The result of this operation is then passed through the Rectified Linear Unit (ReLU) activation function as follows:

$$y^{s'} := \text{ReLU}(W_1^s \cdot x^s + b_1^s), \tag{5.4}$$

where:

- x^s specifies the concatenated embedding as computed in Equation 5.3,
- this concatenated embedding is multiplied by a weight matrix $W_1^s \in \mathbb{R}^{(2d_l+nw+d_a)\times m}$, and then a bias term $b_1^s \in \mathbb{R}^{(2d_l+nw+d_a)}$ is added to it,
- d_l is the dimension of BERT's last hidden layer,
- nw is the dimension of w_k,
- d_a is the number of BERT's attention heads,
- and m is the number of hidden layer units of the MLP.

Hence, this input is fed into a Softmax classifier:

$$\hat{y^s} := \text{Softmax}(W_2^s \cdot y^{s'} + b_2^s), \tag{5.5}$$

where:

- $\hat{y^s}$ is the entity probability,
- $y^{s'}$ is computed based on Equation 5.4, and multiplied by another weight matrix $W_2^s \in \mathbb{R}^{m\times c}$ originates from the output layer of the model. W_2^s is part of the trainable weight matrices,
- c is the number of entity classes (including none), and
- a bias term $b_2^s \in \mathbb{R}^m$ is added to it.

In summary, ASpERT employs an attentional mechanism to weigh the importance of different tokens in the context of a candidate span. ASpERT addresses the limitations of SpERT by enhancing boundary supervision and employing an attention mechanism to better capture contextual information, resulting in slight performance gains on JNERE, ranging from 0.20 % to 1.39 % across different datasets.

5.1.2 Span Filtering

ASpERT adopts the same filtering scheme for the "none" class as SpERT (see Section 4.2.2), where spans longer than ten tokens are pre-filtered. Additionally, based on the highest-scored class, the output of the novel span classifier (as defined in Equation 5.5) estimates which class each span belongs to. The spans assigned to the "none" class are filtered out, leaving behind a set of spans, denoted as S, that are considered entities belonging to the set $\mathcal{E}$ of predefined categories.

5.1.3 Relation Classification

ASpERT's relation classifier adopts the same approach as SpERT, with additional enhancements incorporating attentional contributions and refined embeddings via Algorithms 5.1 and 5.2. In summary, after filtering out "none" entities, the relation classifier processes each candidate pair (e.g., "Entity 1" and "Entity 2" in Figure 5.1).
The input to ASpERT's relation classifier consists of two components:

1. Two entity candidates, fused with BERT's width embeddings, leveraging attentional contributions and refined embeddings using Equation 5.3. For example, these are represented as $e(s_1)$ and $e(s_2)$.

2. The localised context between the two entity candidates, given the span ranging from the end of "Entity 1" and the beginning of "Entity 2" as shown in the grey block named "Localized Context" in Figure 5.1. If the both entities overlap, this context is empty.

Both input representations are concatenated and passed through a single-layer classifier, which outputs scores indicating the likelihood of a relation between the two entities. Since relations can be asymmetric, both (s_1, s_2) and (s_2, s_1) pairs are classified. As a result, two input representations, x_1^r and x_2^r, are generated as follows:

$$x_1^r := e(s_1) \circ c(s_1, s_2) \circ e(s_2),$$
$$x_2^r := e(s_2) \circ c(s_1, s_2) \circ e(s_1)$$

Both x_1^r and x_2^r are passed through a single-layer classifier:

$$\hat{y}_{1/2}^r := \gamma\left(W^r \cdot x_{1/2}^r + b^r\right), \tag{5.6}$$

where:

- γ denotes the sigmoid function. A high response from the sigmoid layer indicates that a relation holds between s_1 and s_2,

- W^r is the weight matrix for the relation classification layer, and
- b^r is the bias term.

Given a confidence threshold α, any relation with a score $\geq \alpha$ is considered activated. Jianquan Ouyang (2022) set the same relation filtering threshold α as SpERT, which is 0.4. If no relation is activated, the sentence is assumed to express no known relation between the two entities. The input feature vectors for the relation classifier, $x^r_{1/2}$, are constructed by concatenating the representations of the asymmetric relations x^r_1 and x^r_2.

5.1.4 Negative Sampling Strategy

Despite ASpERT's enhancements in boundary supervision through attention weights, which lead to more precise span extraction, and better integration of span and contextual information via the MLP for span classification, ASpERT still follows SpERT's negative sampling strategy. Similar to SpERT, negative sampling is performed on each sentence d_i in the training dataset td. The authors set a fixed number of random negative samples from each sentence d_i, labelled as "none". These negative samples are combined with the positive ones in the dataset, which include candidate spans and candidate entity pairs. The fixed values for n_e (i.e., non-entity spans) and n_r (i.e., negative relation samples) are both set to 150.

5.1.5 ASpERT Loss Functions

ASpERT applies a supervised training strategy on sentences annotated with named entities and relations. The joint loss function for entity and relation classification is defined as:

$$\mathcal{L} := \lambda \mathcal{L}^s + \mathcal{L}^r, \tag{5.7}$$

where:

- λ is the weight for the joint loss function,
- $\mathcal{L}^s$ is the loss of the span classifier $\hat{y}^s$ (as specified in Equation 5.5) using the cross-entropy loss function,
- and $\mathcal{L}^r$ is the loss of the relation classifier $\hat{y}^r_{1/2}$ (as specified in Equation 5.6) using the binary cross-entropy loss function. Both $\mathcal{L}^s$ and $\mathcal{L}^r$ are averaged over each batch's samples.

Jianquan Ouyang (2022) mentioned using a weighted loss function but did not specify the value of λ. However, upon reviewing the ASpERT's public repository,[2] we found that λ is set to

[2] ASpERT code: https://github.com/holire/AspERT, last accessed: 01.08.2024.

0.6. Interestingly, the weighted loss function was not used in the actual implementation as indicated in their code.[3] Therefore, we used the standard joint loss function for both the ASpERT's span and relation classifiers.

5.1.6 Impact of the Novel Span Classifier

Jianquan Ouyang (2022) conducted two ablation studies to assess the benefits of the attentional contribution algorithm (ACD) and evaluate the performance of the proposed novel span classifier. The key findings are as follows:

- **Effects of ACD**: the authors compared the full ASpERT model, which includes the ACD algorithm, with a variant that excluded it. The results showed that the full ASpERT model has a slight improvement compared to the variant without ACD. The ACD algorithm improved relation classification accuracy by enhancing focus on word-to-word relationships, thereby reducing the impact of noisy or irrelevant data on relation predictions. Specifically, ASpERT's F1 score increased by 0.48 % for entity classification and by 1.46 % for relation classification with the inclusion of ACD.

- **Impact of using the MLP for entity extraction**: the authors evaluated two models: the full ASpERT model and a variant where the MLP structure was replaced with a fully connected layer. The results indicated that removing the MLP reduced the model's ability to capture span boundary information, leading to a slight decrease in the F1 score by approximately 0.74 %.

5.2 Cost-sensitive ASpERT using WeLT

Despite the improvements introduced in ASpERT, including enhanced feature embedding by concatenating attention head information from each layer with the final hidden layer using the ACD algorithm, as well as the integration of MLP and Softmax span classification, ASpERT still employs the same negative sampling strategy discussed in Section 5.1.4.

While ASpERT has demonstrated improved performance over SpERT on benchmark datasets (as detailed in Section 5.1.6), we argue that the inclusion of random non-entity spans and relations may degrade overall performance.

Although these negative samples are integral to ASpERT's training, as demonstrated in ablation studies, they contribute to imbalanced data distributions. Specifically, there is an imbalance between positive samples (i.e., predefined categories, which may themselves be imbalanced) and negative

[3] Weighted loss function of ASpERT: https://github.com/holire/AspERT/blob/main/aspert/loss.py#L33, last accessed: 01.08.2024.

samples (i.e., those introduced by ASpERT). To address this, we propose a cost-sensitive version of ASpERT using WeLT's loss function, referred to as "WeLT-ASpERT", to balance the span and relation classification tasks. Hence, WeLT-ASpERT modifies ASpERT to incorporate cost-sensitive learning.

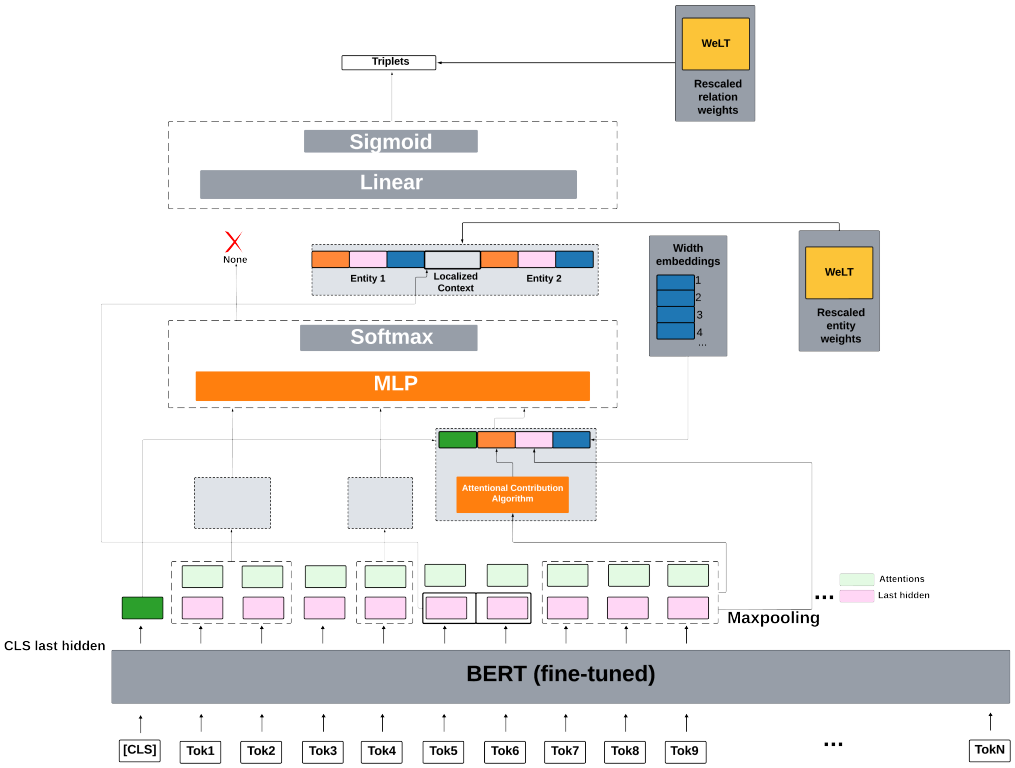

Figure 5.2: Overview of the WeLT-ASpERT model for JNERE. Image adopted from (Jianquan Ouyang, 2022).

In the following sections, we detail the differences in the span and relation classifiers, as well as the changes made to the joint loss function.

5.2.1 WeLT Span Classifier

As shown in Figure 5.2, the input sentences are tokenized and processed through a fine-tuned BERT model to obtain contextualized token embeddings. A span representation is generated for each possible subsequence of tokens within the input sequence. This representation is formed by concatenating the embeddings of the start and end tokens of the span, along with a learned width embedding that encodes the length of the span. Rather than relying solely on the last hidden layer, attention head information from each transformer layer is aggregated. The ACD algorithm is applied to weigh the contributions of each layer. The final feature representation is a weighted sum of these layers, incorporating the attentional contribution scores.

Once the feature embedding is complete, the span classification proceeds through the following steps:

1. **Span Extraction**: potential entity spans are extracted from the input text. Each span is characterized by its start and end positions, which form the basis for subsequent classification.
2. **MLP Structure**: an MLP processes the span features. The MLP consists of several fully connected layers with non-linear activation functions, enabling the learning of complex patterns and interactions within the data.
3. **Softmax Classification**: the output layer of the MLP is a Softmax classifier that assigns probability scores to each possible span label. This probabilistic approach allows the model to handle multiple classes and provides a measure of confidence in its predictions.
4. **Classification Output**: the final output includes the predicted entity types, including the "none" class, and their corresponding spans. This structured output is critical for accurate entity and relation extraction in downstream tasks.

To train the span classifier, ASpERT uses a standard cross-entropy loss function. In contrast, "ASpERT-NER" employs a weighted cross-entropy loss to mitigate class imbalance for span classification:

$$\mathcal{L}^{s}_{\text{ASpERT-NER}} := -\frac{1}{ns}\sum_{i=1}^{ns}\sum_{j=1}^{c}\sigma\left(1-\frac{n_j}{\sum_{k=1}^{c} n_k}\right) y_{i,j}\log \hat{y}^{s}_{i,j} \tag{5.8}$$

where:

- ns: the number of spans,
- c: the number of entity classes (including "none"),
- n_j: the number of instances belonging to class j,
- $\sigma\left(1-\frac{n_j}{\sum_{k=1}^{c} n_k}\right)$: the re-scaled weight for class j using the WeLT approach with σ as a Softmax function,
- $y_{i,j}$: a binary indicator (0 or 1) if class label j is the correct classification for sample i, and
- $\hat{y}^{s}_{i}$: the predicted probability of span i belonging to class j, as calculated in Equation 5.5.

By incorporating class-specific weights, the WeLT span classifier becomes more sensitive to minority classes, particularly in balancing the distribution of "none" entities. This improves detection and classification performance for under-represented classes.

5.2.2 WeLT Relation Classifier

Once the entity spans are classified, they are paired to form potential relations. For each pair of spans, a relation representation is constructed by concatenating their respective embeddings. This concatenated representation is then fed into the WeLT relation classifier, where final relations are determined based on rescaled weight scores.

In contrast to ASpERT, which uses a standard binary cross-entropy loss function for relation classification, "ASpERT-RE" employs a weighted binary cross-entropy loss to address class imbalance. The calculation of rescaled relation weights and the loss function of the WeLT relation classifier are described below.

First, we compute the rescaled relation weights w_r, defined as follows:

$$w_r := \sum_{j=1}^{r} \frac{n_{\text{head}_j} + n_{\text{tail}_j}}{\sum_{i=1}^{e} n_i} \cdot \frac{n_r}{\sum_{j=1}^{r} n_j} \tag{5.9}$$

where:

- r: total number of relation classes,
- n_{head_j}: number of instances where the entity appears as the head of the relation,
- n_{tail_j}: number of instances where the entity appears as the tail of the relation,
- $\cdot$: denotes multiplication,
- $\sum_{i=1}^{e} n_i$: total number of entities,
- n_r: number of instances of relation class r, and
- $\sum_{j=1}^{r} n_j$: total number of relation instances.

The weight w_r accounts for the frequency of entity pairs involved in the relations and the frequency of the relation class itself. By combining the contributions from the head and tail entities with the relative frequency of the relation class, we derive a comprehensive weight for each relation. We normalize these weights w_r using the Softmax function σ to ensure a probabilistic interpretation:

$$\sigma(w_r) := \frac{e^{w_r}}{\sum_{k=1}^{r} e^{w_k}}$$

In Section 4.3.3, we illustrated the calculation of rescaled relation weights using a dummy example. We now present a binary cross-entropy loss function denoted by $\mathcal{L}^{r}_{\text{ASpERT-RE}}$ using `weight`

parameter to scale the overall loss for each sample, allowing for instance-specific adjustments based on rescaled relation weights:

$$\mathcal{L}^{r}_{\text{ASpERT-RE}} := -\frac{1}{ns}\sum_{i=1}^{ns}\sigma(w_r)\cdot(y_i\log(\hat{y}^{r}_{i}) + (1-y_i)\log(1-\hat{y}^{r}_{i})) \tag{5.10}$$

where:

- ns: number of spans,
- $\sigma(w_r)$: normalized rescaled relation weights as calculated in Equation 5.9,
- y_i: true label for the i-th sample (1 for positive relation, 0 for negative), and
- $\hat{y}^{r}_{i}$: predicted probability of the relation for the i-th sample, as calculated in Equation 5.6.

5.2.3 WeLT Joint Loss Functions

In the previous section, we introduced cost-sensitive WeLT span and relation classifiers. The aim is to examine the effect of (1) balancing entity classes, (2) relation classes, and (3) both combined. We propose three variations of the WeLT-ASpERT models with customised loss functions to handle the class imbalance. These loss functions assign higher weights to minority classes, ensuring better performance across underrepresented categories.

Below, we outline three joint loss functions, each consisting of two core components: one for entity classification and one for relation extraction. Each variant employs its distinct joint loss function, described as follows:

- **Variant 1**: also known as "ASpERT-NER" that utilises the WeLT span classifier from Equation 5.8 and ASpERT's relation classifier from Equation 5.6. This joint loss function re-scales named entity classes while maintaining the standard loss function for relation extraction (RE). Thus, the joint loss is formulated as:

$$\mathcal{L}_{\text{ASpERT-NER}} := \mathcal{L}^{s}_{\text{ASpERT-NER}} + \mathcal{L}^{r} \tag{5.11}$$

- **Variant 2**: also known as "ASpERT-RE" that combines ASpERT's span classifier from Equation 5.5 with the WeLT relation classifier using the `weight` parameter from Equation 5.10. Unlike Variant 1, this joint loss function re-scales relation classes only, using the standard loss function for NER. The joint loss is defined as:

$$\mathcal{L}_{\text{ASpERT-RE}} := \mathcal{L}^{s} + \mathcal{L}^{r}_{\text{ASpERT-RE}} \tag{5.12}$$

- **Variant 3**: also known as "ASpERT-NERE" that incorporates both the WeLT span classifier from Equation 5.8 and the WeLT relation classifier from Equation 5.10, re-scaling both entity and relation classes. The joint loss function is formulated as:

$$\mathcal{L}_{\text{ASpERT-NERE}} := \mathcal{L}^{s}_{\text{ASpERT-NER}} + \mathcal{L}^{r}_{\text{ASpERT-RE}} \tag{5.13}$$

In summary, the primary distinction between ASpERT and WeLT-ASpERT models lies in their approach to handling class imbalance. WeLT-ASpERT introduces a weighted loss mechanism that adjusts for class frequencies, including non-entities and relations, aiming to enhance the detection and classification of minority classes compared to the original ASpERT model. Additionally, this approach differs from the earlier work in Chapter 4, as WeLT joint loss functions are applied to ASpERT, which benefits from improved span boundary handling via the ACD algorithm and MLP-Softmax span classification.

5.3 Evaluating WeLT-ASpERT

In this section, we follow the experimental setup outlined in Section 4.4, with a few exceptions discussed below. We fine-tuned the WeLT-ASpERT models to investigate the impact of addressing class imbalance using the three different WeLT joint loss functions presented in Section 5.2.3, and compared them against the SpERT and ASpERT models.

The evaluation was conducted on two publicly available datasets: ADE and CoNLL04, as introduced in Section 2.2.3. The statistical class distributions of entities and relations for the ADE and CoNLL04 datasets are presented in Table 2.4 and Table 2.5, respectively.

For hyperparameters, we used settings reported in the Appendix (see Table 11), following ASpERT's experimental setup to ensure a fair comparison. As previously mentioned, this work differs from the earlier WeLT-SpERT models in Chapter 4. Below, we outline the key distinctions in model training between ASpERT and the previous SpERT approach:

- The authors of ASpERT used different values for n_e and n_r, setting both to 150, while in the SpERT model these values were set to 100. Jianquan Ouyang (2022) employed fixed non-entity spans and non-relations as negative samples in the ADE and CoNLL04 datasets.

- Unlike SpERT, which was trained using both the training and development datasets, ASpERT was trained only on the training dataset. Consequently,

  – Figures 5.3 and 5.5 illustrate the class distributions and non-entity spans added by ASpERT for ADE and CoNLL04, respectively.

– Similarly, SpERT's negative sampling strategy introduces additional non-relations, as depicted in Figures 5.4 and 5.6 for the ADE and CoNLL04 datasets. Figure 5.4 shows the frequency of the sole relation type and the added "none" relations.

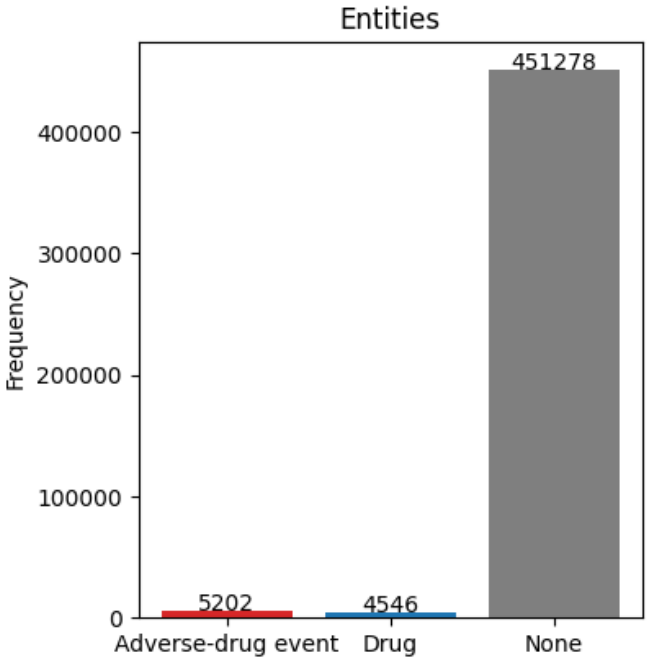

Figure 5.3: Frequency of entities in ADE's training dataset. The "Adverse-drug event" and "Drug" are predefined entity types, while "None" refers to non-entities added by ASpERT.

- In the ASpERT model, the authors increased the batch size, varying it from four to ten. However, due to computational constraints, we retained SpERT's original batch size, setting it to two.
- The ASpERT model introduced configurations for the span classifier's MLP size, MLP dropout, and contribution threshold, which were not applied in the SpERT architecture.
- The ASpERT model was fine-tuned using BioBERT for ADE, whereas SpERT was fine-tuned using BERT.

Figure 5.5 presents the distribution of four predefined entity types along with "none" entities for the CoNLL04 dataset.
Figure 5.6 shows the frequencies of the five pre-defined relations and "none" relations added by ASpERT.
Based on the calculation of relation weights w_r in Equation 5.9, we are interested to know the entity arguments of each relation. For instance, Figure 5.7 illustrates the occurrences of each relation argument type such as "Live_in" relationship with 330 instance that has people and location as entity arguments.

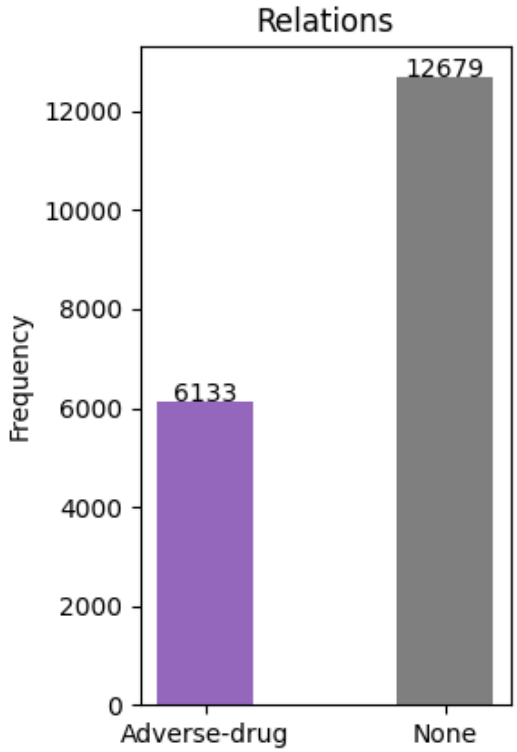

Figure 5.4: Frequency of relations in the ADE training dataset. "Adverse-drug" represents the predefined relation type for adverse drug events, while "None" refers to non-relations added by ASpERT.

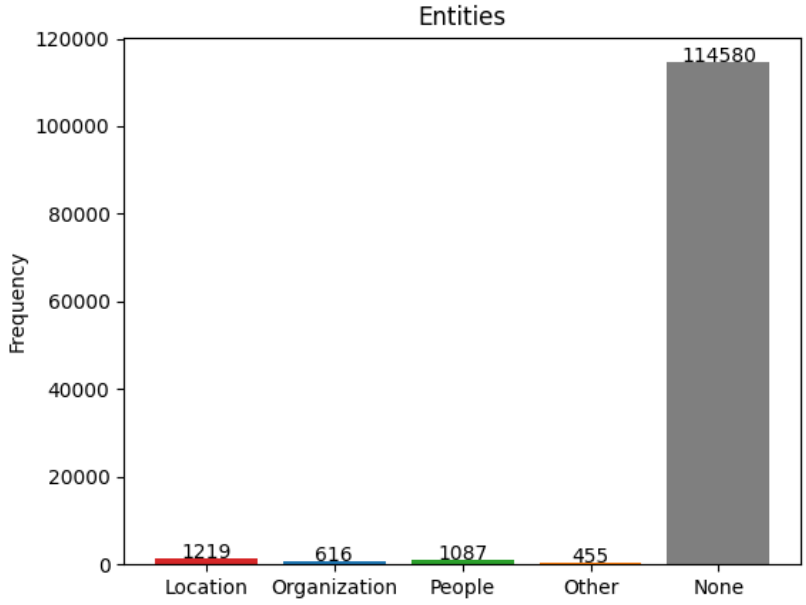

Figure 5.5: Frequency of entities in CoNLL04's training dataset. "Location", "Organization", "People", and "Other" are predefined entity types, while "None" refers to non-entities added by ASpERT.

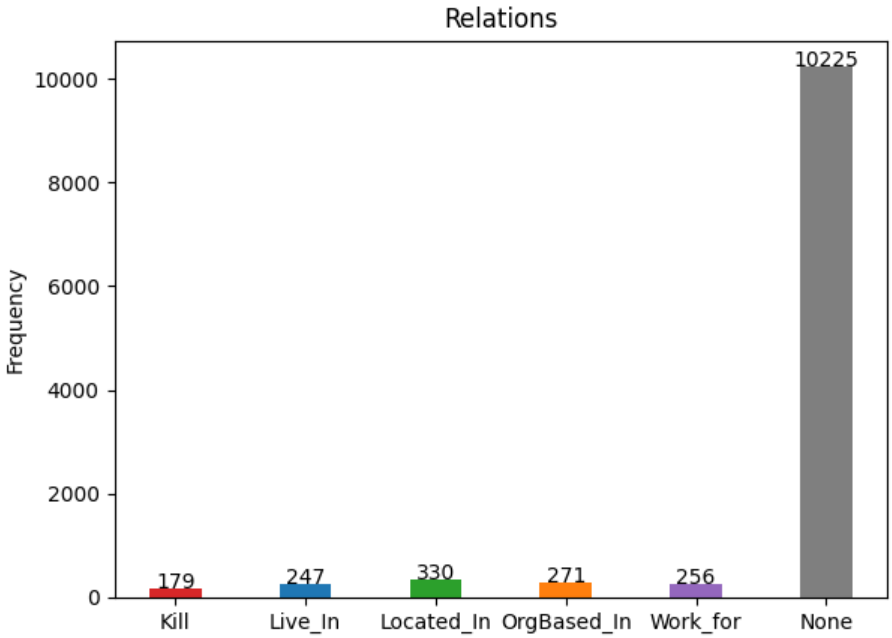

Figure 5.6: Frequency of relations in CoNLL04's training dataset. "Kill", "Live_in", "Located_In", "Org-Based_In", and "Work_for" are predefined relation types, while "None" refers to non-relations added by ASpERT.

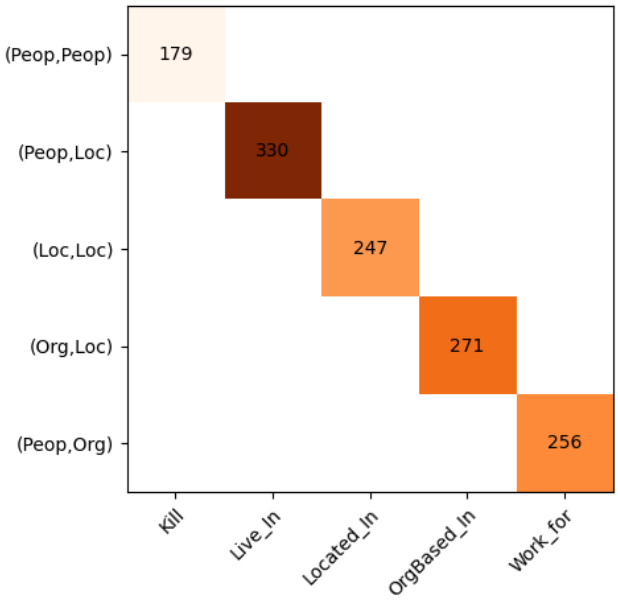

Figure 5.7: Occurrences of each relation and its corresponding entity arguments in CoNLL04.

Due to differences in hyperparameters, Jianquan Ouyang (2022) trained SpERT from scratch and reported different results compared to those in the original SpERT paper (Eberts and Ulges, 2020). We evaluate our WeLT-ASpERT models against both SpERT and ASpERT models.
Since ASpERT employed the same evaluation strategy as SpERT, we adopt this approach to evaluate the WeLT-ASpERT variants:

- An entity prediction is considered correct only if the predicted span and type exactly match the ground truth. For example, if the ground truth contains the entity `(London, Location)` with the span `[0, 6]`, and the model predicts `(London, Organization)` or `(London, Location)` but with a span `[1, 7]`, the prediction is considered incorrect. Only the prediction `(London, Location)` with span `[0, 6]` is valid.

- A relation prediction is correct if it has the correct type and both associated entities are correctly identified. For example, if the ground truth relation is `((London, Location), Located_In, (UK, Location))`, the model must predict the correct relation type `Located_In` and correctly identify both entities `(London, Location)` and `(UK, Location)` with their exact spans and types. If one of the entities is incorrectly identified or has a span/type mismatch, the predicted relation is incorrect.

We report macro-averaged precision, recall, and F1 scores for both the ADE and CoNLL04 datasets, as described in Section 2.2.4. For ADE, the F1 score is averaged over 10-fold cross-validation. Additionally, we report micro-averaged precision, recall, and F1 scores for the CoNLL04 dataset, as stated in Section 2.2.2.

5.3.1 Evaluation Results

		Entity			Relation		
Dataset	Model	Precision	Recall	F1	Precision	Recall	F1
CoNLL04	SpERT (Eberts and Ulges, 2020)	84.75	85.86	85.26	72.11	69.24	70.41
	ASpERT (Jianquan Ouyang, 2022)	**86.57**	85.49	**85.97**	**74.92**	69.01	71.66
	ASpERT-NER	84.18	**86.93**	85.46	71.31	**74.96**	**72.72**
	ASpERT-NERE	83.94	84.03	83.92	68.19	71.61	69.66
	ASpERT-RE	85.15	81.89	83.32	70.89	68.94	69.53
ADE	SpERT (Eberts and Ulges, 2020)	90.10	91.74	90.91	79.70	83.29	81.84
	ASpERT (Jianquan Ouyang, 2022)	90.96	91.87	91.41	**81.65**	83.92	**82.76**
	ASpERT-NER	90.38	91.85	91.11	77.22	83.14	80.34
	ASpERT-NERE	90.38	91.85	91.11	77.72	83.14	80.34
	ASpERT-RE	**91.55**	**92.12**	**91.83**	80.95	**84.01**	82.45

Table 5.1: Macro-average F1-scores comparison between SpERT and ASpERT with the proposed WeLT-ASpERT(⋆) models on the CoNLL04 and ADE datasets. The best scores are shown in bold, and the second-best ones are underlined.

We report the average results over five runs for each dataset. Table 5.1 presents the macro-average scores for CoNLL04 and ADE datasets. We highlight special patterns related to macro-averaged results as follows:

- **Performance on CoNLL04**:
  - For NER, the model ASpERT-NER achieves the highest recall (86.93 %), while ASpERT provides the best precision (86.57 %). The highest overall F1-score is achieved by ASpERT (85.97 %), with ASpERT-NER close behind (85.46 %). This indicates a potential trade-off between precision and recall in entity extraction.
  - For RE, ASpERT-NER achieves the best recall (74.96 %), but its F1-score is (72.72 %) only marginally higher than ASpERT (71.66 %). Here, ASpERT-RE shows the lowest performance.
- **Performance on ADE**:
  - For NER, ASpERT-RE marginally outperforms both baselines and other WeLT variants, achieving the highest F1-score (91.83 %), as well as the best precision and recall.
  - For RE, ASpERT performs slightly better than ASpERT-RE in terms of F1-score (82.76 % vs. 82.45 %).
- **Key Observations**:
  - ASpERT-NER performs well in tasks requiring high recall, particularly in the CoNLL04 dataset, while ASpERT achieves better precision.
  - ASpERT-RE demonstrates modest performance in both entity and relation extraction tasks in the ADE dataset, excelling in F1-scores and precision.
  - ASpERT-NERE generally underperforms relative to the other variants, suggesting that balancing both entities and relations may not provide significant advantages in this context.

In summary, the results show that the proposed models (ASpERT-NER and ASpERT-RE) achieve slightly competitive or modest performance compared to the baseline models in both datasets.
In Table 5.2, we present the micro-average scores for both datasets. We observe distinctive patterns in the micro-averaged results, which are outlined as follows:

- **Performance on CoNLL04**:
  - For NER, ASpERT-NER achieves the best recall (90.08 %), while ASpERT has the highest precision (89.03 %) and the best F1-score (88.77 %). ASpERT-NER closely falls behind with F1-score of (88.36 %), showing that it performs well in terms of recall but slightly lower in precision compared to ASpERT.

Dataset	Model	Entity			Relation		
		Precision	Recall	F1	Precision	Recall	F1
CoNLL04	SpERT (Eberts and Ulges, 2020)	87.64	89.03	88.32	70.72	67.58	69.11
	ASpERT (Jianquan Ouyang, 2022)	**89.03**	88.53	**88.77**	**73.62**	67.39	70.36
	ASpERT-NER	86.71	**90.08**	88.36	69.13	**73.22**	**71.12**
	ASpERT-NERE	86.87	88.13	87.49	67.24	68.80	68.01
	ASpERT-RE	87.83	86.45	87.13	70.06	66.18	68.07
ADE	SpERT (Eberts and Ulges, 2020)	89.83	91.40	90.60	79.70	83.29	81.45
	ASpERT (Jianquan Ouyang, 2022)	90.68	91.56	91.12	**81.65**	83.92	**82.76**
	ASpERT-NER	90.00	91.57	90.78	77.72	83.14	80.34
	ASpERT-NERE	90.00	91.57	90.78	77.72	83.14	80.34
	ASpERT-RE	**91.34**	**91.84**	**91.59**	80.95	**84.01**	82.45

Table 5.2: Micro-average F1-scores comparison between SpERT and ASpERT with the proposed WeLT-ASpERT(⋆) models on the CoNLL04 and ADE datasets. The best scores are shown in bold, and the second-best ones are underlined.

  - For RE, ASpERT-NER demonstrates the best recall (73.22 %) and F1-score (71.12 %), marginally outperforming both ASpERT (70.36 %) and SpERT (69.11 %). This indicates an advantage of the proposed model in the relation extraction task for this dataset.

- **Performance on ADE**:
  - For NER, ASpERT-RE outperforms all models with the highest precision (91.34 %), recall (91.84 %), and F1-score (91.59 %). This shows the modest performance of the proposed model in this dataset.
  - For RE, ASpERT achieves the best F1-score (82.76 %), closely followed by ASpERT-RE (82.45 %). However, ASpERT-RE has a slightly lower recall (80.95 %) compared to ASpERT, but it slightly excels in precision (84.01 %).

- **Key Observations**:
  - ASpERT-NER performs relatively well in recall for entity extraction on CoNLL04 and has the best F1-score for relations.
  - ASpERT-RE excels in entity extraction for the ADE dataset, achieving the highest scores in all metrics. It also shows competitive performance in relation extraction.
  - ASpERT maintains strong precision for both datasets, particularly in relation extraction, where it achieves the best F1-scores for the ADE dataset.
  - ASpERT-NERE generally performs slightly below the other variations, suggesting that combining NER and RE tasks does not provide a significant improvement in micro-averaged performance.

The proposed models (ASpERT-NER and ASpERT-RE) exhibit slightly competitive or superior performance compared to the baseline models in both datasets. The results suggest that JNERE benefit from WeLT-ASpERT variant models showing particular strengths in precision and F1-scores, depending on the task and dataset.

In summary, the WeLT-ASpERT variants demonstrate marginal improvements and comparable performance to the baseline models across both datasets. Specifically, enhancements are more pronounced in the ADE dataset, where the ASpERT-RE model shows slight advantages in both entity recognition and relation extraction tasks. These results suggest that incorporating WeLT loss functions is less effective in ASpERT models compared to SpERT models. We believe this is attributable to the positive impact of the novel span classifier in ASpERT.

5.3.2 Error Analysis

We highlight descriptions of incorrect predictions from the ASpERT-NER to delineate future directions for improvements in Table 5.3. We were interested to use the exact sampled 100 error instances to compare the errors generated by ASpERT-NER with SpERT-NERE as previously highlighted in Table 4.3. We categorised generated errors into multiple predefined error categories as follows:

- **Entity recognition errors** occur when our model fails to correctly identify named entities, leading to false positives (i.e., misclassification of a non-entity as an entity) or false negatives (i.e., failure to identify a valid entity).
- **Relation extraction errors** occur when our model incorrectly predicts a relationship between identified entities, including false positives (i.e., misclassification of a non-relation as a relation) or false negatives (i.e., missing a valid relationship).
- **Joint training errors** stem from the interaction between both tasks (entity recognition and relation extraction), where errors may propagate. Typically, a misclassified entity or span error can lead to incorrect relation predictions.

The following examples illustrate the error cases and highlight the common errors and differences between ASpERT-NER and SpERT-NERE:

- NER misclassification: In this example, the model failed to classify the entity “Organization of the Oppressed on Earth” as an organization. Instead, “Earth” was falsely recognised as a location entity, leading to the omission of a “Live_In” relation. In contrast, SpERT-NERE correctly classified the relation but misclassified the entity “Organization of the Oppressed on Earth”.

(a) NER misclassification + missing relations	
Sentence	Text of the statement issued by the [Organization of the Oppressed on Earth]$_{ORG}$ claiming [U.S.]$_{LOC}$ Marine Lt.[William R. Higgins]$_{PEOP}$ was hanged.
Ground-Truth	[Organization of the Oppressed on Earth]$_{ORG}$ ([William R. Higgins]$_{PEOP}$,Live_In,[U. S.]$_{LOC}$)
Prediction	[Earth]$_{LOC}$ [U. S.]$_{LOC}$ [William R. Higgins]$_{PEOP}$
(b) Missing entities and relations	
Sentence	The "poison pill," ruled illegal in November by [U. S.]$_{LOC}$ District [G. Ernest Tidwell]$_{PEOP}$, would become effective after a shareholder had acquired 10 percent of the outstanding stock.
Ground-Truth	([G. Ernest Tidwell]$_{PEOP}$,Live_In,[U. S.]$_{LOC}$)
Prediction	([ ])
(c) Incorrect relations	
Sentence	[Port Arthur]$_{LOC}$ Mayor [Malcolm Grant]$_{PEOP}$ asked the 800 residents of [Sabine Pass]$_{LOC}$ to evacuate the coastal community just west of the [Louisiana] $_{LOC}$ line, citing the likelihood of high water closing the only highway between the town and [Port Arthur]$_{LOC}$.
Ground-Truth	([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$) ([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$)
Prediction	([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$) ([Malcolm Grant]$_{PEOP}$,Live_In,[Sabine Pass]$_{LOC}$) ([Sabine Pass]$_{LOC}$,Located_In,[Port Arthur]$_{LOC}$)
(d) Missing relations	
Sentence	[Judith C. Toth]$_{PEOP}$ says she returned for a fourth term in [Maryland]$_{LOC}$'s [House of Delegates]$_{ORG}$ because she couldn't find a better job.
Ground-Truth	[Judith C. Toth]$_{PEOP}$ ([House of Delegates]$_{ORG}$,OrgBased_In,[Maryland]$_{LOC}$)
Prediction	[Judith C. Toth]$_{PEOP}$ ([ ])
(e) Logical error	
Sentence	[Soviet]$_{LOC}$ Foreign [Eduard A. Shevardnadze]$_{PEOP}$ is to visit [China]$_{LOC}$ next month to pave the way for the first Chinese - Soviet summit in 30 years, Chinese television reported Monday.
Ground-Truth	([Eduard A. Shevardnadze]$_{PEOP}$,Live_In,[Soviet]$_{LOC}$)
Prediction	([Eduard A. Shevardnadze]$_{PEOP}$,Live_In,[Soviet]$_{LOC}$) ([Eduard A. Shevardnadze]$_{PEOP}$,Live_In,[China]$_{LOC}$)
(f) Propagated error	
Sentence	An art exhibit at the [Hakawati Theatre]$_{ORG}$ in Arab east [Jerusalem]$_{LOC}$ was a series of portraits of Palestinians killed in the rebellion.
Ground-Truth	([Hakawati Theatre]$_{ORG}$,OrgBased_In,[Jerusalem]$_{LOC}$)
Prediction	([Hakawati Theatre]$_{LOC}$,Located_In,[Jerusalem]$_{LOC}$)
Correct predictions	
Sentence	"He is the same easy-going, soft-spoken, self-effacing man we knew as governor of [New Hampshire]$_{LOC}$", said [Virginia]$_{LOC}$ Gov. [Gerald Baliles]$_{PEOP}$, a Democrat.
Ground-Truth	([Gerald Baliles]$_{PEOP}$,Live_In,[Virginia]$_{LOC}$) [New Hampshire]$_{LOC}$
Prediction	([Gerald Baliles]$_{PEOP}$,Live_In,[Virginia]$_{LOC}$) [New Hampshire]$_{LOC}$

Table 5.3: Common error cases of the ASpERT-NER on the CoNLL04's test set. The red colour expresses error cases and blue colour illustrates the ASpERT-NER improvements over the SpERT-NERE.

- Missing entities and relations: The model entirely failed to recognise the entities and relations in error (b). However, SpERT-NERE also encountered issues with incorrect named entity spans.
- Incorrect relation extraction: The model incorrectly predicted a "Live_In" relation between "Malcolm Grant" and "Sabine Pass". Additionally, it misidentified a "Located_In" relation between "Sabine Pass" and "Port Arthur", while missing a "Live_In" relation between "Malcolm Grant" and "Port Arthur" at position 38 in the sentence.
- Missing relations: The model missed the "OrgBased_In" relation between "House of Delegates" and "Maryland". However, it correctly identified "Judith C. Toth", unlike SpERT-NERE.
- Logical error: This common error occurred in both models, which incorrectly assigned a "Live_In" relation between "Eduard A. Shevardnadze" and "China", when the correct relation is with "Soviet".
- Propagated error: This common error involves both models misclassifying "Hakawati Theatre" as a location rather than an organization, leading to incorrect relation predictions.

It is noteworthy that no incorrect named entity spans were observed. However, there were additional missing entities and relations. For instance, error (e) in Table 4.3 was correctly predicted by ASpERT-NER, as shown in the last rows of Table 5.3.

5.4 Summary and Discussion

The WeLT-ASpERT models are the cost-sensitive versions of the ASpERT model. We compare the performance of WeLT-ASpERT and ASpERT, focusing on precision, recall, and F1 scores across the CoNLL04 and ADE datasets.
Key patterns observed in the experimental macro-averaged results include:

CoNLL04 Dataset

- For NER, ASpERT achieves the highest precision (86.57 %), outperforming all other models. ASpERT-NER excels in recall (86.93 %), slightly surpassing ASpERT (85.49 %). ASpERT maintains the highest F1 score (85.97 %), closely followed by ASpERT-NER (85.46 %).
- For RE, ASpERT-NER leads in recall (74.96 %), significantly outperforming ASpERT (69.01 %). The best F1 score is achieved by ASpERT-NER (72.72 %), marginally edging out ASpERT (71.66 %).

ADE Dataset:

- For NER, ASpERT-RE achieves the highest precision (91.55 %), slightly outperforming ASpERT (90.96 %). Both also lead in recall (92.12 %), slightly ahead of ASpERT (91.87 %), and achieve the best F1 score (91.83 %), marginally surpassing ASpERT (91.41 %) and SpERT (90.91 %).
- For RE, ASpERT achieves the highest precision (81.65 %), closely followed by ASpERT-RE (80.95 %). ASpERT-RE leads in recall (84.01 %), marginally surpassing ASpERT (83.92 %). ASpERT maintains the highest F1 score (82.76 %), with ASpERT-RE close behind (82.45 %).

Overall, the WeLT-ASpERT models show modest competitive performance with the same training costs as the original ASpERT model, particularly on the ADE dataset, where they consistently marginally outperform ASpERT and SpERT. In RE, ASpERT-RE achieves small gains in precision, but these come with a slight reduction in recall, impacting the overall F1 score. On the ADE dataset, WeLT-ASpERT models perform similarly to ASpERT, with minimal differences in F1 scores.
In summary, the WeLT-ASpERT models, particularly ASpERT-NER and ASpERT-RE, exhibit modest competitive and sometimes superior performance compared to the baselines. Slight improvements by WeLT-ASpERT variants are most noticeable in the ADE dataset, while ASpERT remains the best choice for NER in CoNLL04.

Study Limitations

Throughout Chapters 4 and 5, we were constrained to examining WeLT's span-based approach on English JNERE datasets. Nevertheless, we assert that the proposed approach can be seamlessly adapted to other languages. The study was limited to using $\text{BERT}_{\text{BASE}}$ and BioBERT as encoders. To balance the relation classifier, the evaluation was restricted to datasets containing relations with specific pairs of entity types. Future research will extend the assessment to more complex datasets, such as SciERC, which include relations among multiple possible pairs of entity types.

6 Table-filling JNERE Using WeLT

Chapters 3 to 5 show WeLT's advantages on addressing the class imbalance problem in the grossly skewed general-domain and biomedical real-world applications. The results show the modest improvements of WeLT, as weighted loss trainer with cost-sensitive fine-tuning on various downstream tasks: single-label BioNER, impact of recognised named entities by the WeLT-based model on BioNEL, and nested and overlapping NER. We only focused on span-based JNERE models including SpERT and ASpERT. Hence, in this chapter, we investigate the performance of WeLT on a table-filling JNERE approach utilizing the BILOU tagging scheme that inherits the class imbalance problem.

As mentioned in Section 2.1.3, some of the JNERE approaches are cast as a table-filling approach. Typically, a two-dimensional table is constructed where each entry captures the relation between two individual words within a sentence.

As shown in Figure 6.1, NER is regarded as a sequence labelling problem. Thus, a label is assigned to a word based on its relative position to the corresponding named entity and type.

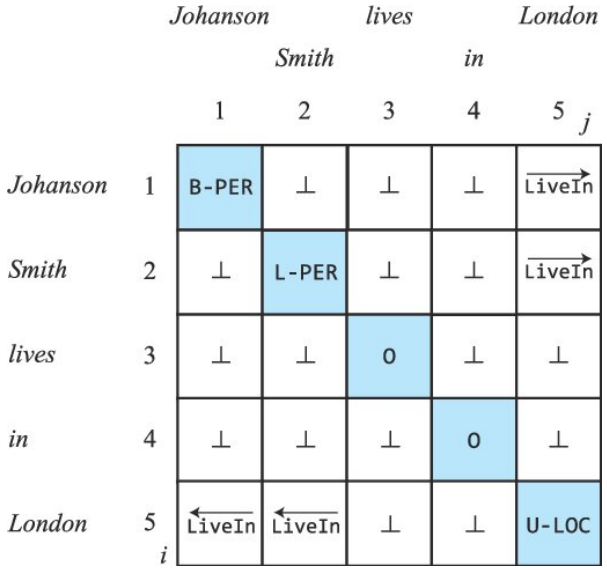

Figure 6.1: A basic example of table-filling strategy as proposed by Miwa and Sasaki (2014). This image is taken from (Ma et al., 2020). ⊥ denotes a non-relation label.

Miwa and Sasaki (2014) proposed one of the earliest table-filling approaches that modelled JNERE using a table representation. A key trade-off of this approach is that it does not tackle the issue

of nested or overlapping entities. Furthermore, these methods do not adopt the strong negative sampling techniques discussed in Chapters 4 and 5.
Recently, Ma et al. (2022) presented a refined table-filling approach referred to as TablERT-CNN. Such table-filling approaches annotate entity labels using the BILOU tagging scheme, along with directed relation labels. As a result, these tagging schemes inherit the class imbalance problem, with "O" tags constituting one of the majority classes. Thus, fine-tuning TablERT-CNN naively on training data without considering class distributions results in a biased model.
To address the class imbalance issue, a modified joint training loss using WeLT is proposed to mitigate the class imbalance problem in table-filling approaches. One of the latest promising methods is TablERT-CNN (Ma et al., 2022), which introduces a novel approach to JNERE by stacking CNNs on BERT. Therefore, WeLT-TablERT-CNN is proposed to specifically tackle the class imbalance problem in training datasets. Experimental results on the CoNLL04, ADE, and SciERC datasets demonstrate that the proposed model marginally outperforms the table-filling baselines.

In summary, our contributions in this chapter are as follows:

- We present cost-sensitive named entity and relation classification approaches to balance the data distribution gap between majority and minority classes in TablERT-CNN. Thus, we propose a novel joint loss function using WeLT.
- We investigate several WeLT loss functions on the overall performance:
  - Applying only cost-sensitive NER classification, named "TablERT-CNN-NER".
  - Applying only cost-sensitive RE classification using `weight` parameter in the cross-entropy loss function, referred to as "TablERT-CNN-RE".
  - Combining both cost-sensitive NER and RE classification termed "TablERT-CNN-NERE".
- Extensive experiments where conducted to evaluate WeLT-TablERT-CNN variants on CoNLL04, ADE and SCiERC datasets. The results show the marginal outperformance of the WeLT-TablERT-CNN variants over TablERT-CNN for JNERE tasks.
- We release the code[1] and share hyperparameters to reproduce our research results.

Structure. Section 6.1 provides an overview of related work on JNERE models, with a primary focus on table-filling approaches. Subsequently, in Section 6.2, we discuss the TablERT-CNN model in detail and delineate the main trade-offs of this approach, identifying the research

[1]WeLT-TablERT-CNN code: https://github.com/mobashgr/WeLT-TablERT-CNN, last accessed: 01.08.2024.

gaps. Building upon this background, we then justify the need for cost-sensitive TablERT-CNN using WeLT in Section 6.3. In Section 6.4, we highlight the experimental settings and present our results. Finally, a chapter's summary and discussion are presented in Section 6.5.

6.1 Related Work

Earlier studies have designed sophisticated features to encode contexts and long-range dependencies between named entities and relations. For instance, Miwa and Sasaki (2014) applied hand-crafted syntactic features such as the shortest path between two words in a syntactic tree. Miwa and Sasaki proposed a table-filling approach on which the entry at row i and column j of the table corresponds to the pair of i-th and j-th word of the input sentence, as illustrated in Figure 6.1.

Zhang et al. (2017) extracted syntactic information using pre-trained syntactic parser encoder. Gupta et al. (2016) proposed an enhanced table-filling approach by adapting recurrent neural networks to fill table's cells in a pre-defined sequential order. Miwa and Bansal (2016) proposed a bidirectional tree-structured and sequential LSTM-RNNs to represent entities and relations. Several studies explored the deep contextualized word representations to address the sequential labelling problem. Straková et al. (2019) proposed two neural network architectures for nested named entities. Their work shows that contextualized representations improve the accuracy of information extraction.

Liu et al. (2019a) designed an enhanced deep transition architecture utilizing the global context. Their results demonstrate improvements on entity extraction and chunking tasks due to contextualized word embeddings. Tran and Kavuluru (2019) proposed novel CNNs utilizing the table-filling approach. Recently, efforts have been made to incorporate BERT into table-filling framework. Wang and Lu (2020) designed two separate encoders for named entities and relations. They leveraged the attention weights from BERT's relation encoder to capture word-word interactions. Ren et al. (2021) proposed a global feature-oriented triple extraction model using a transformer-based approach to capture global information through iterative processes.

Later in Chapter 7, we review various LLMs for NER and RE and compare their performance against the proposed WeLT-based models, as discussed in Chapters 4 to 6.

Table-filling by Contextualized Representations

TablERT is an enhanced table-filling approach that utilizes BERT's contextualized representations initialized with pre-trained weights (Ma et al., 2020). TablERT represents entity mentions and encodes long-range dependencies among entities to simplify feature engineering via BERT's pre-trained weights. The authors utilized a tensor dot product to fill in the relation labels cells in the table simultaneously, unlike former table-filling approaches (Miwa and Sasaki, 2014).

TablERT is formally defined as an $n \times n$ upper triangular matrix Y, as shown in Figure 6.2. The diagonal element $Y_{i,i}$ represents the named entity label for the word w_i, which is part of the predefined set of named entities $\mathcal{E}$, following the BILOU scheme (see Section 2.2.3), as illustrated in Figure 2.8. Thus, $Y_{i,i} \in \mathcal{E}$ for $1 \leq i \leq n$.

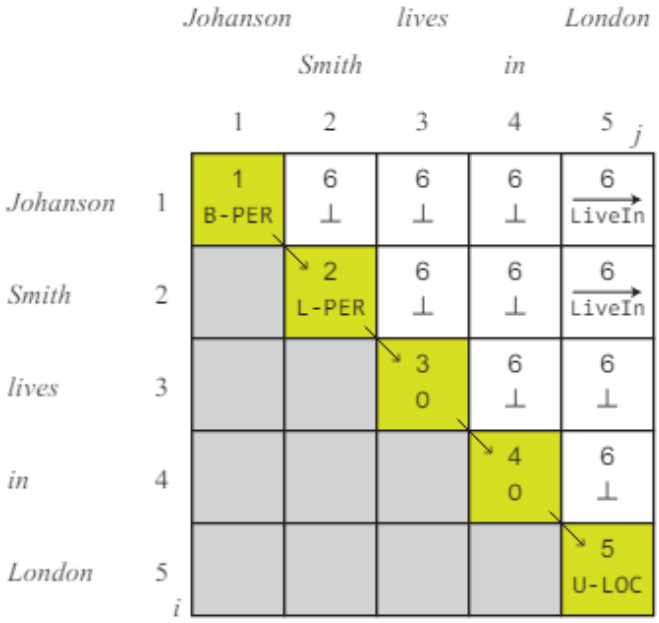

Figure 6.2: An example of TablERT table-filling strategy taken from (Ma et al., 2020). ⊥ denotes a non-relation label.

An off-diagonal element $Y_{i,j} \in \mathcal{R} (1 \leq i < j \leq n)$ specifies a directed relation label between words w_i and w_j. TablERT utilizes the upper triangular part of the table to represent the directed relation labels $\mathcal{R}$. Thus, TablERT maps a sequence of words $[w_1, w_2, ..., w_n]$ to the upper triangular matrix Y.

Example 6.1. *(TablERT Demonstration)*
Given the following sentence, "Johanson Smith lives in London" as depicted in Figure 6.2. The named entity labels are added in the diagonal tabular cells. Thus, the named entities are denoted as $Y_{i,i}$ and defined as: "Johanson" in $Y_{1,1}$ is labelled as (B-PER), "Smith" in $Y_{2,2}$ is labelled as (L-PER), and "London" in $Y_{5,5}$ is labelled as (U-LOC). The other words that do not correspond to pre-defined entity types are labelled as (O).
Each word within an entity span is annotated with a corresponding relation label added in the off-diagonal tabular cells in $Y_{i,j}$. Consequently, "Johanson Smith" is labelled as a (Person) and there exists a relation $\overrightarrow{LiveIn}$ added to both "Johanson" in $Y_{1,5}$ and "Smith" in $Y_{2,5}$. ⊥ indicates a non-relation label.

Ma et al. (2022) designed separate prediction models for NER and RE. The authors sequentially assign a label to each word via features at the current and previous time-steps. Concerning RE, they concatenate word embeddings with their corresponding entity label embeddings as relation embeddings. Hence, the relation scores of each word pair are computed based on a matrix multiplication of the linearly transformed relation embeddings. In addition, Ma et al. (2022) adopted hyper-parameter tuned weight losses to balance the NER and RE training. The authors modified the joint loss functions by adding two hyperparameters $\lambda^{(ent)}$ and $\lambda^{(rel)}$ for entity and relation losses, respectively. The authors trained TablERT with various $\lambda^{(ent)}$ and $\lambda^{(ent)}$ from 0.1 to 0.9 so that their sum is equal to 1. The F1 scores of predicted results on the CoNLL04 test data show that adjusting weights did not have a significant impact on the overall performance. However, they observed that increasing $\lambda^{(ent)}$ leads to improved NER F1 score and the same applies with high $\lambda^{(rel)}$ for the highest strict RE F1 score. In general, Ma et al. (2022) observed that downscaling $\lambda^{(rel)}$ worsened the results, and the authors suggested that this is due to complexity of the RE task over NER. Thus, they recommended that $\lambda^{(rel)}$ should be greater than $\lambda^{(ent)}$ during training.

TablERT results reveal a promising performance. However, it also introduces some limitations:

- TablERT employs two separate prediction models for NER and RE, which limits its ability to capture the full range of interactions between entities and relations across the table cells.
- TablERT predicts relation labels for pairs of entities independently, without accounting for potential dependencies or relationships between labels assigned to adjacent or nearby table cells. While it processes all relations in parallel (predicting them simultaneously), this approach overlooks possible correlations between neighbouring entity pairs, which could capture useful dependencies.

To this end, Ma et al. (2022) proposed an extension of TablERT that incorporates local dependencies along with the contextualized representations of BERT, referred as "TablERT-CNN". The following section discusses the adoption of two-dimensional convolutional neural networks (2D-CNNs) to the output of BERT.

6.2 Table Labelling Using CNNs

TablERT-CNN considers each table as a two-dimensional image (2D image), and each cell as a pixel, transforming the JNERE task into a table-labelling problem at the cell level. By applying 2D-CNNs to the output of BERT, TablERT-CNN is able to implicitly perceive local information and label dependencies from neighbouring cells. Figure 6.3 depicts an overview of TablERT-CNN under the setting in which the prediction model contains only a single CNN layer.

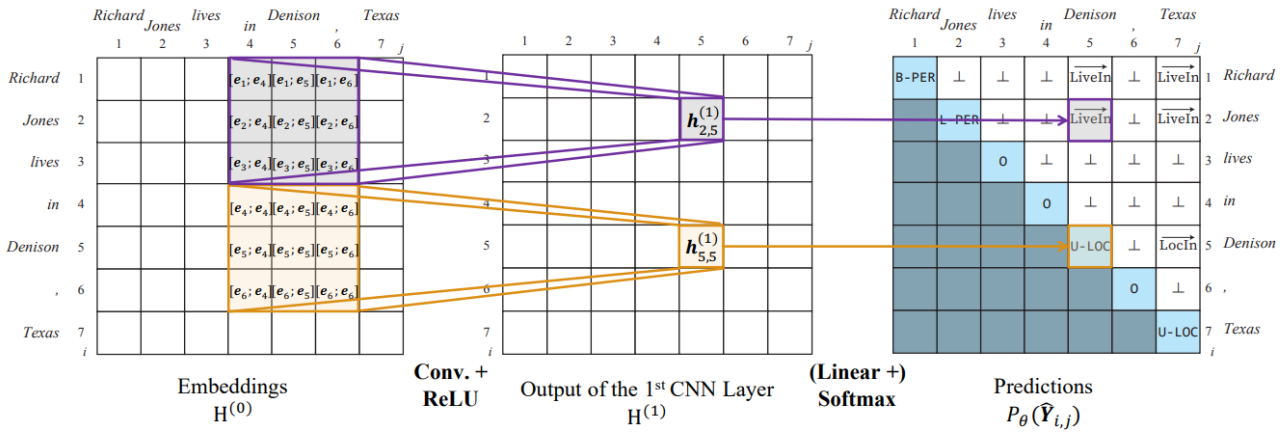

Figure 6.3: An example of the TablERT-CNN with a single CNN layer table-filling strategy taken from (Ma et al., 2022). The right side represents the table-filling representation. The entire table illustrates the features and the upper triangular part to represent the labels.

Similar to TablERT's approach, TablERT-CNN uses the upper triangular part of the table to represent both entity and relation labels. In the following sections, we present the TablERT's approach based on: table representation, word embeddings, prediction model, and joint loss function.

6.2.1 Table Representation

Ma et al. (2020) defined a matrix $Y \in \mathbb{R}^{n_w \times n_w}$ where n_w is the number of words in the sentence. Using the upper triangular part to represent the label space of NER and RE, a diagonal entry $Y_{i,i}$ indicates the entity label of word w_i and off-diagonal entry $Y_{i,j}$ $(j > i)$ represents the relation label of word pair (w_i, w_j). The NER labels are annotated using the BILOU tagging scheme (see Section 2.2.3), and each relation with a directed hard-encoded relation label.

6.2.2 Word Embeddings

TablERT-CNN obtains word embeddings from BERT's contextualised representations. The embedding e_i for a word w_i, which is split into subwords $[\text{start}(i), \ldots, \text{end}(i)]$ and computed via Equation 6.1. The word embedding process is typically employed when subword tokenisation is applied (e.g., WordPiece tokenisation in the case of BERT).

$$e_i := \max(x^l_{start(i)}, \ldots, x^l_{end(i)}) \tag{6.1}$$

where:

- $x^l \in \mathbb{R}^{d_{emb}}$ is the output of the BERT model,

- l is the layer index,
- d_{emb} is the dimension size, and
- max is the max-pooling function.

6.2.3 Prediction Model

TablERT-CNN adopts a 2D-CNN, capturing the local dependencies among neighbouring cells. The 2D table is treated as an image and each table cell is considered to be a pixel. The 2D-CNN encodes the representation of each cell, as depicted in Figure 6.3. For each word pair (w_i, w_j), word embeddings e_i, e_j are concatenated, and the bottom layer $H^{(0)} \in \mathbb{R}^{n_w \times n_w \times 2d_{\text{emb}}}$ is constructed as defined in Equation 6.2.

$$H^{(0)}_{i,j} := [e_i \circ e_j] \tag{6.2}$$

where:

- $\circ$ denotes the concatenation of two vectors, and
- the dimension of the vector representation for each cell in layer l as d_l.

The output of the first 2D-CNN layer $H^{(1)}$ is computed based on the output of the bottom layer $H^{(0)}$.
For the NER task, TablERT-CNN linearly transforms the representations of the diagonal cells at the last layer L to compute the entity label distribution of each word (w_i):

$$P_\theta(\hat{Y}_{i,i}) := \text{softmax}(W \cdot H^{(L)}_{i,i} + b) \tag{6.3}$$

where P is the estimated probability function, θ denotes the model parameters, $W \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{R}|}$ denotes the trainable weight matrix and $b \in \mathbb{R}^{|\mathcal{E}|}$ represents the bias vector.
Concerning the RE task, the last layer of the 2D-CNN is employed as a convolutional classifier. Hence, for each word pair (w_i, w_j) where $i \neq j$, the relation label $P_\theta(\hat{Y}_{i,j})$ is computed using Equation 6.4:

$$P_\theta(\hat{Y}_{i,j}) := \text{softmax}(H^{(L)}_{i,j}) \tag{6.4}$$

where L is the last layer, thus the output dimension is identical to the relation labels $d_L := |\mathcal{R}|$.

6.2.4 Training and Prediction

The training objective function is defined as the sum of cross-entropy losses for NER as defined in Equation 6.5 and RE in Equation 6.6. The main objective is to minimize the total loss to update the model parameters θ.
Given the ground-truth label matrix $Y \in \mathbb{R}^{n_w \times n_w}$, the losses are computed as:

- For the entity label prediction of each word w_i, TablERT-CNN selects the highest probability from $P_\theta(\hat{Y}_{i,i})$. Otherwise, the entity type label for the last word is selected as a final prediction in case of a conflict.

$$\mathcal{L}^{\text{NER}} := -\sum_{1 \le i \le n_w} \log P_\theta(\hat{Y}_{i,i} = Y_{i,i}) \tag{6.5}$$

- Regarding the prediction of the relation label for each entity pair (s_i, s_j), the last words of both entity spans s_i and s_j are selected. For instance, if the last words of the entity spans s_i and s_j are indexed as end(i) and end(j), respectively, the predicted relation label for the entity pair (s_i, s_j) is determined by the label with the highest probability from $P_\theta(\hat{Y}_{\text{end}(i),\text{end}(j)})$.

$$\mathcal{L}^{\text{RE}} := -\sum_{\substack{1 \le i \le n_w \\ i < j \le n_w}} \log P_\theta(\hat{Y}_{i,j} = Y_{i,j}) \tag{6.6}$$

Thus, the total loss for the JNERE task in TablERT-CNN is defined as:

$$\mathcal{L}^{\text{TablERT-CNN}} := \mathcal{L}^{\text{NER}} + \mathcal{L}^{\text{RE}} \tag{6.7}$$

6.3 Cost-sensitive TablERT-CNN Using WeLT

TablERT-CNN is a JNERE approach that stacks CNNs on BERT. The table representations model the entities and relations, casting the entity and relation extraction as a table-labelling problem. TablERT-CNN does not utilize the strong negative sampling as in previous models such as SpERT (Eberts and Ulges, 2020) and ASpERT (Jianquan Ouyang, 2022).
However, as shown in Figure 6.3, the table is filled with BILOU entity labels in the diagonal cells and relation directed labels (i.e., right and left). As previously mentioned, the BILOU tagging scheme exhibits inherent imbalance issues:

- The (O) tag appear far more frequently than others. Most words in a text are not part of named entities, leading to a high prevalence of the (O) tag.

- The (B), (I), (L), and (U) tags are much less frequent because they only apply to words that are part of named entities. Moreover, within named entities, the distribution of these tags can vary. For instance, the unit label (U), which denotes single-token entities, may appear more frequently in texts with a higher occurrence of such entities, while labels like (I) and (L) are more common in texts with longer, multi-token entities.

Figures 6.4, 6.5, and 6.6 depict the BILOU entity label values of CoNLL04's, ADE's and SciERC's, respectively. There are common patterns in these three figures: (O) tags are the majority class, and

(U) tags are one of the minority classes with an exception of "U-Loc" in CoNLL04's dataset, "U-drug" in ADE's dataset and "U-generic" in SciERC's dataset.

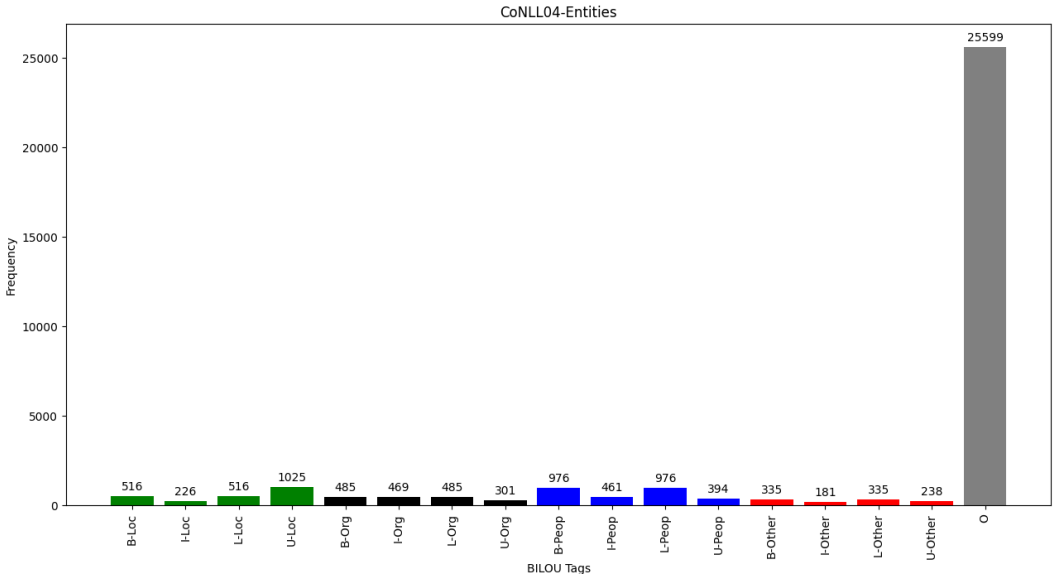

Figure 6.4: Frequency of entities in CoNLL04's training dataset. The "O" tags are not part of the pre-defined entity types.

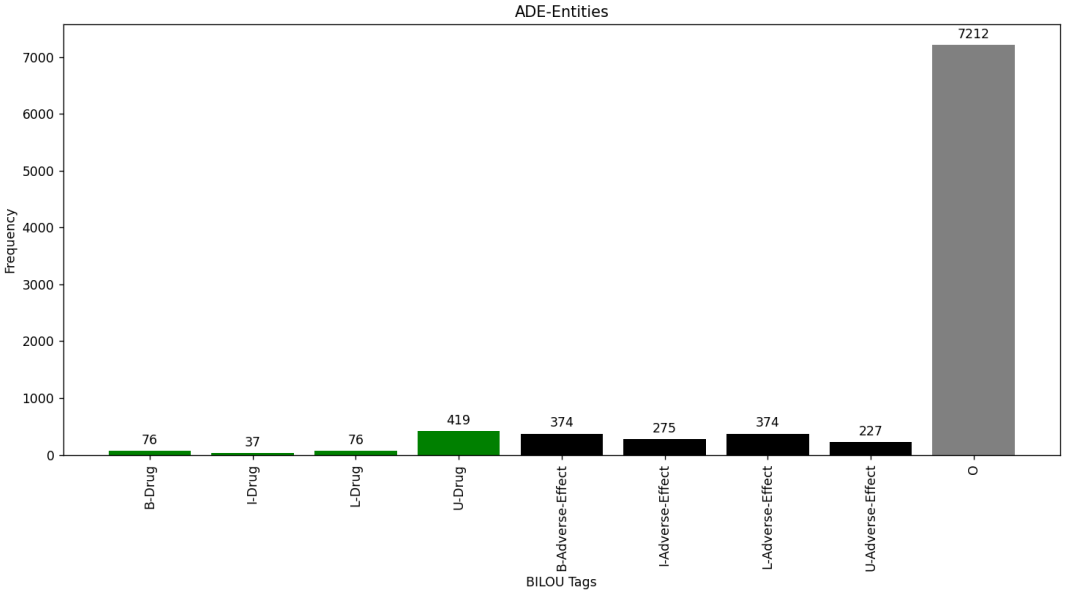

Figure 6.5: Frequency of entities in ADE's training dataset. The "O" tags are not part of the pre-defined entity types.

Since ADE has only one relationship and lacks 'none' relations, addressing relation class imbalance is not applicable in this case. Hence, Figures 6.7 and 6.8 show the values of relation distribution respectively.

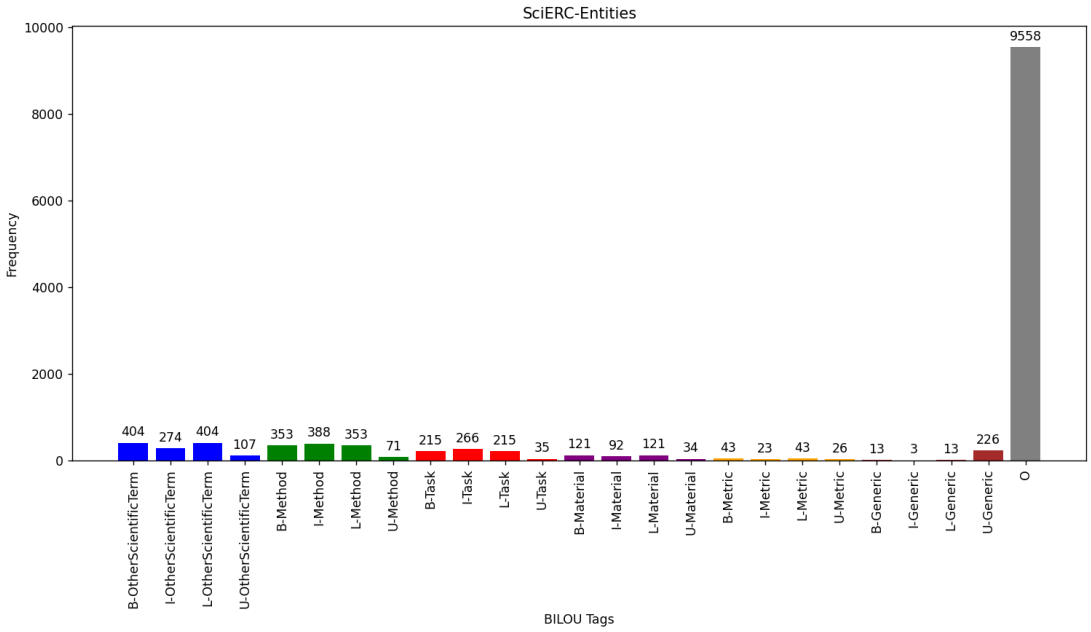

Figure 6.6: Frequency of entities in SciERC's training dataset. The "O" tags are not part of the pre-defined entity types.

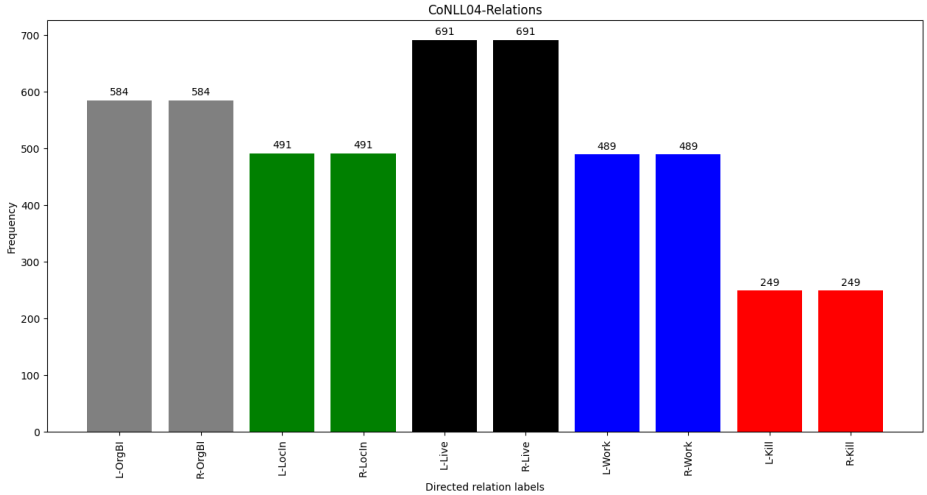

Figure 6.7: Frequency of directed relations in CoNLL04's training dataset. "L-" and "R-" mean left and right direction, respectively.

In CoNLL04's dataset, the "Kill" relationship is considered to be one of the minority classes and the "Live" relationship is one of the majority classes. For the SciERC's dataset, the "Feature-of" relationship is one of the minority classes and the "Used-for" relationship is one of the majority classes.

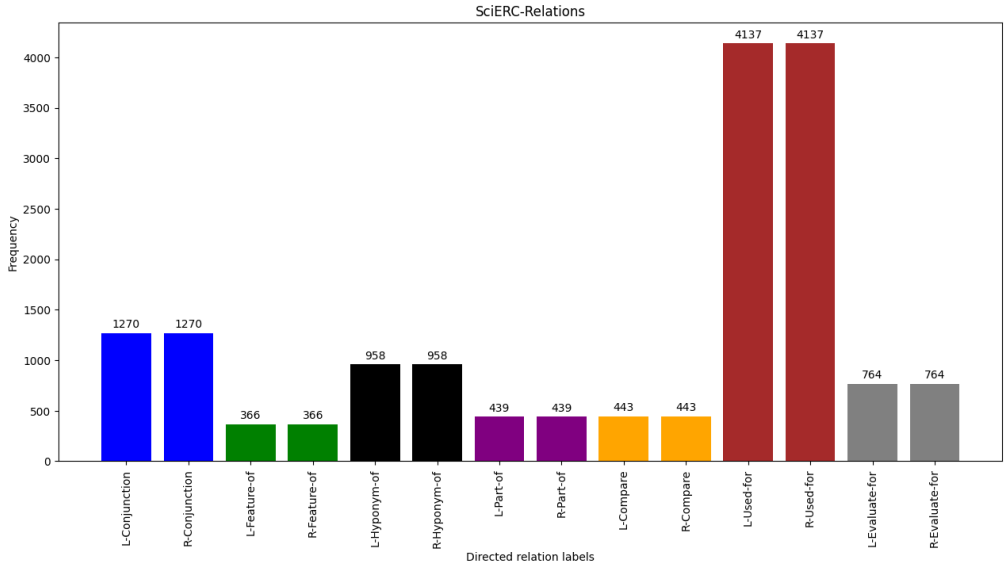

Figure 6.8: Frequency of directed relations' in SciERC's training dataset."L-" and "R-" mean left and right direction, respectively.

Consequently, the entity and relation labels in TablERT-CNN are imbalanced. Thus, we argue that training TablERT-CNN naively without addressing the class imbalance may degrade the performance. Thus, this may result in a biased model towards the more frequent tags like "O". To this end, the WeLT-TablERT-CNN is proposed as a cost-sensitive version of TablERT-CNN. We present three joint loss functions to address the class imbalance by (1) only rescaling the entity labels, (2) only rescaling the relation labels, and (3) both entity and relation labels. The following sections present three loss functions and provide the joint loss functions accordingly.

6.3.1 WeLT Span Classifier

Given a sentence with number of words n_w, a matrix Y is constructed accordingly. The diagonal entries $Y_{i,i}$ represent the BILOU entity labels and off-diagonal entries $Y_{i,j}$ such that ($j > i$) are relation labels between words w_i and w_j.

The word embeddings are obtained from the BERT model, in which each word w_i is composed of the subwords and the embedding e_i is computed by max-pooling the BERT outputs of the subwords. Furthermore, a 2D-CNN is applied to the constructed matrix to capture the local dependencies among neighbouring cells.

For NER, TablERT-CNN uses a standard cross-entropy loss function. In contrast, "TablERT-CNN-NER" employs a weighted cross-entropy loss to mitigate the class imbalance.
First, we present the rescaled entities as defined in Equation 6.8:

$$w_e := \sum_{x=1}^{c} \sigma\left(1 - \frac{n_x}{\sum_{k=1}^{c} n_k}\right) \tag{6.8}$$

where:

- w_e: denotes the rescaled entity weights,
- σ: is the Softmax function,
- c: represents the total number of entity classes, and
- n_x: represents the number of instances that belong to class x.

Equation 6.9 represents the loss of WeLT entity classifier:

$$\mathcal{L}^{s}_{\text{TablERT-CNN-NER}} := -\sum_{1 \le i \le n_w} w_e \log P_\theta(\hat{Y}_{i,i} = Y_{i,i}) \tag{6.9}$$

where:

- n_w: denotes the number of words in the input sentence,
- w_e: as defined in Equation 6.8,
- c: is the number of entity classes (including "O" class),
- n_x: is the number of instances that belong to class x, and
- $\hat{Y}_{i,i}$: is the predicted probability of entity label of each word w_i as calculated in Equation 6.3.

By incorporating class-specific weights, the WeLT NER classifier becomes more sensitive to minority classes and assigns lower weights to "O" tags, penalizing the majority classes.

6.3.2 WeLT Relation Classifier

After the entity spans are classified, they are paired to form potential relations. For each pair of spans, a relation representation is constructed by concatenating their respective embedding. Then, this concatenated representation is fed into a WeLT relation classifier, and finally relations are determined based on rescaled weight scores.

TablERT-CNN uses a standard cross-entropy loss function to train the relation classifier. In contrast, "TablERT-CNN-RE" employs a weighted cross-entropy loss to address the class imbalance.
Equation 6.10 defines the calculation of rescaled relation weights:

$$w_r := \sum_{y=1}^{c_r} \sigma\left(1 - \frac{n_y}{\sum_{k=1}^{c_r} n_k}\right) \tag{6.10}$$

where:

- w_r: denotes the rescaled relation weights,
- σ: is the softmax function,
- c_r: is the total number of relation classes, and
- n_y: is the number of instances that belong to class y.

Equation 6.11 represents the WeLT relation classifier's loss using the `weight` parameter scaling the overall loss for each individual sample, allowing for instance-specific adjustments:

$$\mathcal{L}^r_{\text{TablERT-CNN-RE}} := -\sum_{\substack{1 \le i \le n_w \\ i < j \le n_w}} w_r \log P_\theta(\hat{Y}_{i,j} = Y_{i,j}) \tag{6.11}$$

where:

- n_w: is the number of words in the input sentence,
- w_r: as defined in Equation 6.10, and
- $\hat{Y}_{i,j}$: the predicted relation label for the entity pair (s_i, s_j), determined as the label with the highest probability, as defined in Equation 6.4.

6.3.3 WeLT Joint Loss Functions

In Sections 6.3.1 and 6.3.2, we introduced cost-sensitive WeLT span and relation classifiers. The objective here is to examine the effects of balancing entity classes, relation classes, and their combination. We present three variants of the WeLT-TablERT-CNN approach, each comprising two core components: one for entity classification and one for relation extraction.
Each variant employs a distinct joint loss function, which is described as follows:

- **Variant 1**: also known as "TablERT-CNN-NER" using WeLT's NER classifier as defined in Equation 6.9 and TablERT-CNN's relation classifier as defined in Equation 6.6. Hence, the joint loss function is defined as:

$$\mathcal{L}_{\text{TablERT-CNN-NER}} := \mathcal{L}^{s}_{\text{TablERT-CNN-NER}} + \mathcal{L}^{\text{RE}} \tag{6.12}$$

- **Variant 2**: also known as "TablERT-CNN-NERE" that combines WeLT's span classifier as specified in Equation 6.9 and relation classifier as defined in Equation 6.11, thus the joint loss function is defined as:

$$\mathcal{L}_{WeLT-\text{TablERT-CNN-NERE}} := \mathcal{L}^{s}_{\text{TablERT-CNN-NER}} + \mathcal{L}^{r}_{\text{TablERT-CNN-RE}} \tag{6.13}$$

- **Variant 3**: also known as "TablERT-CNN-RE" using TablERT-CNN's span classifier as specified in Equation 6.5 and WeLT's relation classifier using `weight` parameter as defined in Equation 6.11, thus the joint loss function is defined as:

$$\mathcal{L}_{\text{TablERT-CNN-RE}} := \mathcal{L}^{\text{NER}} + \mathcal{L}^{r}_{\text{TablERT-CNN-RE}} \tag{6.14}$$

In summary, the primary difference between TablERT-CNN and WeLT-TablERT-CNN variants lies in the handling of class imbalance in TablERT-CNN that fills in entity labels using the BILOU tagging scheme and hard-coded directed relation labels. WeLT-TablERT-CNN incorporates a weighted loss training mechanism that adjusts the loss function based on the frequency of classes including "O " class, thus ensuring that minority classes receive more focus during training. This strategy is absent in the original TablERT-CNN model.

6.4 Evaluating WeLT-TablERT-CNN

This section describes experiments conducted to investigate the effectiveness of addressing the class imbalance problem at the token-tagging level. The experiments utilise a table-filling approach with three different WeLT joint loss functions, as presented in Section 6.3. The results are compared to the baselines including TablERT and TablERT-CNN. All the experiments were carried out using a single Tesla P40 GPUs with 24GB memory. The hyperparameters are reported in the Appendix (see Table 12). TablERT-CNN's experimental settings have been adopted. The experiments were conducted on three public gold-standard datasets CoNLL04, ADE, and SciERC.
Since the labels are filled in the table as a token-labelling strategy on which named entities are tagged using the BILOU scheme as discussed in Section 2.2.3, we included SciERC avoiding the complications we had in Chapters 4 and 5 for balancing the relation classifier. Due to

the characteristics of the ADE dataset, which contains only one relation, we applied only the WeLT-NER joint loss function. Since TablERT-CNN does not adopt the strong negative samples, there are no non-relations. Hence, it is infeasible to balance relations in this case.

The WeLT-TablERT-CNN variant models are evaluated on both entity recognition and relation extraction with the same TablERT-CNN's evaluation strategy for a fair comparison:

- A correct entity is only considered if the predicted span and type match the ground-truth. For example, if the ground truth contains the entity `(London, Location)` with the span `[0, 6]`, and the model predicts `(London, Organization)` or `(London, Location)` but with a span `[1, 7]`, the prediction is considered incorrect. Only the prediction `(London, Location)` with span `[0, 6]` is valid.

- In relaxed relation settings (RE), a correct relation is considered with the right predicted relation label and spans of both entities. For example, suppose the ground truth relation is `(Located_In, (London, Location), (UK, Location))`. In the relaxed setting, a correct relation is considered when the model predicts the correct relation label `Located_In` and identifies the spans of both entities `(London, Position)` and `(UK, Location)`. Minor inaccuracies in entity types do not affect the correctness of the predicted relation, provided that the relation label and entity spans are correct.

- In the strict relation (RE+) context, the same considerations apply as in the relaxed relation context, with the additional requirement of identifying the correct entity types. For instance, if the ground truth relation is `(Located_In, (London, Location), (UK, Location))`, the model must predict both the correct relation type `Located_In` and accurately identify the entities `(London, Location)` and `(UK, Location)` with their exact spans and types. Any mismatch in entity spans or types renders the predicted relation incorrect.

We report micro-averaged F1 scores for the CoNLL04 and SciERC datasets, as stated in Section 2.2.2. In addition, we present the macro-averaged F1 scores for ADE and SciERC datasets, as discussed in Section 2.2.4. For ADE, the F1-score is averaged over 10-fold cross-validation. For SciERC, we only report the F1-score for relaxed relation which is in line with previous work (Luan et al., 2019; Wadden et al., 2019; Eberts and Ulges, 2020).

6.4.1 Evaluation Results

Table 6.1 presents the micro-average F1 score results of the CoNLL04 dataset. In addition to the macro-average F1 scores of ADE and SciERC datasets. We report the average results over five runs for each dataset.

Dataset	Model	Encoder	NER	RE	RE+
CoNLL04 Δ	TablERT (Ma et al., 2020)	$BERT_{\text{BASE}}$	90.2	72.8	72.6
	TablERT (Ma et al., 2020)	$BERT_{\text{LARGE}}$	90.5	**73.8**	**73.8**
	TablERT-CNN (Ma et al., 2022)	$BERT_{\text{BASE}}$	90.5	73.2	73.2
	TablERT-CNN-NER	$BERT_{\text{BASE}}$	90.4	73.3	73.1
	TablERT-CNN-NERE	$BERT_{\text{BASE}}$	90.8	73.6	73.4
	TablERT-CNN-RE	$BERT_{\text{BASE}}$	**90.9**	72.12	71.9
ADE ▲	TablERT (Ma et al., 2020)	$BERT_{\text{BASE}}$	89.9	80.6	80.6
	TablERT-CNN (Ma et al., 2022)	$BERT_{\text{BASE}}$	89.7	80.5	80.5
	TablERT-CNN-NER	$BERT_{\text{BASE}}$	**91.7**	**85.7**	**85.7**
SciERC (BERT) ▲	TablERT-CNN (Ma et al., 2022)	$BERT_{\text{BASE}}$	**67.2**	41.3	-
	TablERT-CNN-NER	$BERT_{\text{BASE}}$	63.2	38.9	-
	TablERT-CNN-NERE	$BERT_{\text{BASE}}$	66.2	**42.6**	-
	TablERT-CNN-RE	$BERT_{\text{BASE}}$	67.0	41.4	-
SciERC (SciBERT) ▲	TablERT-CNN (Ma et al., 2022)	SciBERT	**68.6**	44.7	-
	TablERT-CNN-NER	SciBERT	65.4	43.5	-
	TablERT-CNN-NERE	SciBERT	68.5	45.2	-
	TablERT-CNN-RE	SciBERT	**68.6**	**46.6**	-
SciERC (BERT) Δ	TablERT-CNN (Ma et al., 2022)	$BERT_{\text{BASE}}$	66.8	45.7	-
	TablERT-CNN-NER	$BERT_{\text{BASE}}$	63.2	44.1	-
	TablERT-CNN-NERE	$BERT_{\text{BASE}}$	65.9	**45.8**	-
	TablERT-CNN-RE	$BERT_{\text{BASE}}$	**67.2**	44.9	-
SciERC (SciBERT) Δ	TablERT-CNN (Ma et al., 2022)	SciBERT	**68.6**	48.7	-
	TablERT-CNN-NER	SciBERT	65.2	47.3	-
	TablERT-CNN-NERE	SciBERT	67.8	48.2	-
	TablERT-CNN-RE	SciBERT	**68.6**	**48.9**	-

Table 6.1: Comparison between existing methods and the proposed WeLT-TablERT-CNN(★) model on the CONLL04, ADE and SciERC datasets. The symbols Δ and ▲ represent evaluation using micro and macro average F1 values, respectively. The best scores are shown in bold, and the second-best ones are underlined.

We report the results based on the dataset level, entity recognition and two relation extraction evaluation (i.e., RE and RE+):

- For the CoNLL04 dataset, the TablERT model with $\text{BERT}_{\text{LARGE}}$ achieved the best scores in RE and RE+, with F1 scores of 73.8 % in both evaluations. The second-best performance was from the TablERT-CNN-NERE variant, which achieved 73.6 % in RE and 73.4 % in RE+. TablERT-CNN-RE achieved the highest NER score of 90.9 % , marginally surpassing all other models.

- For the ADE dataset, the proposed TablERT-CNN-NER variant achieved the highest scores across all metrics, with significant improvements in NER (91.7 %), RE (85.7 %), and RE+ (85.7 %). The second-best model was the original TablERT with $\text{BERT}_{\text{BASE}}$, obtaining 89.9 % in NER and 80.6 % in both RE and RE+.

- For the SciERC dataset evaluated with $\text{BERT}_{\text{BASE}}$ macro evaluation, the baseline TablERT-CNN achieved the best NER score of 67.2 % closely followed by the second-best performance was from TablERT-CNN-RE-weight with 67.0 % as an F1 score. TablERT-CNN-NERE slightly outperformed others in RE with a score of 42.6 % and TablERT-CNN-RE-weight achieved the second-best score with 41.4 %. Regarding the results of SciERC that are fine-tuned using SciBERT, in general they are improvements with respect to the models fine-tuned using $\text{BERT}_{\text{BASE}}$. TablERT-CNN-RE model has the highest RE score of 46.6 % and TablERT-CNN-NERE has the second-best score (45.2 %). Regarding the NER score, TablERT-CNN and TablERT-CNN-RE have the highest score of 68.8 % and TablERT-CNN-NERE has the second-best score (68.5 %).

- For the SciERC dataset evaluated with $\text{BERT}_{\text{BASE}}$ micro evaluation, the baseline TablERT-CNN model achieves a competitive F1 score of 66.8 %. The TablERT-CNN-RE model exhibits a marginal improvement, reaching an F1 score of 67.2 %. This slight enhancement suggests that the incorporation of re-weighting strategies for relations can indirectly benefit NER performance. Regarding the RE task, the TablERT-CNN-NERE model marginally leads with an F1 score of 45.8 %, slightly outperforming the baseline's 45.7 %. When utilizing the SciBERT encoder, the TablERT-CNN and TablERT-CNN-RE achieve a notable F1 score of 68.6 % for NER tasks. The TablERT-CNN-RE model closely outperforms all others, achieving an F1 score of 48.9 %. The baseline model achieved the second-best score with an F1 score of 48.7 % also performs well.

It is worth noting that the TablERT model with $\text{BERT}_{\text{LARGE}}$, which has larger number of trainable parameters (345M), achieved the best scores in RE and RE+ for the CoNLL04 dataset. However, the second-best scores, particularly in NER for the CoNLL04 dataset and

across evaluations for the ADE and SciERC datasets, were achieved by models with the proposed WeLT-TablERT-CNN variants with lower trainable parameters (110M), demonstrating efficient performance improvements without the necessity for larger models.
In summary, the analysis across datasets and models shows that the proposed WeLT-TablERT-CNN variants generally offer modest improvements over the baseline TablERT-CNN, particularly in the ADE dataset and specific configurations within the SciERC dataset. Additionally, the use of domain-specific encoders, such as SciBERT, enhances performance, especially for scientific text processing. Moreover, the WeLT re-weighting mechanisms for balancing entities and relations consistently improve performance, with some exceptions, such as relation predictions in CoNLL04 and the macro-average evaluation of entity recognition in SciERC using BERT as the encoder, where the baseline remains the best choice.

6.4.2 Error Analysis

We outline descriptions of incorrect predictions from the TablERT-CNN-NERE to delineate future directions for improvements. We unified the same error cases as in Tables 4.3 and 5.3 to compare WeLT-TablERT-CNN with the SpERT-NERE and the ASpERT-NER, respectively. Unlike the previous two error analyses in Sections 4.4.2 and 5.3.2, we only observed two missing entities and six correct predictions on the CoNLL04's test data. Regarding the missing entities, the proposed model failed to classify "Organization of the Oppressed on Earth" as an organization entity in the first sentence. Similarly, in the second sentence, it misclassified "China" by failing to recognise it as a location entity.

We highlight the correct predictions in a sequential order as presented in Table 6.2 by the TablERT-CNN-NERE model in contrast to the SpERT-NERE and the ASpERT-NER models:

- Unlike the SpERT-NERE model that lacks syntactic information for a wrong relation prediction as ([Gerald Baliles]$_{PEOP}$,Live_In,[New Hampshire]$_{LOC}$), the proposed model correctly predicts `Live_In` relationship.

- In contrast to the SpERT-NERE that missed classifying "Judith C. Toth" and the ASpERT-NER that missed predicting the relation ([House of Delegates]$_{ORG}$,OrgBased_In,[Maryland]$_{LOC}$), the proposed model correctly predicts entities and relations.

- The SpERT-NERE model had an incorrect entity span for "G. Ernest Tidwell" and missing entities and relations by the ASpERT-NER. In contrast, the proposed model matches the ground-truth.

	Missing Entities
Sentence	Text of the statement issued by the [Organization of the Oppressed on Earth]$_{ORG}$ claiming [U. S.]$_{LOC}$ Marine Lt. [William R. Higgins]$_{PEOP}$ was hanged.
Ground-Truth	[Organization of the Oppressed on Earth]$_{ORG}$ ([William R. Higgins]$_{PEOP}$,Live_In,[U. S.]$_{LOC}$)
Prediction	([William R. Higgins]$_{PEOP}$,Live_In,[U. S.]$_{LOC}$) [Organization of the Oppressed on Earth]
Sentence	[Soviet]$_{LOC}$ Foreign [Eduard A. Shevardnadze]$_{PEOP}$ is to visit [China]$_{LOC}$ next month to pave the way for the first Chinese - Soviet summit in 30 years, Chinese television reported Monday.
Ground-Truth	([Eduard A. Shevardnadze]$_{PEOP}$,Live_In,[Soviet]$_{LOC}$)
Prediction	([Eduard A. Shevardnadze]$_{PEOP}$,Live_In,[Soviet]$_{LOC}$ [China])
	Correct predictions
Sentence	"He is the same easy-going, soft-spoken, self-effacing man we knew as governor of [New Hampshire]$_{LOC}$", said [Virginia]$_{LOC}$ Gov. [Gerald Baliles]$_{PEOP}$, a Democrat.
Ground-Truth	([Gerald Baliles]$_{PEOP}$,Live_In,[Virginia]$_{LOC}$) [New Hampshire]$_{LOC}$
Prediction	([Gerald Baliles]$_{PEOP}$,Live_In,[Virginia]$_{LOC}$)
Sentence	[Judith C. Toth]$_{PEOP}$ says she returned for a fourth term in [Maryland]$_{LOC}$'s [House of Delegates]$_{ORG}$ because she couldn't find a better job.
Ground-Truth	[Judith C. Toth]$_{PEOP}$ ([House of Delegates]$_{ORG}$,OrgBased_In,[Maryland]$_{LOC}$)
Prediction	[Judith C. Toth]$_{PEOP}$ ([House of Delegates]$_{ORG}$,OrgBased_In,[Maryland]$_{LOC}$)
Sentence	The "poison pill," ruled illegal in November by [U. S.]$_{LOC}$ District [G. Ernest Tidwell]$_{PEOP}$, would become effective after a shareholder had acquired 10 percent of the outstanding stock.
Ground-Truth	([G. Ernest Tidwell]$_{PEOP}$,Live_In,[U. S.]$_{LOC}$)
Prediction	([G. Ernest Tidwell]$_{PEOP}$,Live_In,[U. S.]$_{LOC}$)
Sentence	[Port Arthur]$_{LOC}$ Mayor [Malcolm Grant]$_{PEOP}$ asked the 800 residents of [Sabine Pass]$_{LOC}$ to evacuate the coastal community just west of the [Louisiana] $_{LOC}$ line, citing the likelihood of high water closing the only highway between the town and [Port Arthur]$_{LOC}$.
Ground-Truth	([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$) ([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$)
Prediction	([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$) ([Malcolm Grant]$_{PEOP}$,Live_In,[Port Arthur]$_{LOC}$)
Sentence	An art exhibit at the [Hakawati Theatre]$_{ORG}$ in Arab east [Jerusalem]$_{LOC}$ was a series of portraits of Palestinians killed in the rebellion.
Ground-Truth	([Hakawati Theatre]$_{ORG}$,OrgBased_In,[Jerusalem]$_{LOC}$)
Prediction	([Hakawati Theatre]$_{ORG}$,OrgBased_In,[Jerusalem]$_{LOC}$)

Table 6.2: Common error cases of the TablERT-CNN-NERE on the CoNLL04's test set. The red colour expresses error cases and blue colour illustrates WeLT-TablERT-CNN improvements over the SpERT-NERE and the ASpERT-NER models.

- Incorrect relations, such as ([Sabine Pass]$_{LOC}$, Located_In, [Port Arthur]$_{LOC}$) and ([Malcolm Grant]$_{PEOP}$, Live_In, [Sabine Pass]$_{LOC}$), are predicted by both SpERT-NERE and ASpERT-NER models. However, the TablERT-CNN-NERE model correctly predicts these relations.

- Propagated errors, such as misclassifying "Hakawati Theatre" as a location instead of an organisation, lead to incorrect relation predictions in both models. In contrast, the TablERT-CNN-NERE model correctly identifies the entities and their relations.

It is noteworthy that no incorrect NE spans, NER misclassifications, incorrect relations, syntactic errors, logical errors, or propagated errors were observed in the TablERT-CNN-NERE model.

6.5 Summary and Discussion

We propose WeLT-TablERT-CNN approach, a cost-sensitive JNERE with 2D CNNs as a table-filling problem. This method is adapted from TablERT-CNN (Ma et al., 2022) that represents each table cell as pixel and each table as a 2D image. Although TablERT-CNN does not apply the strong sampling strategy as proposed by SpERT (Eberts and Ulges, 2020), the table-filling approach adopts the token-level labels for NER and directed relation labels which are imbalanced as discussed in Section 6.4. To this end, three different joint loss functions using WeLT were proposed as presented in Section 6.3. Experiments conducted on CoNLL04, ADE, and SciERC demonstrated that modest performance of the WeLT-TablERT-CNN variants compared to the two baselines: TablERT, and TablERT-CNN. The results prove the advantages of addressing the class problem.
The common patterns based on the experimental results for the three datasets are as follows:

- For the CoNLL04 dataset, the WeLT-TablERT-CNN variants show modest performance improvements. The TablERT-CNN-NERE variant achieves the second-best scores in RE (73.6 %) and RE+ (73.4 %), closely trailing the TablERT model with BERT$_{\text{LARGE}}$, which has the best scores in these metrics with F1 scores of (73.8 %) for both RE and RE+. The highest NER score is achieved by the TablERT-CNN-RE variant (90.9 %), surpassing all other models.

- In the ADE dataset, the TablERT-CNN-NER model outperformed all other models across all metrics, achieving scores of 91.7 % in NER and 85.7 % in RE and RE+. Compared to the baselines TablERT and TablERT-CNN, it achieved a higher performance by 1.8 % in NER and by 5.1 % in RE and RE+, highlighting the effectiveness of WeLT.

- For the SciERC dataset, the results indicate that the WeLT-TablERT-CNN variants provide competitive performance. Using BERT$_{\text{BASE}}$, the WeLT-Tab-NER-RE variant achieves the highest RE score (42.6 %). When using SciBERT, the TablERT-CNN-RE model attains

the highest RE score (46.6 %), while the NER scores are closely matched between the TablERT-CNN and TablERT-CNN-RE models.

The WeLT-TablERT-CNN variant models exhibit modest performance across various datasets and evaluation metrics, validating the effectiveness of the proposed modifications. While using TablERT with $\text{BERT}_{\text{LARGE}}$ achieves the highest scores in CoNLL04's RE and RE+ tasks, the WeLT variants provide competitive results with fewer parameters, offering an advantage in computational efficiency.

7 Conclusions and Future Work

With the rapid growth of unstructured text from diverse sources such as scientific articles and news, information extraction has become critically important for transforming this vast and disorganised data into structured and actionable insights. Named Entity Recognition (NER) and Relation Extraction (RE) are key components of information extraction. NER facilitates the structuring of data by categorising key pieces of information, while RE adds another layer of understanding by connecting entities with meaningful relationships. Therefore, developing a robust information extraction pipeline is essential for constructing applications such as knowledge graphs, information retrieval systems, and natural language understanding systems.

Class imbalance is a common issue in supervised learning, where certain classes (i.e., categories of data) are under-represented in the training data compared to other classes. This imbalance may degrade the performance on the under-represented classes, as the model may become biased towards predicting the majority classes, which it encounters more frequently. In this work, we have sought to address the class imbalance problem in supervised learning. In this concluding chapter, we briefly reiterate our contributions in Section 7.1, along with a discussion of potential future work in Section 7.2.

7.1 Key Insights

First, we highlighted the need for a highly accurate and robust information extraction pipeline to process large volumes of unstructured data described in Chapter 1. Continuously improving the automation of entity identification and relationship extraction is important, as converting unstructured data into structured formats enables better search, analysis, and decision-making. Moreover, effective information extraction significantly eases the tedious work of data annotators and curators by providing pre-annotated data, thereby reducing the amount of manual effort required. This allows annotators to focus on refining the extracted entities and their relationships, rather than starting from scratch, enabling them to concentrate on more complex, high-value aspects of the annotation and curation process, ultimately leading to more efficient, scalable, and higher-quality data management.

In Chapter 2, we introduced the fundamental concepts of information extraction, focusing on the practical requirements of named entity recognition and relation extraction. We have

devoted considerable effort to establish the essential foundations and providing the necessary background. Additionally, we highlight different types of named entities and the distinctive characteristics of relationships. Beyond presenting a formal model of information extraction, we reviewed the various paradigms of joint named entity recognition and relation extraction. Furthermore, we explored several gold-standard datasets that exhibit class imbalance, which is one of the central challenges in NER and RE. The research community has made significant efforts to develop gold-standard datasets to facilitate the creation of supervised prediction models, yet these real-world datasets are often highly skewed. We also examined the conventional evaluation metrics used for relevant information extraction tasks. As we conclude the motivating chapters, we consider the evolution of pre-trained language models, noting that standard fine-tuning approaches may struggle to identify and extract under-represented entities and relationships.

In Chapter 3, we explored the state-of-the-art in flat biomedical named entity recognition, focusing particularly on pre-trained language models. Based on leading performance scores, we selected encoder-only models as the best-performing approach. Despite their high scores, vanilla fine-tuning often overlooks the potential for improvement of class distribution imbalances in biomedical datasets. Given the highly imbalanced nature of gold-standard datasets in this field, we emphasise the potential need to address these imbalances to improve model performance. We highlighted three state-of-the-art approaches to address class imbalance: data-level, algorithmic-level, and hybrid methods. We argue that traditional data sampling techniques are insufficient, as duplicating training data tends to degrade language model performance (Lee et al., 2022). To address these challenges, we propose a cost-sensitive approach that modifies the loss function in encoder-only models. As our first contribution, we introduce a new loss function, the **We**ighted **L**oss **T**rainer (WeLT). WeLT incorporates coefficients that re-scale class weights based on the inverse relative frequencies of the classes. Unlike other existing loss functions, such as those based on the inverse number of samples or the effective number of samples, WeLT is the first loss function to re-scale class weights according to their frequency within the overall class distribution.

7.1.1 WeLT's Application

We evaluate the performance of our proposed cost-sensitive fine-tuning approach, applying WeLT to various downstream tasks and domain applications. Our efforts led to:

- *Single-label biomedical named entity recognition*: we evaluated the impact of addressing class imbalance via WeLT, comparing it against three existing weighting schemes across eight biomedical datasets in Chapter 3. In this study, we fine-tuned 280 models, including vanilla fine-tuning, and two weighting schemes: inverse number of samples and inverse square root of the number of samples. Additionally, we tested the effective number of samples, applying lower, median, and upper bounds of hyperparameters for a fair comparison. Our results

demonstrate the positive effect of addressing class imbalance, with WeLT and comparable weighting schemes consistently outperforming vanilla fine-tuning, which neglects class distribution.

- *Impact of WeLT-recognised entities on biomedical entity linking*: as an extension of our work in Chapter 3, we investigated the effect of re-scaled class weights on entity recognition via WeLT. In collaboration with Pedro Ruas and Francisco M. Couto during a secondment at LASIGE, University of Lisbon, we evaluated WeLT-recognised entities and compared them to vanilla fine-tuning results. Our findings show that addressing the class imbalance problem in named entity recognition positively enhances the overall performance of the named entity linking task.

- *Overlapping and nested named entity recognition*: building on the success of WeLT in flat or single-label biomedical named entity recognition and linking, we extended our cost-sensitive approach to other biomedical datasets and non-biomedical domains. We aimed to explore the impact of addressing class imbalance on additional downstream tasks. In Chapters 4 and 5, we adapted the original span classifiers proposed by Eberts and Ulges (2020) and Jianquan Ouyang (2022), respectively, by incorporating WeLT's weighted cross-entropy loss to address class imbalance. Our experiments show that WeLT-based span classifiers outperform the original models. This improvement was observed on general domain datasets such as CoNLL04 from newswire articles, as well as on biomedical datasets with overlapping entities, such as the ADE dataset.

- *Joint named entity recognition and relation extraction*: after successfully applying WeLT to flat, nested, and overlapping named entities, as well as testing it on biomedical entity linking, we were motivated to extend WeLT to relation extraction tasks. Throughout Chapters 4 to 6, we proposed different variants of relation classifiers that address class imbalance by introducing various WeLT-based joint loss functions using two distinct paradigms:

  - *Span-based*: in Chapter 4, we tackled the class imbalance problem in the SpERT model (Eberts and Ulges, 2020). In this context, the imbalance arises not only from the class distribution but also from SpERT's strong negative sampling strategy that adds non-entities and relations. To address this imbalance problem, we adapted SpERT by incorporating the WeLT loss function for the span classifier to recognise entities, as introduced in Chapter 3. For the relation classifier, we designed a new loss function that accounts for both the frequency of entity pairs and the frequency of the relation class itself. We compared four different loss functions against the original SpERT. Our results demonstrate that re-scaling class weights for both named entities and relations via WeLT leads to superior performance compared to the original SpERT model.

Jianquan Ouyang (2022) addressed certain limitations of SpERT's span classifier, proposing the ASpERT model. ASpERT introduces attention mechanisms that enhance boundary supervision and utilises a multi-layer perceptron to improve span filtering. While ASpERT outperforms SpERT, it still relies on the same negative sampling strategy. Therefore, in Chapter 5, we applied WeLT loss functions to ASpERT. For a fair comparison, we used the same experimental settings and hyperparameters reported for ASpERT. Our results show that the best-performing model incorporates WeLT-re-scaled entities, proving that WeLT loss functions effectively mitigate class imbalance and enhance overall performance in joint entity and relation extraction.

– *Table-filling*: we applied our WeLT approach to tackle class imbalance in span-based joint named entity and relation extraction models, demonstrating its effectiveness in improving overall model performance. Recently, significant efforts have focused on the table-filling approach. For instance, Ma et al. (2022) proposed TablERT-CNN, a model that jointly extracts entities and relations using stacked convolutional neural networks. TablERT-CNN treats named entity recognition as a sequential labelling task using the BILOU tagging scheme, but it does not account for overlapping entities. Despite this limitation, TablERT-CNN outperforms SpERT by removing strong negative sampling.

  Building on the success of WeLT in Chapter 3 for flat named entity recognition as a sequence-labelling task, we extended our method to TablERT-CNN. We proposed three WeLT-based loss functions that re-scale class weights to balance entities, relations, or both in Chapter 6. We used the same experimental settings as TablERT-CNN and included an additional dataset, SciERC, which contains scientific texts from the computer science domain with complex class distributions and relationships involving multiple entity arguments. Our results showed that the proposed WeLT approach achieved relative improvements over the original TablERT-CNN. However, in some cases, addressing class imbalance did not have the expected positive impact, revealing a bottleneck and leaving room for further exploration in future work.

In the following section, we present an extensive evaluation of our proposed JNERE WeLT-based models, comparing them with other PLMs, including recent LLMs.

7.1.2 Performance of WeLT-Based Models Against Comparable Models

In this comparative analysis, we evaluate the performance of the proposed WeLT-based models developed in Chapters 4 to 6 against a diverse set of models, as discussed in Sections 2.3 and 3.2.1, including encoder-only, encoder-decoder, and decoder-only architectures, as follows:

- Deeper Task-Specificity (Crone, 2020): this novel neural architecture introduces additional task-specific bidirectional recurrent neural network (BiRNN) layers for both NER and RE tasks. The model allows for deeper task-specificity by tuning the number of shared and task-specific layers independently for different datasets.
- REBEL (Huguet Cabot and Navigli, 2021): an auto-regressive sequence-to-sequence model. The authors employ BART-large (Lewis et al., 2020) as the base model and utilise various encoder models, such as bert-base-cased, albert-xxlarge-v1, and scibert-scivocab-uncased.
- PFN (Yan et al., 2021): a partition filter network that encodes task-specific features in joint entity and relation extraction. This model uses a partition filter encoder to generate these task-specific features jointly.
- Boundary Assembling (Tang et al., 2022): this model integrates entity boundary detection, span classification, and relation extraction into an end-to-end framework. It employs albert-xxlarge-v2 as the encoder and assembles entity boundaries to enumerate entity spans.
- SpERT.MT (Xue and Lu, 2023): a multi-task learning model that incorporates the intersection over union concept. It introduces positional information into the entity classifier for span boundary detection and integrates entity logits into the embedded representation of entity pairs.
- Table-Sequence (Wang and Lu, 2020): a novel approach using two encoders as follows: one for tables and another for sequences that assist each other in the representation learning process.
- Translation between Augmented Natural Languages (TANL) (Paolini et al., 2021): a unified text-to-text approach for structured prediction, functioning as a translation task between augmented natural languages.
- Flan T5 (Large) (Chung et al., 2022): a large language model trained using instruction fine-tuning, utilising chain-of-thoughts (CoT) generations produced by the T5 model.
- FlanT5+GPT-3-generated CoT (Wadhwa et al., 2023): a fully supervised, fine-tuned T5 model that incorporates chain-of-thoughts style explanations generated by GPT-3.
- In-Context GPT-3 (Brown et al., 2020): an autoregressive language model with 175 billion parameters, trained using in-context learning.
- GPT-3.5 distilled (Gu et al., 2023): a significantly smaller variant of GPT-3.5, employing the PubMedBERT model. This model is 1,000 times smaller than the original GPT-3.5.

- Multi-turn QA (Li et al., 2019a): this approach applies BERT as the core model in a question-answering setting, where entity- and relation-specific questions guide the model to identify head and tail entities.
- Two-Phase Paradigm Bin Ji (2023): this approach reduces the gap between negative entities and other predefined entities, as well as between relations. It incorporates a gated mechanism for effectively fusing various semantic representations.
- PURE Zhong and Chen (2021): a simple yet effective model for end-to-end relation extraction, employing two encoders as follows: one for entity extraction and another for relation extraction.
- SpERT.PL Santosh et al. (2021): a deep neural model that leverages part-of-speech information and entity logits to boost classification performance.

The analysis of NER and RE results across various models, particularly the proposed WeLT models, is outlined below. The discussion focuses on F1 scores. Additionally, model sizes, expressed in terms of the number of training parameters, are considered as a critical factor in evaluating the trade-off between performance and computational efficiency.

Regarding the CoNLL04 dataset, the macro-averaged F1 scores are presented in Table 7.1. The SpERT-NERE model attained the highest NER F1 score of 87.70 %, slightly surpassing Table-Sequence, which achieved an F1 score of 86.90 %. The SpERT-pos-weight model followed closely with an NER F1 score of 86.49 %. In terms of RE, Boundary Assembling achieved the highest F1 score of 76.70 %, with REBEL following closely with an F1 score of 76.65 %. On the other hand, SpERT-pos-weight falls behind, with an F1 score of 73.56 %. It is worth mentioning that the WeLT-SpERT models use 102M training parameters, which is significantly fewer than the 235M parameters required by Boundary Assembling, the 223M required by Table-Sequence and the 760M required by REBEL. This demonstrates that the WeLT models can deliver competitive NER and RE results while being much more computationally efficient.

			NER			RE		
Method	Encoder	Parameters	Precision	Recall	F1	Precision	Recall	F1
SpERT Eberts and Ulges (2020)	$BERT_{\text{BASE}}$	102 M	85.78	86.84	86.25	74.75	71.52	72.87
ASpERT Jianquan Ouyang (2022)	$BERT_{\text{BASE}}$	102 M	86.57	85.49	85.97	74.92	69.01	71.66
REBEL Huguet Cabot and Navigli (2021)	BART	**760 M**	-	-	-	-	-	76.65
Table-Sequence Wang and Lu (2020)	$ALBERT_{XXLARGE_{v1}}$	223 M	-	-	86.90	-	-	75.40
Boundary Assembling Tang et al. (2022)	$ALBERT_{XXLARGE_{v2}}$	235 M	88.50	85.40	86.80	77.50	76.30	**76.70**
SpERT-NERE	$BERT_{\text{BASE}}$	102 M	86.07	89.46	**87.70**	64.26	74.44	68.77
SpERT-pos-weight	$BERT_{\text{BASE}}$	102 M	85.37	87.65	86.49	76.65	70.99	73.56

Table 7.1: Comparison of macro-averaged CoNLL04 test scores between various language models and WeLT-based models. The best F1 scores are in bold, with the second-best underlined. For training parameters, the largest are in bold and the smallest are underlined.

Method	Encoder / Pre-trained LM	Parameters	NER			RE		
			Precision	Recall	F1	Precision	Recall	F1
SpERT Eberts and Ulges (2020)	$BERT_{\text{BASE}}$	102 M	88.25	89.64	88.94	73.04	70.00	71.47
TablERT-CNN Ma et al. (2022)	$BERT_{\text{BASE}}$	110 M	-	-	90.50	-	-	73.20
SpERT.MT Xue and Lu (2023)	$BERT_{\text{BASE}}$	102 M	90.13	91.27	90.70	75.36	71.94	73.61
Multi-turn QA Li et al. (2019b)	$BERT_{\text{BASE}}$	108 M	89.00	86.60	87.80	69.20	68.20	68.90
Boundary Assembling Tang et al. (2022)	$ALBERT_{XXLARGE_{v2}}$	235 M	90.80	88.80	89.80	76.60	74.40	75.50
Table-Sequence Wang and Lu (2020)	$ALBERT_{XXLARGE_{v1}}$	223 M	-	-	90.10	-	-	73.60
TablERT Ma et al. (2020)	$BERT_{\text{BASE}}$	110 M	-	-	90.20	-	-	72.60
TablERT Ma et al. (2020)	$BERT_{\text{LARGE}}$	345 M	-	-	90.50	-	-	73.80
ASpERT Jianquan Ouyang (2022)	$BERT_{\text{BASE}}$	102 M	89.03	88.53	88.77	73.62	67.39	70.36
TANL Paolini et al. (2021)	T5-base	220 M	-	-	89.40	-	-	71.40
TANL (Multi-task) Paolini et al. (2021)	T5-base	220 M			89.80			72.60
REBEL Huguet Cabot and Navigli (2021)	BART	460 M	-	-	-	-	-	75.40
Flan T5 (Large) Chung et al. (2022)	Flan-T5–LARGE	780 M	-	-	-	-	-	75.28
+ GPT-3-generated CoT Wadhwa et al. (2023)	Flan-GPT-3-generated-CoT	760 M	-	-	-	-	-	**80.76**
In-Context GPT-3 Brown et al. (2020)	GPT-3	**175 B**	-	-	-	-	-	76.53
+ CoT Wadhwa et al. (2023)	GPT-3	**175 B**	-	-	-	-	-	78.18
Flan T5 (Large) w/ CoT Explanations and reference labels generated from GPT-3 Wadhwa et al. (2023)	Flan-GPT-3-generated-CoT	760 M	-	-	-	-	-	76.13
TablERT-CNN-NERE	$BERT_{\text{BASE}}$	110 M	-	-	90.80	-	-	73.40
TablERT-CNN-RE	$BERT_{\text{BASE}}$	110 M	-	-	**90.90**	-	-	71.90

Table 7.2: Comparison of micro-averaged CoNLL04 test scores between various language models and WeLT-based models. The best F1 scores are in bold, with the second-best underlined. For training parameters, the largest are in bold and the smallest are underlined.

Regarding the CoNLL04 dataset, the micro-averaged F1 scores are presented in Table 7.2. The TablERT-CNN-RE model achieved the highest NER F1 score of 90.90 %, and TablERT-CNN-NERE follows closely with an F1 score of 90.90 %. Both models marginally outperform SpERT.MT, TablERT-CNN ($BERT_{BASE}$), and TablERT ($BERT_{LARGE}$), which achieved F1 scores of 90.70 %, 90.50 %, and 90.50 %, respectively.
For RE, the highest F1 score is 80.76 %, achieved by the Flan T5+GPT-3-generated CoT, followed by In-Context GPT-3+CoT with an F1 score of 78.18 %. On the other hand, TablERT-CNN-NERE falls behind by a large margin, with an F1 score of 73.40 %. When comparing the models with respect to training parameters, In-Context GPT-3+CoT has the largest number of parameters at 175B, followed by Flan T5+GPT-3-generated CoT with 760M. In contrast, the WeLT models, such as TablERT-CNN-NERE and TablERT-CNN-RE, both use 110M parameters, highlighting that the WeLT models provide competitive NER and RE performance with a significantly lower computational cost.

Regarding the SciERC dataset, the micro-averaged F1 scores are presented in Table 7.3. The TablERT-CNN-RE model achieved an NER F1 score of 68.60 % and an RE F1 score of 48.90 %, which falls behind the top-performing models such as SpERT.MT, which has an NER F1 score of 73.22 % and an RE F1 score of 53.72 %. In this case, SpERT.MT remains the best choice.

			NER			RE		
Method	Encoder	Parameters	Precision	Recall	F1	Precision	Recall	F1
SpERT Eberts and Ulges (2020)	SciBERT	102 M	70.87	69.79	70.33	53.40	48.54	50.84
Two phase Paradigm Bin Ji (2023)	$BERT_{BASE}$	110 M	69.70	72.30	71.00	52.90	52.50	52.70
PURE Zhong and Chen (2021)	SciBERT	110 M	-	-	66.60	-	-	48.20
Syntax-informed multi-head self-attention Zhang et al. (2021b)	$BERT_{BASE}$	110 M	69.70	71.10	70.40	55.30	50.00	52.50
SpERT.PL Santosh et al. (2021)	$BERT_{BASE}$	102 M	69.80	71.30	70.50	51.90	50.60	51.30
Boundary Assembling Tang et al. (2022)	$ALBERT_{XXLARGE_{v2}}$	**235 M**	62.40	67.10	64.70	56.60	48.20	52.10
SpERT.MT Xue and Lu (2023)	$BERT_{BASE}$	102 M	71.21	75.35	**73.22**	53.63	53.81	**53.72**
TablERT-CNN-RE	SciBERT	110 M	-	-	68.60	-	-	48.90

Table 7.3: Comparison of micro-averaged SciERC test scores between various language models and WeLT-based models.The best F1 scores are in bold, with the second-best underlined. For training parameters, the largest are in bold and the smallest are underlined.

Regarding the ADE dataset, the macro-averaged F1 scores are presented in Table 7.4. The SpERT-NERE model achieved competitive performance with the second-highest NER F1 score of 92.37 %, while Boundary Assembling achieved the best F1 score of 92.50 %.

Concerning the performance of RE, TablERT-CNN-RE achieved the second-best score of 85.70 %, while the Flan T5 (Large)+GPT-3-generated CoT achieved an F1 score of 92.17 %. In terms of parameters, SpERT-NERE requires only 102M, significantly fewer than the 760M needed for the

Flan T5 (Large)+GPT-3-generated CoT. This again illustrates the competitive performance of WeLT models with fewer computational resources.

Across the three datasets, the WeLT models deliver modest performance in both NER and RE tasks, often ranking among the best or second-best models in terms of F1 scores. The parameter efficiency of WeLT models, typically around 102M to 110M, is a significant advantage over competing models that require upwards of 235M to 175B parameters. This balance of high performance and lower computational demands makes the WeLT models particularly attractive for deployment in environments where resource efficiency is critical. Overall, the WeLT models demonstrate a strong ability to maintain competitive and modest performance while also offering significant savings in model size, making them a compelling choice for both NER and RE tasks.

Method	Encoder	Parameters	NER			RE		
			Precision	Recall	F1	Precision	Recall	F1
SpERT Eberts and Ulges (2020)	$BERT_{\text{BASE}}$	102 M	89.26	89.26	89.25	78.09	80.43	79.24
Relation-MetricTran and Kavuluru (2019)	LSTM+CNN	-	86.16	88.08	87.11	77.36	77.25	77.29
ASpERTJianquan Ouyang (2022)	$BERT_{\text{BASE}}$	102 M	90.96	91.87	91.41	81.65	83.92	82.76
REBELHuguet Cabot and Navigli (2021)	BART	760 M	-	-	-	-	-	82.20
PFN Yan et al. (2021)	$ALBERT_{XXLARGE_{v1}}$	223 M	-	-	91.30	-	-	83.20
PFN Yan et al. (2021)	$BERT_{\text{BASE}}$	110 M	-	-	89.60	-	-	80.00
Boundary AssemblingTang et al. (2022)	$ALBERT_{XXLARGE_{v2}}$	235 M	93.60	91.50	**92.50**	85.40	85.00	85.20
SpERT.MT Xue and Lu (2023)	$BERT_{\text{BASE}}$	102 M	91.35	92.77	92.05	81.06	86.55	83.72
Table-SequenceWang and Lu (2020)	$ALBERT_{XXLARGE_{v1}}$	223 M	-	-	89.70	-	-	80.10
TANL Paolini et al. (2021)	T5-base	220 M	-	-	-	-	-	80.61
TANL (Mult-task) Paolini et al. (2021)	T5-base	220 M	-	-	-	-	-	80.00
REBEL Huguet Cabot and Navigli (2021)	BART	460 M	-	-	-	-	-	82.21
Flan T5 (Large) Chung et al. (2022)	Flan-T5–LARGE	780 M	-	-	-	-	-	83.15
+ GPT-3-generated CoT Wadhwa et al. (2023)	Flan-GPT-3-generated-CoT	760 M	-	-	-	-	-	**92.17**
In-Context GPT-3 Brown et al. (2020)	GPT-3	**175 B**	-	-	-	-	-	82.66
TablERT Ma et al. (2020)	$BERT_{\text{BASE}}$	110 M	-	-	89.90	-	-	80.60
TablERT-CNN Ma et al. (2022)	$BERT_{\text{BASE}}$	110 M	-	-	89.70	-	-	80.50
GPT-3.5 distilled-PubMedBERT Gu et al. (2023)	$BERT_{\text{BASE}}$	110 M	-	-	-	-	-	84.27
GPT-3.5 distilled-PubMedBERTGu et al. (2023)	$BERT_{LARGE}$	345 M	-	-	-	-	-	84.53
SpERT-NERE	$BERT_{\text{BASE}}$	102 M	92.13	92.60	92.37	83.82	86.60	85.19
TablERT-CNN-RE	$BERT_{\text{BASE}}$	110 M	-	-	91.70	-	-	85.70

Table 7.4: Comparison of macro-averaged ADE test scores between various language models and WeLT-based models. The best F1 scores are in bold, with the second-best underlined. For training parameters, the largest are in bold and the smallest are underlined.

7.1.3 Final Assessment of WeLT Models

WeLT adjusts class weights for cost-sensitive learning to address class imbalance. As previously mentioned, the datasets are skewed, and the use of standard loss functions, which assume that error costs are equal for all classes, often leads to biased models. WeLT is a simple approach that adjusts class weights, providing a straightforward solution to assign higher loss contributions without needing to modify the model architecture, as it is typically implemented at the loss function level. This flexibility enables WeLT models to be adaptable across a wide range of architectures. WeLT does not involve augmenting the datasets with synthetic samples from the minority class, which may lead to potential overfitting, and reduces the risk of inflating the data size, which could increase training time. In contrast, WeLT models have the same training costs as vanilla models.
Despite adjusting the class weights by setting higher weights for minority classes and penalising the majority classes, WeLT still offers modest improvements. We suggest some justifications based on overall performance observations:

- Although in WeLT we normalised class weights using the Softmax function to avoid discrepancies in new rescaled weights between minority and majority classes, the emphasis on boosting the performance of underrepresented classes by penalising misclassification more heavily for those classes may inadvertently reduce the focus on correctly classifying instances of majority classes. While the performance of minority classes improves, the performance of majority classes may drop slightly, resulting in only modest overall improvements.

- Adjusting class weights mainly affects the decision threshold by increasing the importance of the minority class. However, this does not introduce fundamentally new information to improve feature extraction. Thus, the performance gains become marginal. Moreover, for datasets with less pronounced imbalances, such as CoNLL04, the improvements are less significant, and in some cases, the baselines remain the best choice. In contrast, the ADE dataset and other highly skewed datasets in Chapter 3 show more consistent gains, especially for the NER task.

- In noisy datasets, particularly in tasks with manual labelling, minority class labels often have a higher error rate. Adjusting class weights emphasises these noisy instances, making the model disproportionately affected by label noise in the minority classes. Hence, this may degrade overall performance by making the model sensitive to incorrectly labelled instances, particularly when working with noisy real-world datasets.

In summary, the improvements observed throughout Chapters 3 to 6, though modest, suggest that WeLT-based approaches effectively contribute to better entity recognition and relation

extraction. However, they do not enhance the feature extraction capability of the model, which is often crucial for achieving significant gains.

7.2 Outlook

While several clear pathways for future research are evident, predicting the immediate next steps is challenging, given the swift pace of progress in NLP. Notably, even during the course of writing this thesis, we have witnessed multiple iterations and innovations addressing some of the key limitations outlined in this work. We believe that future research should focus on improving modelling techniques, with an emphasis on the extraction of document-level relations and the establishment of benchmarks for class imbalance in NLP:

- Enhancing entity and relation predictions with human-in-the-Loop: in our work, we have released fine-tuned biomedical named entity recognition and joint entity-relation extraction models. A promising extension involves using these models to pre-annotate text, which can then be reviewed by domain experts. This process can be supported by the INCEpTION (Klie et al., 2018) semantic annotation platform, which leverages external recommender systems for annotation prediction. INCEpTION's uncertainty sampling technique aids in selecting and presenting uncertain examples to human annotators. This human-in-the-loop approach enhances model performance, reduces the annotation workload, and ensures the model effectively learns from the most challenging and informative data points.

- Potential improvements for large language models in information extraction: as discussed in Chapter 3, LLMs still struggle to surpass the state-of-the-art scores achieved by encoder-only models. We propose that this difficulty may stem from the limitations of zero-shot and few-shot learning in effectively training LLMs for classification and information extraction tasks. Therefore, we suggest incorporating additional domain-specific knowledge sources, such as ontologies and taxonomies, into instruction-based techniques to potentially enhance LLM performance for these tasks.

- Knowledge distillation: in Chapters 4 to 6, we presented fine-tuned WeLT models for joint named entity and relation extraction. A promising extension of this work could involve applying the knowledge distillation framework outlined by Gu et al. (2023). This framework uses GPT-3.5 as a teacher model for self-supervision and employs PubMedBERT and BioGPT as student models that learn from the teacher's self-supervised labels. Similarly, we suggest that our WeLT-based models could serve as effective teacher models. Given that these models address class imbalance issues previously overlooked, they may offer

further performance enhancements. Additionally, exploring domain-specific LLMs, such as BioMedLM, as teacher models, could be a valuable direction for future research.

- Establishing a benchmark for class imbalance in NLP: at present, there is no established benchmark specifically addressing class-imbalanced settings (Henning et al., 2023). This absence makes it challenging to compare evaluation results consistently across studies, unlike the widely recognised benchmarks such as the General Language Understanding Evaluation (Wang et al., 2018) and the Biomedical Language Understanding Evaluation (Peng et al., 2019). While F1 scores remain the predominant metric in NLP, in scenarios involving class imbalance, it is essential to supplement these scores with per-class performance metrics to ensure a more comprehensive evaluation.

Finally, while significant efforts have been dedicated to sentence-level relation extraction, we believe that relatively few attempts have addressed the use of document-level context (Christopoulou et al., 2019; Zhou et al., 2021; Le et al., 2022). In our view, one of the key future directions should involve adjusting model architectures to extend beyond sentence-level analysis and incorporate co-reference resolution.

Acronyms

2D image	Two-dimensional Image
2D table	Two-dimensional Table
2D-CNNs	Two-dimensional Convolutional Neural Networks
ACD	Attentional Contribution Algorithm
ACE	Angiotensin-converting Enzyme
ACL	Association for Computational Linguistics
ADE	Adverse Drug Events
AMCA	Tranexamic Acid
APIs	Application Programming Interface
ARTHS	Arboleda-Tham Syndrome
ASpERT	Attention and Span-based Entity and Relation Transformer
BBFL	Batch-balanced Loss
BC2GM	BioCreative II Gene Mention
BC4CHEMD	BioCreative IV Chemical and Drug
BC5CDR	BioCreative V Chemical Disease Relation corpus
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-term Memory
BioCreative	Critical Assessment of Information Extraction Systems in Biology
BioIE	Biomedical Information Extraction
BioNEL	Biomedical Named Entity Linking
BioNER	Biomedical Named Entity Recognition
BioNLP	Biomedical Natural Language Processing
BioNLP-ST	BioNLP Shared Tasks
BioPLMs	Biomedical Pre-trained Language Models
BioRE	Biomedical Relation Extraction
BioRED	Biomedical Relation extraction dataset
BiRNN	Bidirectional Recurrent Neural Network
BLEU	Bilingual Evaluation Understudy Score

BPE	Byte-pair Encoding
CAS	Class-aware Sampling
CDR	Chemical Disease Relation
ChatGPT	Chat Generative Pre-trained Transformer
CID	Chemical-induced Disease
CLMLE	Cluster-based Large-margin Local Embedding
CNN	Convolutional Neural Network
CoSen CNN	Cost-sensitive Convolutional Neural Network
CoT	Chain-of-thoughts
CTD Taxonomy	Comparative Toxicogenomics Database
DL	Deep Learning
DSPT	Domain-specific Pre-training
DyGIE	Dynamic Graph IE
EDA	Easy Data Augmentation
EE	Event Extraction
ENS	Effective Number of Samples
ER-LAC	Span-based Joint Entity and Relation Extraction Model with Multi-level Lexical and Attention on Context Features
F1	F1 Score
FL	Focal Loss
FN	False Negatives
GDA	Gene Disease Associations
GPT	Generative Pre-trained Transformers
ICL	In-context Learning
IE	Information Extraction
IFT	Instruction Fine-tuning
iNERD	Informed Named Entity Recognition Decoding
INS	Inverse of the Number of Samples
IOB	Inside–outside–beginning Format
IR	Imbalance Ratio
ISNS	Inverse of the Square Root of the Number of Samples
JNERE	Joint Entity and Relation Extraction
LDAM	Label-distribution-aware Margin Loss
LLMs	Large Language Models
LMLE	Large Margin Local Embedding
LMs	Language Models
LoRA	Low-rank Adaption of Large Language Models

LSTM	Long Short-term Memory
MASS	MAsked Sequence to Sequence Pre-training
MEDIC	Medical dictionary for regulatory activities
MeSH	Medical Subject Headings
Meta Llama	Large Language Model Meta AI
MIMIC-III	Medical Information Mart for Intensive Care
MLM	Masked Language Modelling
MLP	Multi-layer Perceptron
NCBI	National Centre for Biotechnology Information
NER	Named Entity Recognition
NestedNER	Nested Entity Recognition
NLM	National Library of Medicine
NLP	Natural Language Processing
NSP	Next Sentence Prediction
OMIM	Online Mendelian Inheritance in Man
OOD	Out-of-distribution
OOV	Out-of-vocabulary
OverlapNER	Overlapping Entities
P	Precision
PaLM	Pathways Language Models
PAS	Performance-based Sampling
PEFT	Parameter-Efficient Fine-tuning
PLMs	Pre-trained Language models
PMC	PubMed Central
PPI	Protein-protein Interaction
PubMed	PubMed Abstracts
R	Recall
RE	Relation extraction
ReLU	Rectified Linear Unit
RETRO	Retrieval Enhanced Transformer
RLHF	Reinforcement Learning from Human Feedback
ROS	Random Oversampling
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
RUS	Random Undersampling
Seq2Seq	Sequence-to-sequence Model
SFT	Supervised Fine-tuningg
SOTA	State-of-the-art

SpERT	Span-based Entity and Relation Transformer
SwiGLU	Swish-Gated Linear Unit
T5	Text-to-Text Transfer Transformer
TANL	Translation between Augmented Natural Languages
TP	True Positives
UMLS	Unified Medical Language System Semantic Network
WBCE	Weighted Binary Cross-Entropy
WCE	Weighted Cross-Entropy Loss
WeLT	**We**ighted **L**oss **T**rainer
WP	WordPiece
XLMs	Cross-lingual Language Model

Glossary

pr	A vector of relative class frequencies
$f_a(s)$	Attentional contribution degree
ES	BERT embedded sequence
b^r	Bias term for relation classification layer
b^s	Bias vector for span classifier
B	Billion
BCE	Binary cross-entropy loss
α_j	Weight associated to class j
$e_{[CLS]}$	Classifier token
$1 - \frac{n_j}{\sum_{k=1}^{c} n_k}$	The complement of the relative frequency of class j compared to the overall frequency of all classes
$\circ$	Concatenation
α	Confidence threshold
Θ	Contribution degree threshold
CE	Cross-entropy loss
d	Document
d_1	Embedding dimension
e_j	Embedding vector
e_i	Entity type in the document $d \in \mathcal{E}$
$F1_i$	F1 score for class i
x^s	Final input to the span classifier
n_e	Fixed number of random non-entity spans
n_r	Fixed number of random non-relation samples
f	Fusion function via max-pooling
e_h	Head entity
n_{head_j}	Total number of instances where the entity appears as the head of the relation
$\mathcal{L}^r$	Loss of relation classifier
$\mathcal{L}^s$	Loss of the span classifier

$mask_s$	Mask score
max	Max-pooling function
f_l	Max-pooling fusion function in ASpERT
M	Million
p	Model prediction vector
n_j	Number of instances in class j
ns	Number of spans
P_i	Percision for class i
S^{gt}	Positive span sample
$\mathcal{E}$	Predefined set of entity types/categories
$\mathcal{R}$	Predefined set of relation types/categories
$\hat{y}^s$	The predicted posterior probability distribution over each entity class (including the none class) for the span s
$\hat{y}^r_{1/2}$	The predicted posterior probability distribution over each relation class (including the none class) for the span s
p_j	Predicted probabilty for class j
R_i	Recall for class i
nr	The number of instances of the relation class indexed by j
r_i	Relation type in the document
W^r	Relation's weight matrix
w_e	Rescaled entity weights
w_r	Rescaled relation weights
d_i	Sentence
$\mathcal{T}$	Sequence of tokens in the document
$\mathbb{R}^{d_1}$	Set of all d_l dimensional vectors
$\mathcal{A}$	Set of all possible annotations in the document
$\mathcal{S}$	Set of Spans
γ	Sigmoid function
σ	Softmax activation function
l_s	Span's length
$k+1$	Span width
$e(s)$	Span's embedding
ε	Span's variable length threshold
W^s	Span's weight matrix
s	Span
Y	Table-filling matrix
e_t	Tail entity

n_{tail_j}	Total number of instances where the entity appears as the tail of the relation
y	Target vector
IR_j	The imbalance ratio for class j
d_a	The number of BERT's attention heads
k_s	The number of entity classes (including none)
m	The number of hidden layer units of the MLP
K	Thousand
t_i	Token in the document
n_k	Total class frequency
c	Total number of entity types/classes
ps	Total number of possible spans
r	Total number of relation classes
n	Total number of tokens
td	Training Datasets
y_j	True label for class j
β	Tunable focusing parameter
W^s	The weight matrix of the Softmax classifier
w_k	Width embedding
pr_j	is the proportion of samples in class j with respect to the total class frequency n_k

Bibliography

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021. Better Fine-Tuning by Reducing Representational Collapse. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Asifuddin Nasiruddin Ahmed and Ravinder Saini. 2023. Detection of Credit Card Fraudulent Transactions Utilizing Machine Learning Algorithms. In *2023 2nd International Conference for Innovation in Technology (INOCON)*, pages 1–5.

Abbas Akkasi, Ekrem Varoğlu, and Nazife Dimililer. 2018. Balanced undersampling: a novel sentence-based undersampling method to improve recognition of named entities in chemical and biomedical text. *Applied Intelligence*, 48(8):1965–1978.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The Falcon Series of Open Language Models. *CoRR*, abs/2311.16867.

Shin Ando and Chun-Yuan Huang. 2017. Deep Over-sampling Framework for Classifying Imbalanced Data. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part I*, volume 10534 of *Lecture Notes in Computer Science*, pages 770–785. Springer.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023a. Gemini: A Family of Highly Capable Multimodal Models. *CoRR*, abs/2312.11805.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. 2023b. PaLM 2 Technical Report. *CoRR*, abs/2305.10403.

Yuki Arase and Jun'ichi Tsujii. 2019. Transfer Fine-Tuning: A BERT Case Study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5392–5403. Association for Computational Linguistics.

SM Archana and Jay Prakash. 2024. Biomedical named entity recognition through improved balanced undersampling for addressing class imbalance and preserving contextual information. *International Journal of Information Technology*, pages 1–9.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen Technical Report. *CoRR*, abs/2309.16609.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Syst. Appl.*, 114:34–45.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3613–3618. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *CoRR*, abs/2004.05150.

Jie Yu Shasha Li Jun Ma Yuke Ji Huijun Liu Bin Ji, Hao Xu. 2023. A two-phase paradigm for joint entity-relation extraction. *Computers, Materials & Continua*, 74(1):1303–1318.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res.*, 32(Database-Issue):267–270.

Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, and Christopher D. Manning. 2024. BioMedLM: A 2.7B Parameter Language Model Trained On Biomedical Text. *CoRR*, abs/2403.18421.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving Language Models by Retrieving from Trillions of Tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Hamza Bousselham, El Habib Nfaoui, and Asmaa Mourhir. 2024. Fine-Tuning GPT on Biomedical NLP Tasks: An Empirical Evaluation. In *2024 International Conference on Computer, Electrical & Communication Engineering (ICCECE)*, pages 1–6.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. 2017. A systematic study of the class imbalance problem in convolutional neural networks. *CoRR*, abs/1710.05381.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. 2019. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. *Advances in neural information processing systems*, pages 1565–1576.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. *CoRR*, abs/2107.03374.

Qingyu Chen, Jingcheng Du, Yan Hu, Vipina Kuttichi Keloth, Xueqing Peng, Kalpana Raja, Rui Zhang, Zhiyong Lu, and Hua Xu. 2023. Large language models in biomedical natural language processing: benchmarks, baselines, and recommendations. *CoRR*, abs/2305.16326.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. PaLM: Scaling Language Modeling with Pathways. *Journal of Machine Learning Research*, 24:240:1–240:113.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4925–4936, Hong Kong, China. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. *CoRR*, abs/2210.11416.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *OpenReview.net*.

Cyril Cleverdon. 1997. *The Cranfield tests on index language devices*, page 47–59. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Donald C. Comeau, Rezarta Islamaj Dogan, Paolo Ciccarese, Kevin Bretonnel Cohen, Martin Krallinger, Florian Leitner, Zhiyong Lu, Yifan Peng, Fabio Rinaldi, Manabu Torii, Alfonso Valencia, Karin Verspoor, Thomas C. Wiegers, Cathy H. Wu, and W. John Wilbur. 2013. BioC: a minimalist approach to interoperability for biomedical text processing. *Database J. Biol. Databases Curation*, 2013.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Phil Crone. 2020. Deeper Task-Specificity Improves Joint Entity and Relation Extraction. *CoRR*, abs/2002.06424.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. 2019. Class-Balanced Loss Based on Effective Number of Samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9268–9277. Computer Vision Foundation / IEEE.

Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. 2020. Comparative Toxicogenomics Database (CTD): update 2021. *Nucleic Acids Research*. Gkaa891.

Allan Peter Davis, Thomas C. Wiegers, Michael C. Rosenstein, and Carolyn J. Mattingly. 2012. MEDIC: a practical disease vocabulary used at the Comparative Toxicogenomics Database. *Database J. Biol. Databases Curation*, 2012.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tobias Deußer, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa. 2023. Informed Named Entity Recognition Decoding for Generative Language Models. *CoRR*, abs/2308.07791.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

L. G. Divyanth, Afef Marzougui, Maria Jose González-Bernal, Rebecca J. McGee, Diego Rubiales, and Sindhuja Sankaran. 2022. Evaluation of Effective Class-Balancing Techniques for CNN-Based Assessment of Aphanomyces Root Rot Resistance in Pea (Pisum sativum L.). *Sensors*, 22(19):7237.

Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5308–5314, Florence, Italy. Association for Computational Linguistics.

Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: A resource for disease name recognition and concept normalization. *J. Biomed. Informatics*, 47:1–10.

Qi Dong, Shaogang Gong, and Xiatian Zhu. 2019. Imbalanced Deep Learning by Minority Class Incremental Rectification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(6):1367–1381.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Markus Eberts and Adrian Ulges. 2020. Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training. In *ECAI 2020 - 24th European Conference on Artificial Intelligence)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2006–2013. IOS Press.

Charles Elkan. 2001. The Foundations of Cost-Sensitive Learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001*, pages 973–978. Morgan Kaufmann.

Eyad Elyan, Carlos Francisco Moreno-García, and Chrisina Jayne. 2021. CDSMOTE: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification. *Neural Comput. Appl.*, 33(7):2839–2851.

M Eriksson and T Saldeen. 1989. Effects of an inhibitor of angiotensin converting enzyme (captopril) on pulmonary and renal insufficiency due to intravascular coagulation in the rat. *International journal of microcirculation, clinical and experimental*, 8(3):245—258.

Li Fang, Qingyu Chen, Chih-Hsuan Wei, Zhiyong Lu, and Kai Wang. 2023. Bioformer: an efficient transformer language model for biomedical text mining. *CoRR*, abs/2302.01588.

Hui Feng, Francesco Ronzano, Jude LaFleur, Matthew Garber, Rodrigo de Oliveira, Kathryn Rough, Katharine Roth, Jay Nanavati, Khaldoun Zine El Abidine, and Christina Mack. 2024. Evaluation of large language model performance on the biomedical language understanding and reasoning benchmark. *medRxiv*.

Nicolas Fiorini, Robert Leaman, David J Lipman, and Zhiyong Lu. 2018. How user intelligence is improving PubMed. *Nature biotechnology*, 36(10):937–945.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: Program-aided Language Models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.

Martin Gerner, Goran Nenadic, and Casey M. Bergman. 2010. LINNAEUS: A species name identification system for biomedical literature. *BMC Bioinform.*, 11:85.

Jiri Gesi and Iftekhar Ahmed. 2024. Beyond Self-learned Attention: Mitigating Attention Bias in Transformer-based Models Using Attention Guidance. *CoRR*, abs/2402.16790.

Tomas Goldsack, Zheheng Luo, Qianqian Xie, Carolina Scarton, Matthew Shardlow, Sophia Ananiadou, and Chenghua Lin. 2023. Overview of the biolaysumm 2023 shared task on lay summarization of biomedical research articles. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 468–477, Toronto, Canada. Association for Computational Linguistics.

Mitchell A. Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing BERT: Studying the Effects of Weight Pruning on Transfer Learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2020, Online, July 9, 2020*, pages 143–155. Association for Computational Linguistics.

Maria Carmela Groccia, Rosita Guido, Domenico Conforti, Corrado Pelaia, Giuseppe Armentaro, Alfredo Francesco Toscani, Sofia Miceli, Elena Succurro, Marta Letizia Hribal, and Angela Sciacqua. 2023. Cost-Sensitive Models to Predict Risk of Cardiovascular Events in Patients with Chronic Heart Failure. *Information*, 14(10).

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Trans. Comput. Heal.*, 3(1):2:1–2:23.

Yu Gu, Sheng Zhang, Naoto Usuyama, Yonas Woldesenbet, Cliff Wong, Praneeth Sanapathi, Mu Wei, Naveen Valluri, Erika Strandberg, Tristan Naumann, and Hoifung Poon. 2023. Distilling Large Language Models for Biomedical Knowledge Extraction: A Case Study on Adverse Drug Events. *CoRR*, abs/2307.06439.

Demi Guo, Alexander M. Rush, and Yoon Kim. 2021. Parameter-Efficient Transfer Learning with Diff Pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4884–4896. Association for Computational Linguistics.

Deepak Gupta and Dina Demner-Fushman. 2022. Overview of the MedVidQA 2022 shared task on medical video question-answering. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 264–274, Dublin, Ireland. Association for Computational Linguistics.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *J. Biomed. Informatics*, 45(5):885–892.

Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about GPT-3 In-Context Learning for Biomedical IE? Think Again. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4497–4512. Association for Computational Linguistics.

Maryam Habibi, Leon Weber, Mariana L. Neves, David Luis Wiegandt, and Ulf Leser. 2017. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinform.*, 33(14):i37–i48.

Ada Hamosh, Alan F. Scott, Joanna S. Amberger, Carol A. Bocchini, David Valle, and Victor A. McKusick. 2002. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res.*, 30(1):52–55.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. *CoRR*, abs/2403.14608.

Yaru Hao, Haoyu Song, Li Dong, Shaohan Huang, Zewen Chi, Wenhui Wang, Shuming Ma, and Furu Wei. 2022. Language Models are General-Purpose Interfaces. *CoRR*, abs/2206.06336.

Haibo He and Edwardo A. Garcia. 2009. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284.

Sophie Henning, William Beluch, Alexander Fraser, and Annemarie Friedrich. 2023. A survey of methods for addressing class imbalance in deep-learning based natural language processing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 523–540, Dubrovnik, Croatia. Association for Computational Linguistics.

Jerry R. Hobbs. 2002. Information extraction from biomedical text. *J. Biomed. Informatics*, 35(4):260–264.

Grant Van Horn and Pietro Perona. 2017. The Devil is in the Tails: Fine-grained Classification in the Wild. *CoRR*, abs/1709.01450.

Bofeng Huang. 2023. Vigogne: French Instruction-following and Chat Models. `https://github.com/bofenghuang/vigogne`.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning Deep Representation for Imbalanced Classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5375–5384. IEEE Computer Society.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2020. Deep Imbalanced Learning for Face Recognition and Attribute Prediction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(11):2781–2794.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rezarta Islamaj Doğan, Sun Kim, Andrew Chatr-aryamontri, Chih-Hsuan Wei, Donald C Comeau, Rui Antunes, Sérgio Matos, Qingyu Chen, Aparna Elangovan, Nagesh C Panyam, Karin Verspoor, Hongfang Liu, Yanshan Wang, Zhuang Liu, Berna Altınel, Zehra Melce Hüsünbeyi, Arzucan Özgür, Aris Fergadis, Chen-Kai Wang, Hong-Jie Dai, Tung Tran, Ramakanth Kavuluru, Ling Luo, Albert Steppi, Jinfeng Zhang, Jinchan Qu, and Zhiyong Lu. 2019. Overview of the BioCreative VI Precision Medicine Track: mining protein interactions and mutations for precision medicine. volume 2019, page bay147.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. 2022. OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization. *CoRR*, abs/2212.12017.

Bin Ji, Jie Yu, Shasha Li, Jun Ma, Qingbo Wu, Yusong Tan, and Huijun Liu. 2020. Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 88–99, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier,

Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *CoRR*, abs/2310.06825.

Jing Jiang. 2012. Information Extraction from Text. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 11–41. Springer.

Tianming Liu Jianquan Ouyang, Jing Zhang. 2022. Attention weight is indispensable in joint entity and relation extraction. *Intelligent Automation & Soft Computing*, 34(3):1707–1723.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Justin M. Johnson and Taghi M. Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *J. Big Data*, 6:27.

Mika Juuti, Tommi Gröndahl, Adrian Flanagan, and N. Asokan. 2020. A little goes a long way: Improving toxic language classification despite data scarcity. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2991–3009, Online. Association for Computational Linguistics.

Katikapalli Subramanyam Kalyan. 2024. A survey of GPT-3 family large language models including ChatGPT and GPT-4. *Nat. Lang. Process. J.*, 6:100048.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2022. AMMU: A survey of transformer-based biomedical pretrained language models. *J. Biomed. Informatics*, 126:103982.

Kamal raj Kanakarajan, Bhuvana Kundumani, and Malaikannan Sankarasubbu. 2021. Bio-ELECTRA:pretrained biomedical text encoder using discriminators. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 143–154, Online. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Salman H. Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous Ahmed Sohel, and Roberto Togneri. 2018. Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data. *IEEE Trans. Neural Networks Learn. Syst.*, 29(8):3573–3587.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, BioNLP@HLT-NAACL 2009 - Shared Task, Boulder, Colorado, USA, June 5, 2009*, pages 1–9. Association for Computational Linguistics.

Jin-Dong Kim and Sampo Pyysalo. 2013. *BioNLP Shared Task*, pages 138–141. Springer New York, New York, NY.

Shuhei Kinoshita, K. Bretonnel Cohen, Philip V. Ogren, and Lawrence Hunter. 2005. BioCreAtIvE Task1A: entity identification with a stochastic tagger. *BMC Bioinform.*, 6(S-1).

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics. Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).

Anis Koubaa. 2023. GPT-4 vs. GPT-3.5: A Concise Showdown. *Preprints*.

M.M.A. KOZEL, J.R. MEKKES, and J.D. BOS. 1995. Increased frequency and severity of angio-oedema related to long-term therapy with angiotensin-converting enzyme inhibitor in two patients. *Clinical and Experimental Dermatology*, 20(1):60–61.

Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. 2008. Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome biology*, 9:1–19.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M. Lowe, Roger A. Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, S. V. Ramanan, P. Senthil Nathan, Slavko Zitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A. Akhondi, Jan A. Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M. Dieb, Miji Choi, Karin Verspoor, Madian Khabsa, C. Lee Giles, Hongfang Liu, Ravikumar Komandur

Elayavilli, Andre Lamurias, Francisco M. Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usie, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez, Julen Oyarzabal, and Alfonso Valencia. 2015. The CHEMDNER corpus of chemicals and drugs and its annotation principles. *J. Cheminformatics*, 7(S-1):S2.

Ananya Kumar, Tengyu Ma, Percy Liang, and Aditi Raghunathan. 2022a. Calibrated ensembles can mitigate accuracy tradeoffs under distribution shift. In *Uncertainty in Artificial Intelligence, Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands*, volume 180 of *Proceedings of Machine Learning Research*, pages 1041–1051. PMLR.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022b. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Tuan Lai, Heng Ji, ChengXiang Zhai, and Quan Hung Tran. 2021. Joint biomedical entity and relation extraction with knowledge-enhanced collective inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6248–6260, Online. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *OpenReview.net*.

Hoang-Quynh Le, Duy-Cat Can, and Nigel Collier. 2022. Exploiting document graphs for inter sentence relation extraction. *J. Biomed. Semant.*, 13(1):15.

Robert Leaman, Rezarta Islamaj, Virginia Adams, Mohammed Alliheedi, João Rafael Almeida, Rui Antunes, Robert Bevan, Yung-Chun Chang, Arslan Erdengasileng, Matthew Hodgskiss, Ryuki Ida, Hyunjae Kim, Keqiao Li, Robert E. Mercer, Lukrécia Mertová, Ghadeer Mobasher, Hoo-Chang Shin, Mujeen Sung, Tomoki Tsujimura, Wen-Chao Yeh, and Zhiyong Lu. 2023. Chemical identification and indexing in full-text articles: an overview of the NLM-Chem track at BioCreative VII. *Database J. Biol. Databases Curation*, 2023.

Beth S Lee, David M Underhill, Monica K Crane, and Stephen L Gluck. 1995. Transcriptional Regulation of the Vacuolar H+-ATPase B2 Subunit Gene in Differentiating THP-1 Cells (⋆). *Journal of Biological Chemistry*, 270(13):7320–7329.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020a. Mixout: Effective Regularization to Finetune Large-scale Pretrained Language Models. *OpenReview.net*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020b. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4):1234–1240.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

David D. Lewis. 1995. Evaluating and Optimizing Autonomous Text Classification Systems. In *SIGIR'95, Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA, July 9-13, 1995 (Special Issue of the SIGIR Forum)*, pages 246–254. ACM Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fei Li, ZhiChao Lin, Meishan Zhang, and Donghong Ji. 2021. A span-based model for joint overlapped and discontinuous named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4814–4828, Online. Association for Computational Linguistics.

Jianning Li, Amin Dada, Behrus Puladi, Jens Kleesiek, and Jan Egger. 2024. Chat GPT in healthcare: A taxonomy and systematic review. *Computer Methods and Programs in Biomedicine*, 245:108013.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016:baw068.

Jinfen Li and Lu Xiao. 2020. syrapropa at SemEval-2020 task 11: BERT-based models design for propagandistic technique and span detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1808–1816, Barcelona (online). International Committee for Computational Linguistics.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019a. Entity-Relation Extraction as Multi-Turn Question Answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1340–1350. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019b. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society.

Carolyn E Lipscomb. 2000. Medical subject headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019a. GCDT: A global context enhanced deep transition architecture for sequence labeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Michelle Lo, Shay B. Cohen, and Fazl Barez. 2024. Large Language Models Relearn Removed Concepts. *CoRR*, abs/2401.01814.

Qiuhao Lu, Dejing Dou, and Thien Nguyen. 2022. ClinicalT5: A generative language model for clinical text. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5436–5443, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3036–3046. Association for Computational Linguistics.

Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N. Arighi, and Zhiyong Lu. 2022a. BioRED: a rich biomedical relation extraction dataset. *Briefings Bioinform.*, 23(5).

Ling Luo, Jinzhong Ning, Yingwen Zhao, Zhijun Wang, Zeyuan Ding, Peng Chen, Weiru Fu, Qinyu Han, Guangtao Xu, Yunzhi Qiu, Dinghao Pan, Jiru Li, Hao Li, Wenduo Feng, Senbo Tu, Yuqi Liu, Zhihao Yang, Jian Wang, Yuanyuan Sun, and Hongfei Lin. 2023a. Taiyi: A Bilingual Fine-Tuned Large Language Model for Diverse Biomedical Tasks. *CoRR*, abs/2311.11608.

Ling Luo, Chih-Hsuan Wei, Po-Ting Lai, Robert Leaman, Qingyu Chen, and Zhiyong Lu. 2023b. AIONER: all-in-one scheme-based biomedical named entity recognition using deep learning. *Bioinform.*, 39(5).

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022b. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings Bioinform.*, 23(6).

Youmi Ma, Tatsuya Hiraoka, and Naoaki Okazaki. 2020. Named Entity Recognition and Relation Extraction using Enhanced Table Filling by Contextualized Representations. *CoRR*, abs/2010.07522.

Youmi Ma, Tatsuya Hiraoka, and Naoaki Okazaki. 2022. Joint Entity and Relation Extraction Based on Table Labeling Using Convolutional Neural Networks. In *Proceedings of the Sixth Workshop on Structured Prediction for NLP, SPNLP@ACL 2022, Dublin, Ireland, May 27, 2022*, pages 11–21. Association for Computational Linguistics.

Dakota Mahan, Ryan Carlow, Louis Castricato, Nathan Cooper, and Christian Laforte. 2023. Stable Beluga models. *Hugging Face*.

David Masko and Paulina Hensman. 2015. The Impact of Imbalanced Training Data for Convolutional Neural Networks. *KTH, School of Computer Science and Communication (CSC)*.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. pages 3111–3119.

Shervin Minaee, Tomás Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large Language Models: A Survey. *CoRR*, abs/2402.06196.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.

Ghadeer Mobasher, Wolfgang Müller, Olga Krebs, and Michael Gertz. 2023. WeLT: Improving biomedical fine-tuned pre-trained language models with cost-sensitive learning. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 427–438, Toronto, Canada. Association for Computational Linguistics.

Masoud Monajatipoor, Jiaxin Yang, Joel Stremmel, Melika Emami, Fazlolah Mohaghegh, Mozhdeh Rouhsedaghat, and Kai-Wei Chang. 2024. LLMs in Biomedicine: A study on clinical Named Entity Recognition. *CoRR*, abs/2404.07376.

Milad Moradi, Kathrin Blagec, Florian Haberl, and Matthias Samwald. 2021. GPT-3 Models are Poor Few-Shot Learners in the Biomedical Domain. *CoRR*, abs/2109.02555.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. WebGPT: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332.

Hiroki Nakayama. 2018. seqeval: A Python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Keisuke Nemoto, Ryuhei Hamaguchi, Tomoyuki Imaizumi, and Shuhei Hikosaka. 2018. Classification of Rare Building Change Using CNN with Multi-Class Focal Loss. In *2018 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2018, Valencia, Spain, July 22-27, 2018*, pages 4663–4666. IEEE.

Sota Nemoto, Shunsuke Kitada, and Hitoshi Iyatomi. 2024. Majority or Minority: Data Imbalance Learning Method for Named Entity Recognition. *CoRR*, abs/2401.11431.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Katrin Ortmann. 2022. Fine-Grained Error Analysis and Fair Evaluation of Labeled Spans. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 1400–1407. European Language Resources Association.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Ibrahim Burak Ozyurt. 2020. On the effectiveness of small, discriminatively pre-trained language representation models for biomedical text mining. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 104–112, Online. Association for Computational Linguistics.

Arka Pal, Deep Karkhanis, Manley Roberts, Samuel Dooley, Arvind Sundararajan, and Siddartha Naidu. 2023. Giraffe: Adventures in Expanding Context Lengths in LLMs. *CoRR*, abs/2308.10882.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured Prediction as Translation between Augmented Natural Languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic Differentiation in PyTorch. *NIPS 2017 Workshop on Autodiff*.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2023. Gorilla: Large Language Model Connected with Massive APIs. *CoRR*, abs/2305.15334.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy. Association for Computational Linguistics.

Nadeesha Perera, Matthias Dehmer, and Frank Emmert-Streib. 2020. Named entity recognition and relation detection for biomedical information extraction. *Frontiers in cell and developmental biology*, 8:673.

Long N. Phan, James T. Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. SciFive: a text-to-text transformer model for biomedical literature. *CoRR*, abs/2106.03598.

Samira Pouyanfar, Yudong Tao, Anup Mohan, Haiman Tian, Ahmed S. Kaseb, Kent Gauen, Ryan Dailey, Sarah Aghajanzadeh, Yung-Hsiang Lu, Shu-Ching Chen, and Mei-Ling Shyu. 2018. Dynamic Sampling in Convolutional Neural Networks for Imbalanced Data Classification. In *IEEE 1st Conference on Multimedia Information Processing and Retrieval, MIPR 2018, Miami, FL, USA, April 10-12, 2018*, pages 112–117. IEEE.

Chen Qian, Guan Chunxiang, Xin Guo, Wang Suge, and Li Deyu. 2022. Inter-sentence Entity Relation Extraction based on GNN of Message Propagation.

Alec Radford. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Evani Radiya-Dixit and Xin Wang. 2020. How fine can fine-tuning be? Learning efficient language models. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*

2020, 26-28 August 2020, Online [Palermo, Sicily, Italy], volume 108 of *Proceedings of Machine Learning Research*, pages 2435–2443. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Feiliang Ren, Longhui Zhang, Shujuan Yin, Xiaofeng Zhao, Shilei Liu, Bochao Li, and Yaduo Liu. 2021. A novel global feature-oriented relational triple extraction model based on table filling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2646–2656, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Omid Rohanian, Mohammadmahdi Nouriborji, and David A. Clifton. 2024. Exploring the Effectiveness of Instruction Tuning in Biomedical Language Processing. *CoRR*, abs/2401.00579.

Omid Rohanian, Mohammadmahdi Nouriborji, Samaneh Kouchaki, and David A. Clifton. 2023. On the effectiveness of compact biomedical transformers. *Bioinform.*, 39(3).

Dan Roth and Wen-tau Yih. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning, CoNLL 2004, Held in cooperation with HLT-NAACL 2004, Boston, Massachusetts, USA, May 6-7, 2004*, pages 1–8. ACL.

Pedro Ruas and Francisco M. Couto. 2022. NILINKER: Attention-based approach to NIL Entity Linking. *Journal of Biomedical Informatics*, 132:104137.

Pedro Ruas, Andre Lamurias, and Francisco M. Couto. 2020. Linking chemical and disease entities to ontologies by integrating PageRank with extracted relations from literature. *J. Cheminformatics*, 12(1):57.

Naseer Ahmed Sajid, Atta Rahman, Munir Ahmad, Dhiaa Musleh, Mohammed Imran Basheer Ahmed, Reem Alassaf, Sghaier Chabani, Mohammed Salih Ahmed, Asiya Abdus Salam, and Dania AlKhulaifi. 2023. Single vs. multi-label: The issues, challenges and insights of contemporary classification schemes. *Applied Sciences*, 13(11).

Malik Sallam. 2023. ChatGPT Utility in Healthcare Education, Research, and Practice: Systematic Review on the Promising Perspectives and Valid Concerns. *Healthcare*, 11(6).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Tokala Yaswanth Sri Sai Santosh, Prantika Chakraborty, Sudakshina Dutta, Debarshi Kumar Sanyal, and Partha Pratim Das. 2021. Joint Entity and Relation Extraction from Scientific Documents: Role of Linguistic Information and Entity Types. *Proceedings of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021) co-located with JCDL 2021, Virtual Event, September 30th, 2021*, 3004:15–19.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *CoRR*, abs/2211.05100.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5248–5264, Online. Association for Computational Linguistics.

Hong Shen and Anoop Sarkar. 2005. Voting between multiple data representations for text chunking. In *Advances in Artificial Intelligence*, Lecture Notes in Computer Science, page 389–400, Berlin, Heidelberg. Springer.

Yongliang Shen, Xinyin Ma, Yechun Tang, and Weiming Lu. 2021. A trigger-sense memory flow framework for joint entity and relation extraction. In *Proceedings of the Web Conference 2021*, WWW '21, page 1704–1715, New York, NY, USA. Association for Computing Machinery.

Yiwen Shi, Taha ValizadehAslani, Jing Wang, Ping Ren, Yi Zhang, Meng Hu, Liang Zhao, and Hualou Liang. 2022. Improving Imbalanced Learning by Pre-finetuning with Data Augmentation. In *Fourth International Workshop on Learning with Imbalanced Domains: Theory and Applications, LIDTA 2022, Grenoble, France, September 23, 2022*, volume 183 of *Proceedings of Machine Learning Research*, pages 68–82. PMLR.

Jun Shu, Xiang Yuan, Deyu Meng, and Zongben Xu. 2023. CMW-Net: Learning a Class-Aware Sample Weighting Mapping for Robust Deep Learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(10):11521–11539.

Jatin Singh, Cameron Beeche, Zhiyi Shi, Oliver Beale, Boris Rosin, Joseph Leader, and Jiantao Pu. 2023. Batch-balanced focal loss: a hybrid solution to class imbalance in deep learning. *Journal of Medical Imaging*, 10(5):051809–051809.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Kumar Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Schärli, Aakanksha Chowdhery, Philip Andrew Mansfield, Blaise Agüera y Arcas, Dale R. Webster, Gregory S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle K. Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large Language Models Encode Clinical Knowledge. *CoRR*, abs/2212.13138.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Andrew Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Agüera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle K. Barral, Dale R. Webster, Gregory S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards Expert-Level Medical Question Answering with Large Language Models. *CoRR*, abs/2305.09617.

Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. 2008. Overview of BioCreative II gene mention recognition. *Genome biology*, 9:1–19.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zheng,

Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model. *CoRR*, abs/2201.11990.

Dandan Song, Jing Xu, Jinhui Pang, and Heyan Huang. 2021. Classifier-adaptation knowledge distillation framework for relation extraction and event detection with imbalanced data. *Information Sciences*, 573:222–238.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Nikos Sourlos, Jingxuan Wang, Yeshaswini Nagaraj, Peter van Ooijen, and Rozemarijn Vliegenthart. 2022. Possible Bias in Supervised Deep Learning Algorithms for CT Lung Nodule Detection and Classification. *Cancers*, 14(16).

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David D. Cox, Yiming Yang, and Chuang Gan. 2023. Principle-Driven Self-Alignment of Language Models from Scratch with Minimal Human Supervision. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Manan Suri. 2022. PiCkLe at SemEval-2022 Task 4: Boosting Pre-trained Language Models with Task Specific Metadata and Cost Sensitive Learning. In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 464–472. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on*

Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 3104–3112.

Ruixue Tang, Yanping Chen, Yongbin Qin, Ruizhang Huang, Bo Dong, and Qinghua Zheng. 2022. Boundary assembling method for joint entity and relation extraction. *Knowl. Based Syst.*, 250:109129.

Erdal Tasci, Ying Zhuge, Kevin Camphausen, and Andra V. Krauze. 2022. Bias and class imbalance in oncologic data—towards inclusive and transferrable ai in large scale oncology data sets. *Cancers*, 14(12).

Yi Tay, Jason Wei, Hyung Won Chung, Vinh Q. Tran, David R. So, Siamak Shakeri, Xavier Garcia, Huaixiu Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, Denny Zhou, Donald Metzler, Slav Petrov, Neil Houlsby, Quoc V. Le, and Mostafa Dehghani. 2023. Transcending Scaling Laws with 0.1% Extra Compute. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1471–1486. Association for Computational Linguistics.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A Large Language Model for Science. *CoRR*, abs/2211.09085.

Harish Tayyar Madabushi, Elena Kochkina, and Michael Castelle. 2019. Cost-sensitive BERT for generalisable sentence classification on imbalanced data. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 125–134, Hong Kong, China. Association for Computational Linguistics.

Shubo Tian, Qiao Jin, Lana Yeganova, Po-Ting Lai, Qingqing Zhu, Xiuying Chen, Yifan Yang, Qingyu Chen, Won Kim, Donald C. Comeau, Rezarta Islamaj, Aadit Kapoor, Xin Gao, and Zhiyong Lu. 2023. Opportunities and Challenges for ChatGPT and Large Language Models in Biomedicine and Health. *CoRR*, abs/2306.10070.

Pooja Tomar, Sonika Shrivastava, and Urjita Thakar. 2021. Ensemble Learning based Credit Card Fraud Detection System. In *2021 5th Conference on Information and Communication Technology (CICT)*, pages 1–5.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and Efficient Foundation Language Models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR*, abs/2307.09288.

Tung Tran and Ramakanth Kavuluru. 2019. Neural Metric Learning for Fast End-to-End Relation Extraction. *CoRR*, abs/1905.07458.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct Distillation of LM Alignment. *CoRR*, abs/2310.16944.

Szymon Tworkowski, Konrad Staniszewski, Mikolaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Milos. 2023. Focused Transformer: Contrastive Training for Context Scaling. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Taha ValizadehAslani, Yiwen Shi, Jing Wang, Ping Ren, Yi Zhang, Meng Hu, Liang Zhao, and Hualou Liang. 2022. Two-Stage Fine-Tuning: A Novel Strategy for Learning Class-Imbalanced Data. *CoRR*, abs/2207.10858.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, Relation, and Event Extraction with Contextualized Span Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International*

Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 5783–5788. Association for Computational Linguistics.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge Fusion of Large Language Models. *CoRR*, abs/2401.10491.

Qian Wan, Luona Wei, Xinhai Chen, and Jie Liu. 2021. A region-based hypergraph network for joint entity-relation extraction. *Knowl. Based Syst.*, 228:107298.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Xiaochen Wang and Yue Wang. 2022. Sentence-level resampling for named entity recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2151–2165, Seattle, United States. Association for Computational Linguistics.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yu Wang, Hanghang Tong, Ziye Zhu, and Yun Li. 2022. Nested Named Entity Recognition: A Survey. *ACM Trans. Knowl. Discov. Data*, 16(6):108:1–108:29.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Ming Wei, Zhipeng Xu, and Jiwei Hu. 2021. Entity relationship extraction based on bi-LSTM and attention mechanism. In *ICAIIS 2021: 2021 2nd International Conference on Artificial Intelligence and Information Systems, Chongqing, China, May 28 - 30, 2021*, pages 268:1–268:5. ACM.

Wei Wei, Jinjiu Li, Longbing Cao, Yuming Ou, and Jiahang Chen. 2013. Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, 16(4):449–475.

Thomas C. Wiegers, Allan Peter Davis, K. Bretonnel Cohen, Lynette Hirschman, and Carolyn J. Mattingly. 2009. Text mining and manual curation of chemical-gene-disease networks for the Comparative Toxicogenomics Database (CTD). *BMC Bioinform.*, 10:326.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. PMC-LLaMA: Further Finetuning LLaMA on Medical Papers. *CoRR*, abs/2304.14454.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ping Luo, and Ying Shan. 2024. LLaMA Pro: Progressive LLaMA with Block Expansion. *CoRR*, abs/2401.02415.

Ye Wu, Ruibang Luo, Henry C. M. Leung, Hing-Fung Ting, and Tak Wah Lam. 2019. RENET: A Deep Learning Approach for Extracting Gene-Disease Associations from Literature. In *Research in Computational Molecular Biology - 23rd Annual International Conference, RECOMB 2019,*

Washington, DC, USA, May 5-8, 2019, Proceedings, volume 11467 of *Lecture Notes in Computer Science*, pages 272–284. Springer.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. 2024. QA-LoRA: Quantization-Aware Low-Rank Adaptation of Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Chenguang Xue and Jiamin Lu. 2023. Dealing with negative samples with multi-task learning on span-based joint entity-relation extraction. *CoRR*, abs/2309.09713.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 483–498. Association for Computational Linguistics.

Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. 2021. A Partition Filter Network for Joint Entity and Relation Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 185–197. Association for Computational Linguistics.

Jufeng Yang, Xiaoping Wu, Jie Liang, Xiaoxiao Sun, Ming-Ming Cheng, Paul L. Rosin, and Liang Wang. 2020a. Self-Paced Balance Learning for Clinical Skin Disease Recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 31(8):2832–2846.

Kaixiang Yang, Zhiwen Yu, C. L. Philip Chen, Wenming Cao, Hau-San Wong, Jane You, and Guoqiang Han. 2022. Progressive Hybrid Classifier Ensemble for Imbalanced Data. *IEEE Trans. Syst. Man Cybern. Syst.*, 52(4):2464–2478.

Wenshuo Yang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2020b. HSCNN: A Hybrid-Siamese Convolutional Neural Network for Extremely Imbalanced Multi-label Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6716–6722. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917, Dublin, Ireland. Association for Computational Linguistics.

Wei Ye, Bo Li, Rui Xie, Zhonghao Sheng, Long Chen, and Shikun Zhang. 2019. Exploiting entity BIO tag embeddings and multi-task learning for relation extraction with imbalanced data. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1351–1360, Florence, Italy. Association for Computational Linguistics.

Alexander S. Yeh, Alexander A. Morgan, Marc E. Colosimo, and Lynette Hirschman. 2005. BioCreAtIvE Task 1A: gene mention finding evaluation. *BMC Bioinform.*, 6(S-1).

Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Tingwen Liu, Yubin Wang, Bin Wang, and Sujian Li. 2020. Joint Extraction of Entities and Relations Based on a Novel Decomposition Strategy. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2282–2289. IOS Press.

Jie Yu, Bin Ji, Shasha Li, Jun Ma, Huijun Liu, and Hao Xu. 2022. S-NER: A Concise and Efficient Span-Based Model for Named Entity Recognition. *Sensors*, 22(8):2852.

Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. 2022. BioBART: Pretraining and Evaluation of A Biomedical Generative Language Model. In *Proceedings of the 21st Workshop on Biomedical Language Processing, BioNLP@ACL 2022, Dublin, Ireland, May 26, 2022*, pages 97–109. Association for Computational Linguistics.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2023. HyPe: Better Pre-trained Language Model Fine-tuning with Hidden Representation Perturbation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3246–3264. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating Generated Text as Text Generation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27263–27277.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9. Association for Computational Linguistics.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. In *The Thirty-Fourth AAAI*

Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 9507–9514. AAAI Press.

Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. 2021a. Dive into Deep Learning. *CoRR*, abs/2106.11342.

Haiyang Zhang, Guanqun Zhang, and Yue Ma. 2021b. Syntax-informed self-attention network for span-based joint entity and relation extraction. *Applied Sciences*, 11(4).

Jiaxin Zhang, Jie Liu, Shaowei Chen, Shaoxin Lin, Bingquan Wang, and Shanpeng Wang. 2022. ADAM: An Attentional Data Augmentation Method for Extreme Multi-label Text Classification. In *Advances in Knowledge Discovery and Data Mining - 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16-19, 2022, Proceedings, Part I*, volume 13280 of *Lecture Notes in Computer Science*, pages 131–142. Springer.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1730–1740, Copenhagen, Denmark. Association for Computational Linguistics.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. TinyLlama: An Open-Source Small Language Model. *CoRR*, abs/2401.02385.

Sheng Zhang, Hao Cheng, Jianfeng Gao, and Hoifung Poon. 2023a. Optimizing Bi-Encoder for Named Entity Recognition via Contrastive Learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023b. Instruction Tuning for Large Language Models: A Survey. *CoRR*, abs/2308.10792.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sanqiang Zhao, Raghav Gupta, Yang Song, and Denny Zhou. 2019. Extreme Language Model Compression with Optimal Subwords and Shared Projections. *CoRR*, abs/1909.11687.

Shan Zhao, Minghao Hu, Zhiping Cai, and Fang Liu. 2020. Modeling Dense Cross-Modal Interactions for Joint Entity-Relation Extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4032–4038. ijcai.org.

Zexuan Zhong and Danqi Chen. 2021. A Frustratingly Easy Approach for Entity and Relation Extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 50–61. Association for Computational Linguistics.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-Level Relation Extraction with Adaptive Thresholding and Localized Context Pooling. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14612–14620. AAAI Press.

Yichu Zhou and Vivek Srikumar. 2022. A Closer Look at How Fine-tuning Changes BERT. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1046–1061. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. 2020. Modifying Memories in Transformer Models. *CoRR*, abs/2012.00363.

Yaqin Zhu, Xuhang Li, Zijian Wang, Jiayong Li, Cairong Yan, and Yanting Zhang. ER-LAC: Span-Based Joint Entity and Relation Extraction Model with Multi-Level Lexical and Attention on Context Features. *Applied Sciences*, 13(18):10538.

Appendix

Hyper-parameters

We report all the hyperparameters for BioNER experiments in Section 3.4.4. Here is the brief description of the hyperparameters as follows:

- maximum sequence length denoted by *max_seq_length*, which represents the maximum length of texts the BERT model can process,
- train batch size designated by *train_batch_size*, referring to the number of training instances in each batch, and
- training epochs denoted by *num_train_epochs*, identifying the number of epochs comprising one complete pass through the entire training dataset with predefined iterations.

Model	max_seq_length	train_batch_size	num_train_epochs
BioBERT	384	5	20
BlueBERT	128	32	30
PubMedBERT	512	5	30
SciBERT	384	5	20
BioELECTRA	512	5	100

Table 1: Hyper-parameters for fine-tuning BioRED-Chemical

Model	max_seq_length	train_batch_size	num_train_epochs
BioBERT	384	5	20
BlueBERT	128	32	30
PubMedBERT	320	8	10
SciBERT	512	5	30
BioELECTRA	256	16	13

Table 2: Hyper-parameters for fine-tuning BioRED-Disease

Model	max_seq_length	train_batch_size	num_train_epochs
BioBERT	320	8	10
BlueBERT	256	16	13
PubMedBERT	512	5	75
SciBERT	320	8	30
BioELECTRA	384	12	20

Table 3: Hyper-parameters for fine-tuning BC5CDR-Chemical

Model	max_seq_length	train_batch_size	num_train_epochs
BioBERT	512	5	30
BlueBERT	256	16	13
PubMedBERT	512	5	30
SciBERT	256	16	13
BioELECTRA	256	16	13

Table 4: Hyper-parameters for fine-tuning BC5CDR-Disease

Model	max_seq_length	train_batch_size	num_train_epochs
BioBERT	128	32	30
BlueBERT	320	8	80
PubMedBERT	384	5	20
SciBERT	320	8	30
BioELECTRA	128	32	30

Table 5: Hyper-parameters for fine-tuning BC4Chem

Model	max_seq_length	train_batch_size	num_train_epochs
BioBERT	320	8	10
BlueBERT	256	16	20
PubMedBERT	320	8	20
SciBERT	256	16	13
BioELECTRA	384	12	10

Table 6: Hyper-parameters for fine-tuning Linnaeus

Model	max_seq_length	train_batch_size	num_train_epochs
BioBERT	128	32	30
BlueBERT	256	16	40
PubMedBERT	256	16	30
SciBERT	128	32	30
BioELECTRA	384	12	10

Table 7: Hyper-parameters for fine-tuning BC2GM-Gene

Model	max_seq_length	train_batch_size	num_train_epochs
BioBERT	384	5	20
BlueBERT	256	16	13
PubMedBERT	384	5	20
SciBERT	256	16	13
BioELECTRA	256	16	5

Table 8: Hyper-parameters for fine-tuning NCBI

We report all the hyperparameters for BioNER and BioNEL experiments in Section 3.6.

Model	max_seq_length	train_batch_size	num_train_epochs
NCBI	320	8	10
BC5CDR-disease	320	8	10
BC5CDR-chemical	256	16	10
BioRED-disease	384	8	10
BioRED-chemical	128	32	10

Table 9: Hyper-parameters for fine-tuning BioNER and BioNEL

We report all the hyperparameters for SpERT-WeLT experiments in Section 4.4.

Hyper-parameters	**CoNLL04 & ADE(train_development)**
Entity negative sampling	100
Relation negative sampling	100
Pre-trained model type	$BERT_{BASE}$(cased)
Width embedding size	25
Batch size	2
Relation classifier threshold	0.4
Optimizer	Adam
Peak learning rate	5e-5
Linear warmup learning rate	0.1
Linear decay learning rate	0.01
Epochs	20
Batch size	2

Table 10: Hyper-parameters for SpERT-WeLT experiments

We report all the hyperparameters for ASpERT-WeLT experiments in Section 5.3. We have used the same hyperparameters as ASpERT experiments with an exception of batch size, due to limited computational resources.

Hyper-parameters	**CoNLL04 & ADE(training)**
Entity negative sampling	150
Relation negative sampling	150
Span classifier MLP size	784
Contribution threshold	0.5
MLP dropout	0.1
Pre-trained model type	$BERT_{BASE}$(cased) & *BioBERT*(cased) for ADE
Width embedding size	25
Batch size	2
Relation classifier threshold	0.4
Optimizer	Adam
Peak learning rate	5e-5
Linear warmup learning rate	0.1
Linear decay learning rate	0.01
Epochs	20
Batch size	2

Table 11: Hyper-parameters for ASpERT-WeLT experiments

We report all the hyperparameters for TablERT-CNN-WeLT experiments in Section 6.4.

Hyper-parameters	**CoNLL04(train_development)**	**ADE(train_development)**
CNN configuration		
Kernel size ($F_h \times F_w$)	3×3	3×3
Number of Layers L	2	3
Hidden Dimension $d^{(l)}$	512	256|512
Training configuration		
Batch size	8	16
Pre-trained model type	$BERT_{BASE}$(cased)	$BERT_{BASE}$(cased)
BERT learning rate	5e-5	5e-5
Learning rate for other parameters	1e-3	1e-3
Dropout	0.3	0.3
Epochs	30	30
Warm-up period	0.2	0.2

Table 12: Hyper-parameters for TablERT-CNN-WeLT experiments