

# **Knowledge-Enhanced Neural Networks for Machine Reading Comprehension**



**Todor Borisov Mihaylov**

Department of Computational Linguistics  
Heidelberg University

This dissertation is submitted for the degree of  
*Doctor of Philosophy*

Supervisor: Prof. Dr. Anette Frank

External Reviewer: Prof. Dr. Sebastian Riedel

Submission: 30 August 2021

Defense date: 20 February 2022

## Acknowledgements

I would like to express my deepest gratitude to my parents, who have been my pillars of strength and unwavering support throughout my life. Their love, encouragement, and belief in my abilities have fueled my desire to pursue a Ph.D. I am grateful for their sacrifices and the countless ways in which they have contributed to my growth, both personally and academically.

I am thankful to all my research advisors and collaborators. First, I would like to thank my supervisor Prof. Dr. Anette Frank, for allowing me to pursue my doctoral studies and giving me the freedom to choose the direction of my research. I would also like to thank Anette for her patience and support and for guiding me through this journey. Special thanks to Preslav Nakov, who hooked me up with research and inspired me to pursue a Ph.D. degree. I want to thank Zornitsa Kozareva for giving me the opportunity to intern with her at Amazon and for teaching me so much about the corporate research world and Ashish Sabharwal, Tushar Khot, and Peter Clark for our productive time at AI2.

I would like to thank Tobias, Markus, Avinesh, Benjamin, Federico, Teresa, Daniil, and all the members of the AIPHES project for learning so much from them and for the enjoyable lunches and retreats. I am grateful to the AIPHES Doctoral School, TU-Darmstadt, and Heidelberg University for having amazing invited speakers.

I am grateful for having my Heidelberg lab mates Ana, Angel, Debjit, Ngoc, Maria, Juri, Eva, Esther, Bhushan, Julia, Sariya, Hiko, and Julius, who made the stressful Ph.D. process an enjoyable and fun day-to-day business. Special thanks to Ana for the fruitful discussions and for constantly reminding me of all administrative deadlines and tasks, and Eva, who was always patient to help with the navigation through the complex German administrative maze.

I am grateful for meeting many amazing people in Heidelberg, and the fun wine fests, hiking trips, climbing evenings, poker nights, and Neckarwiese beers with Angel, Agne, Debjit, Jeff, Fede, Vera, Ngoc, Ana, Esther, and Bhushan among others.

Last but not least, I want to thank my wife Quynh, for her patience and support when I was completing this manuscript and my sister Tsvetomila for help with proofreading!

## Abstract

Machine Reading Comprehension is a language understanding task where a system is expected to read a given passage of text and typically answer questions about it. When humans assess the task of reading comprehension, in addition to the presented text, they usually use the knowledge that they already know, such as commonsense and world knowledge, or language skills that they previously acquired - understanding the events and arguments in a text (who did what to whom), their participants and the relation in discourse. In contrast, neural network approaches for machine reading comprehension focused on training end-to-end systems that rely only on annotated task-specific data.

In this thesis, we explore approaches for tackling the reading comprehension problem, motivated by how a human would solve the task, using existing background and commonsense knowledge or knowledge from various linguistic tasks.

First, we develop a neural reading comprehension model that integrates external commonsense knowledge encoded as a key-value memory. Instead of relying only on document-to-question interaction or discrete features, our model attends to relevant external knowledge and combines this knowledge with the context representation before inferring the answer. This allows the model to attract and imply knowledge from an external knowledge source that is not explicitly stated in the text but is relevant for inferring the answer. We demonstrated that the proposed approach improves the performance of very strong base models for cloze-style reading comprehension and open-book question answering. By including knowledge explicitly, our model can also provide evidence about the background knowledge used in the reasoning process.

Further, we examined the impact of transferring linguistic knowledge from low-level linguistic tasks into a reading comprehension system using neural representations. Our experiments show that the knowledge transferred from the neural representations trained on these linguistic tasks can be adapted and combined together to improve the reading comprehension task early in training and when trained with small portions of the data.

Last, we propose to use structured linguistic annotations as a basis for a Discourse-Aware Semantic Self-Attention encoder that we employ for reading comprehension of narrative texts. We extract relations between discourse units, events, and their arguments, as well

as co-referring mentions, using available annotation tools. The empirical evaluation shows that the investigated structures improve the overall performance (up to +3.4 Rouge-L), especially intra-sentential and cross-sentential discourse relations, sentence-internal semantic role relations, and long-distance coreference relations. We also show that dedicating self-attention heads to intra-sentential relations and relations connecting neighboring sentences is beneficial for finding answers to questions in longer contexts. These findings encourage the use of discourse-semantic annotations to enhance the generalization capacity of self-attention models for machine reading comprehension.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Research Objective . . . . .	3
1.3	Contributions . . . . .	4
1.4	Thesis Outline . . . . .	6
1.5	Publications . . . . .	7
<b>2</b>	<b>Background</b>	<b>9</b>
2.1	Knowledge Encoding in Neural Networks . . . . .	10
2.2	Machine Reading Comprehension . . . . .	12
2.2.1	Task Formulations . . . . .	12
2.2.2	Related Natural Language Processing Tasks . . . . .	20
2.3	Neural Network Approaches for Machine Reading Comprehension . . . . .	22
2.3.1	Common Architecture . . . . .	22
2.3.2	Neural Reader . . . . .	23
2.3.3	Answer Re-Ranking and Ensemble . . . . .	30
2.4	Neural Machine Reading Comprehension with Prior Knowledge . . . . .	30
2.4.1	Token Embeddings . . . . .	31
2.4.2	Features . . . . .	31
2.4.3	Combining Text and External Knowledge Sources . . . . .	31
2.4.4	Neural Transfer Learning for Machine Reading Comprehension . . . . .	33
2.5	Summary . . . . .	36
<b>3</b>	<b>Neural Machine Reading Comprehension using External Declarative Knowledge</b>	<b>37</b>
3.1	Motivation . . . . .	37
3.2	Data and Task Descriptions . . . . .	38
3.2.1	Cloze-style Reading Comprehension with External Commonsense Knowledge . . . . .	38

3.2.2	Open Book Question Answering with External Background and Commonsense Knowledge . . . . .	39
3.3	Cloze-style Reading Comprehension with Background Knowledge Sources	41
3.3.1	Knowledge Retrieval . . . . .	41
3.3.2	Knowledgeable Reader: Neural Reader with Explicit Knowledge Memory . . . . .	42
3.3.3	Technical Details . . . . .	47
3.4	Open Book Question Answering with External Knowledge Sources . . . . .	48
3.4.1	Knowledge Retrieval . . . . .	49
3.4.2	Knowledgeable Reader for Multi-Choice Question Answering . . . . .	50
3.4.3	Baseline Models . . . . .	51
3.4.3.1	No Training, External Knowledge Only . . . . .	51
3.4.3.2	No Training; $\mathcal{F}$ and External Knowledge . . . . .	52
3.4.3.3	Trained Models, No Knowledge . . . . .	52
3.4.4	Technical Details . . . . .	53
3.5	Experiments and Results . . . . .	54
3.5.1	Cloze-style Reading Comprehension . . . . .	54
3.5.1.1	Model Parameters . . . . .	54
3.5.1.2	Empirical Results . . . . .	55
3.5.2	Open Book Question Answering . . . . .	58
3.6	Discussion and Analysis . . . . .	60
3.6.1	Analysis of the empirical results. . . . .	60
3.6.2	Interpreting Component Importance . . . . .	60
3.6.3	Qualitative Data Investigation . . . . .	61
3.6.4	Success and Failure Examples for Open Book QA . . . . .	70
3.7	Summary and Conclusions . . . . .	72
<b>4</b>	<b>Neural Machine Reading Comprehension using Contextual Representations</b>	
	<b>Pre-trained on Lower-Level Supervised Language Tasks</b>	<b>73</b>
4.1	Introduction . . . . .	73
4.2	Method . . . . .	74
4.2.1	Skill Tasks . . . . .	75
4.2.2	Skill Learning Architectures . . . . .	77
4.2.3	Skillful Reader: Reading Comprehension with Skill Representations	81
4.3	Learning Skill Encoders from Tasks . . . . .	83
4.4	Neural Transfer to Machine Reading Comprehension . . . . .	86
4.4.1	Training Details . . . . .	87

4.4.2	Experiments and Results . . . . .	87
4.4.2.1	Overall Results . . . . .	88
4.4.2.2	Limited Data and Training Stages . . . . .	88
4.4.2.3	Skill Learning Architecture and Modifications . . . . .	91
4.5	Discussion . . . . .	97
4.6	Summary and Conclusions . . . . .	100
<b>5</b>	<b>Neural Machine Reading Comprehension with Structured Linguistic Knowledge</b>	<b>101</b>
5.1	Introduction . . . . .	101
5.2	Discourse-aware Semantic Annotations . . . . .	103
5.3	Discourse-Aware Semantic Self-Attention Model . . . . .	105
5.3.1	Base Model . . . . .	105
5.3.2	Discourse-Aware Semantic Self-Attention . . . . .	106
5.4	Data and Task Description . . . . .	107
5.5	Related Work . . . . .	108
5.6	Experiments and Results . . . . .	110
5.6.1	Overall Results . . . . .	110
5.6.2	Fine-grained Evaluation . . . . .	111
5.7	Conclusion and Future Work . . . . .	115
<b>6</b>	<b>Summary and Conclusions</b>	<b>116</b>
6.1	Summary of the Contributions . . . . .	116
6.2	Current Trends and Future Directions . . . . .	117
	<b>List of Figures</b>	<b>121</b>
	<b>List of Tables</b>	<b>125</b>
	<b>References</b>	
<b>A</b>	<b>Source Code</b>	<b>149</b>
<b>B</b>	<b>Discourse Relation Sense Classification</b>	<b>150</b>
B.1	A System for Discourse Relation Sense Classification . . . . .	150
B.1.1	Discourse Relation Sense Classification Data . . . . .	150
B.1.2	Related Work . . . . .	151
B.1.3	Method . . . . .	152
B.1.3.1	Feature-based approach . . . . .	153



---

B.1.3.2	CNNs for sentence classification . . . . .	154
B.1.3.3	Modified ARC-1 CNN for sentence matching . . . . .	155
B.1.4	Experiments and Results . . . . .	155
B.1.4.1	Data . . . . .	155
B.1.4.2	Classifier settings . . . . .	155
B.1.4.3	Official submission (LR with E+Sim) . . . . .	156
B.1.4.4	Further experiments on Non-Explicit relations . . . . .	157
B.1.5	Summary . . . . .	159

# Chapter 1

## Introduction

### 1.1 Motivation

Machine Reading Comprehension (MRC) is a language understanding task, typically evaluated in a question answering setting, where a system reads a text passage (document  $D$ ) and answers questions ( $Q$ ) about it. It is inspired by the standard reading comprehension exams used in schools to measure the ability of students, to read, comprehend, and reason about a given text. The MRC task was originally adapted by (Hirschman et al., 1999) who has proposed using the task to evaluate an automatic language comprehension model. This was later extended by (Breck et al., 2001) and (Richardson et al., 2013) but these were not enough to get the computational linguistics community attention to the MRC task. Recently, work on novel datasets for machine reading comprehension gained a lot of attention when Hermann et al. (2015) automatically created a large scale cloze-style reading comprehension inspired by the cloze test (Taylor, 1953; Bormuth, 1967). Soon several other large-scale automatically created (Weston et al., 2015a; Hill et al., 2016; Onishi et al., 2016) and crowdsourced (Rajpurkar et al., 2016; Trischler et al., 2017; Joshi et al., 2017; Kociský et al., 2017) datasets has appeared and allowed training deep neural networks to perform the task.

Analysis of the data (Sugawara et al., 2017) and where models fail (Chen et al., 2016a; Rajpurkar et al., 2016) show that the Machine Reading Comprehension task in some of these datasets requires a set of linguistic and cognitive skills such as paraphrase detection, recognition of named entities, natural language inference, understanding of the discourse, reasoning, and knowledge such as background and common sense, etc. According to the research in cognitive science, in order to perform well on reading comprehension tests, in addition to being thought to *read and practice answering questions* (Tierney and Cunningham, 1980; Collins and Smith, 1980), humans would greatly benefit from, *prior commonsense and background knowledge* (Pearson et al., 1979; Cai, 2002; Hirsch, 2003; Willingham, 2006),

and *linguistic awareness* (Bialystok, 1988). In this thesis, we hypothesize that solving the task of Machine Reading Comprehension would benefit from some automatic annotations and information that resembles these skills and knowledge.

Initial approaches on MRC has proposed simple neural network models that encode the question and context with a neural networks in a single read (one-hop) (Hermann et al., 2015; Kadlec et al., 2016; Chen et al., 2016a). Later work includes more complex neural models (Weston et al., 2015c; Dhingra et al., 2017b; Cui et al., 2017; Munkhdalai and Yu, 2016; Sordoni et al., 2016) that are focused on *reading and practicing* by performing *reading* of the story and a question on multiple times before inferring the correct answer. While these approaches perform well when trained from scratch (i) they have complex architecture, require large training data, and (iii) are particularly a black-box. These data-hungry approaches can be aligned with the *reading and practice* that students usually do, to perform well on real reading comprehension tests but lack the *background, commonsense, and linguistic knowledge* that they have.

Given the skill-focused analysis of MRC datasets (Chen et al., 2016a; Sugawara et al., 2017) and findings in the cognitive science, mentioned above, we explore different methods to solve the task of Machine Reading Comprehension in a similar way that humans do: by combining prior commonsense and background knowledge and existing linguistic skills in the process of solving the reading comprehension task.

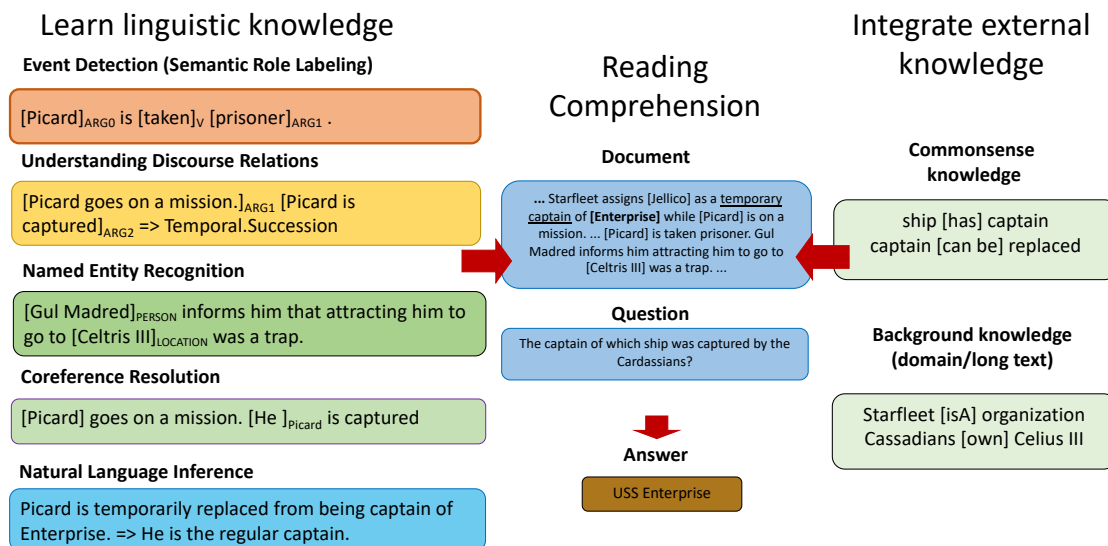


Fig. 1.1 Reading Comprehension requires using external commonsense knowledge and combination of linguistic tasks and background knowledge.

Figure 1.1 shows different types of external background and linguistic knowledge from existing knowledge-bases and natural language processing tasks that can be combined with the input data of the more complex task of Machine Reading Comprehension.<sup>1</sup>

We hypothesize that the knowledge from knowledge-bases such as ConceptNet (Speer et al., 2017) (commonsense: ex. *captian [can be] replaced*) or Wikidata (Vrandečić and Krötzsch, 2014) (world knowledge: ex. *Starfleet [isA] space organization (fictional)*) could be integrated in a neural network model and improve the performance of answering questions about a text document. Moreover, existing natural language processing tasks represent semantic and discourse knowledge that can be beneficial for reading comprehension: how to recognize events and their participants (Semantic Role Labeling), make inferences about statements (Natural Language Inference / Textual Entailment), recognize discourse relations between events (Discourse Parsing), and detecting mentions of story participants (Coreference Resolution).

We hypothesize that prior external background knowledge and linguistic knowledge could help the neural network model by (i) improving the overall performance, (ii) learning the target machine reading comprehension task with less training data, and (iii) making the model decisions easier to analyze.

## 1.2 Research Objective

In this work, we propose new methods for machine reading comprehension and question answering that model the human approach of using previously acquired to reason about the content of a natural language text.

Based on our motivations above, we formulate the following research questions:

- **Question 1: Can existing commonsense knowledge from knowledge-base such as ConceptNet and WordNet be incorporated in a neural network model to improve Machine Reading Comprehension?**
- **Question 2: Can neural network representations learned from linguistic knowledge from existing natural language processing tasks such as discourse parsing, event and argument detection, and natural language inference be transferred to a higher-level task of Machine Reading Comprehension?**

---

<sup>1</sup>Not all of the presented knowledge would be relevant to the question and given context so the model needs to be able to utilize only the useful one.

- **Question 3: Can knowledge from linguistic structured annotations representing linguistic knowledge about discourse relations, events, and coreference resolution, improve neural network models for Machine Reading Comprehension?**

## 1.3 Contributions

The recent success in Machine Reading Comprehension and most of the research in neural networks is mostly due to the large-scale datasets and computational power that allowed scaling of deep neural networks using gradient-based methods (See Chapter 2). At first, most of the MRC neural approaches used to rely on Recurrent Neural Networks, such as Long Short-Term Memory Networks (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRU) (Cho et al., 2014) due to their ability to build a rich contextualized representation of document and question which gives a very well performance when used in very simple systems (Hermann et al., 2015; Kadlec et al., 2016; Chen et al., 2016a). In this thesis, we built upon MRC methods leveraging Recurrent Neural Networks and attention-based transformers (Vaswani et al., 2017). We proposed novel approaches and models for machine reading comprehension that leverage external background knowledge from knowledge bases and linguistic knowledge into recurrent neural networks to seek answers to the first two questions raised in Section 1.2:

First, we tackle the question *Q1: Can existing commonsense knowledge from knowledge-base such as ConceptNet and WordNet be incorporated in a neural network model to improve Machine Reading Comprehension?* and describe our findings in our first contribution:

### **Neural Machine Reading Comprehension with External Commonsense Knowledge.**

We propose extending a standard machine reading comprehension task with explicit external commonsense knowledge and show that this is beneficial for the overall performance and intractability. We develop a method for integrating knowledge in a simple but effective reading comprehension neural model (*AS Reader*, Kadlec et al. (2016)) and improve its results, whereas other models employ features or multiple hops. We examine two sources of commonsense knowledge: WordNet (Miller et al., 1990) and ConceptNet (Speer et al., 2017) and show that this type of knowledge is important for answering *common nouns* questions and *named entities* questions from the Children Book Test (CBTest) dataset (Hill et al., 2016). Our proposed approach is explainable. Unlike concurrent work that encodes the knowledge implicitly in the word embeddings layer (Weissenborn et al., 2017a), our approach of injecting external knowledge is explicit and gives evidence about the used portion of information (facts) for reasoning about the correct answers to a question. We demonstrate

the effectiveness of the injected knowledge by case studies and data statistics in a qualitative evaluation study for Reading Comprehension and Question Answering. We further examine the effectiveness of our approach by adapting it for multi-hop question answering over partial context (Mihaylov et al., 2018) and demonstrate that when the proper background knowledge is available, the model performs with high accuracy on the task and makes the model easy to interpret.

Next, we answer *Q2: Can neural network representations learned from linguistic knowledge from existing natural language processing tasks such as discourse parsing, event and argument detection, and natural language inference be transferred to a higher-level task of Machine Reading Comprehension?:*

**Machine Reading Comprehension using Neural Contextual Representations Trained on Supervised Language Tasks.** In this work, we propose to learn neural representations from multiple supervised lower-level linguistic tasks (referred to as ‘skills’) and transfer them to Machine Reading Comprehension. We develop a simple RC model that allows us to combine the learned ‘skill’ representations easily and analyze the learning behavior of this skillful neural model. We show that using such skills, learned from specialized natural language processing tasks, boosts the performance of a neural reading comprehension model (i) early in training and (ii) when training on smaller portions (2, 5, 10, or 25 percent) of the original training data. We further show which skills are important for the task by performing ablations of neural representations integrated into the target reading comprehension model.

While the recurrent neural networks had great success due to their ease of use and good performance for many tasks in natural language processing, they have issues with scalability due to their recurrent nature. Recently (Vaswani et al., 2017) introduced the *Transformer* - a new class of neural network models that use self-attention and positional encoding and transformations based on feed-forward neural network layers that perform well for many tasks such as Machine Translation and Question Answering. They do not have recurrent connections in their nature and therefore are very scalable. However, these models have other weak spots like the inability to generalize to long sequences (Dai et al., 2019) and large memory consumption due to their  $O(n^2)$  complexity regarding the number of tokens in the text. In our third contribution, we answer our third research question: "*Q3: Can we use lower-level linguistic structured annotations such as discourse relations, events, and coreference, to improve state-of-the-art Machine Reading Comprehension systems?.* Primarily we focus on improving the state-of-the-art model for reading comprehension (QANet (Yu et al., 2018)) that uses a Transformer-based self-attention architecture and has poor performance when used on longer narrative texts:

**Neural Machine Reading Comprehension with Structured Linguistic Knowledge.** We propose a *Discourse-Aware Semantic Self-Attention* mechanism, an extension to the standard self-attention models – without significant increase of computation complexity. We show that discourse and semantic annotations help the self-attention model to improve its performance when evaluated on long texts. We analyze the impact of different discourse and semantic annotations on narrative reading comprehension. We perform an empirical fine-grained evaluation of the discourse-semantic annotations on specific question types and context size regions and show interesting dependence between discourse and semantic types and question types (ex. Semantic Role Labeling (events) improves *who* and *when* questions, intra-sentential *Explicit discourse relations* improve *why* and *where* questions). We also show that all relations improve the performance of answering questions on longer texts. To annotate the raw text, we use existing tools for Semantic Role Labeling (Gardner et al., 2017) and Coreference Resolution. To annotate our documents with discourse relations, we developed a fast and simple method for discourse relation sense disambiguation that achieves state-of-the-art results and has won first place in the overall evaluation of CoNLL 2016 Shared Task on Discourse Relation Sense Classification (Xue et al., 2016a).

## 1.4 Thesis Outline

In Chapter 2 we present the background needed for understanding the content of the work, presented in this thesis. We also review related work on machine reading comprehension including multiple formulations of the task, datasets, standard architecture, and state-of-the-art approaches. We also place our own contributions and show how they fit in the field.

In the next three chapters, we describe proposed methods and approaches for leveraging linguistic and background external knowledge for machine reading comprehension.

In Chapter 3 we propose an approach for integrating external commonsense knowledge in a neural network for cloze-style reading comprehension, published at ACL 2018 (Mihaylov and Frank, 2018), and its adaptation for integrating commonsense and domain knowledge for multi-choice science question answering, published at EMNLP 2018 (Mihaylov et al., 2018).

In Chapter 4 we describe a novel approach for transferring linguistic knowledge from supervised language tasks (‘skills’) to machine reading comprehension and analyze its impact on the task. This work has been presented at the Workshop on Learning with Limited Labeled Data at NeurIPS 2017.

In Chapter 5 we describe an approach for leveraging multiple types of linguistic annotations into a self-attention Transformer architecture for narrative machine reading comprehension. This paper is presented at the EMNLP-IJCNLP 2019 conference.

The last chapter (Chapter 6) summarizes our work and findings and proposes future research directions.

## 1.5 Publications

The contributions, described of this thesis have been published on several conferences and workshops:

- **Todor Mihaylov**, Anette Frank (2016). Discourse Relation Sense Classification Using Cross-argument Semantic Similarity Based on Word Embeddings. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task, 2016*
- **Todor Mihaylov**, Zornitsa Kozareva, Anette Frank (2017). Neural Skill Transfer from Supervised Language Tasks to Reading Comprehension. *Workshop on Learning with Limited Labeled Data (LLD) at NIPS 2017*
- **Todor Mihaylov**, Anette Frank (2018). Knowledgeable Reader: Enhancing Cloze-Style Reading Comprehension with External Commonsense Knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*
- **Todor Mihaylov**, Peter Clark, Tushar Khot, Ashish Sabharwal (2018). Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering, In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*
- **Todor Mihaylov**, Anette Frank (2019). Discourse-Aware Semantic Self-Attention for Narrative Reading Comprehension, In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*

The following articles are related, but will not be discussed in detail:

- **Todor Mihaylov**, Anette Frank (2017). AIPHES-HD system at TAC KBP 2016: Neural Event Trigger Span Detection and Event Type and Realis Disambiguation with



Word Embeddings. In *Proceedings of the TAC Knowledge Base Population (KBP) 2016*

- **Todor Mihaylov**, Anette Frank (2017). Story Cloze Ending Selection Baselines and Data Examination. In *Proceedings of the Linking Models of Lexical, Sentential and Discourse-level Semantics – Shared Task 2017*
- Markus Zopf, Teresa Botschen, Tobias Falke, Ana Marasovic, **Todor Mihaylov**, Avinesh P.V.S, Eneldo Loza Mencía, Johannes Fürnkranz und Anette Frank (2018). What's Important in a Text? An Extensive Evaluation of Linguistic Annotations for Summarization. In *Proceedings of the Second International Workshop on Advances in Natural Language Processing 2018, Valencia, Spanien.*

# Chapter 2

## Background

Machine Reading Comprehension (MRC) is a language understanding task, typically evaluated in a question answering setting, where a system reads a text passage (document  $D$ ) and answers questions ( $Q$ ) about it. Recently, work on novel datasets for MRC gained a lot of attention: ‘CNN/Daily Mail’ (Hermann et al., 2015), and Who Did What (Onishi et al., 2016), Children Book Test (Hill et al., 2016) was created semi-automatically created as cloze-style tasks. bAbI (Weston et al., 2015a) was generated using templates aiming at evaluating particular reasoning skills. Later SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), and TriviaQA (Joshi et al., 2017) were created using crowd-sourcing to evaluate systems’ ability to find answers spanning to multiple tokens. these datasets covered a set of linguistic skills required to match the answers to a given question.

Given the wide variety of Machine Reading Comprehension (MRC) datasets, together with the accessible computation power that allowed the training of large neural networks gained a lot of attention from the computational linguistics community. The common approach to tackling the Reading Comprehension task was to build a complex neural model that reads a large-scale dataset and tries to learn to perform the task at once. Careful analysis of existing datasets (Sugawara et al., 2017; Chen et al., 2016a; Rajpurkar et al., 2016) shows that the RC task requires a set of language skills such as paraphrase detection, recognition of named entities, natural language inference, and understanding of the discourse, among others. While building an end-to-end neural model, trained on a large dataset and achieving state-of-the-art results, was tempting, the work presented in this thesis focused on exploring methods for using linguistic and background knowledge in neural networks for solving the task of Machine Reading Comprehension.

In this chapter, we will start with a high-level discussion of knowledge encoding in neural networks. Then we will provide the background for understanding the Machine Reading Comprehension task and formulations and existing work that influenced the contributions

of this thesis. We will start by introducing different task formulations and the datasets used to evaluate MRC systems. We will also introduce the linguistic tasks that were used in the next chapters. We will review the aspects of a common neural network architecture used for machine reading comprehension and how it was adapted by previous work. We will discuss related approaches for using external knowledge with the MRC and where the work described in this thesis stays in this line of work.

## 2.1 Knowledge Encoding in Neural Networks

In this section, we discuss the notion of knowledge in neural networks more broadly. We discuss the usage of knowledge for tasks closely related to Machine Reading Comprehension in Section 2.4. We refer to knowledge as the information that is required to perform a given task or reason about a specific context. This can be learned by a model that is trained to perform a task or external knowledge that helps with reasoning about concepts and their relations in context. In terms of its scope, we can classify the knowledge required to solve natural language tasks as task-specific and general. Task-specific is the knowledge required to learn to perform a concrete task. Such knowledge is often learned implicitly in the weights of a neural network trained on a supervised dataset that represents the task. To solve a given task, we often need prior general knowledge that might not be presented in the given text. Most often this is knowledge about the world or commonsense knowledge that can be used to reason about concepts and their relations. Such knowledge can be encoded implicitly in the parameters of a neural network that was trained on a large natural language corpus (Mikolov et al., 2013a; Pennington et al., 2014) or together with a large relational knowledge base (Riedel et al., 2010). The method is referred to as Parametric Knowledge Encoding. The knowledge can also be introduced at the inference stage after the model is trained and it is referred to as Non-parametric.

**Parametric Knowledge Encoding** Collobert and Weston (2008) demonstrated that a neural network can be trained to encode knowledge from multiple natural language tasks and transfer it across tasks to improve them simultaneously. Riedel et al. (2010) proposed to use the relations between entities from a knowledge base to better model the relation of these entities in text. Bengio (2011) argued more broadly that neural networks learn knowledge in their parameters and that this knowledge can be used at a later stage to improve other related tasks. In Section 2.4.4 we discuss in more detail how related approaches impacted the Machine Reading Comprehension task that we are targeting in this thesis. In Chapter 4 we describe our approach to learning task-specific representations from multiple tasks and

how they can be used to improve the learning of the MRC task. Mikolov et al. (2013a) and Pennington et al. (2014) demonstrated that semantic knowledge about words can be encoded in an unsupervised way by training to predict the missing word in a given token window. In Section 2.4.1 we discuss the usage of these word representations for MRC.

More recently, towards the end of conducting the research for this thesis, models like ELMO (Peters et al., 2018), GPT-1 (Radford et al., 2018a) and (Devlin et al., 2019a) were trained on a large amount of text in an unsupervised manner and have been shown to perform well on multiple tasks. Petroni et al. (2019) demonstrated that such models have encoded broad factual world knowledge. GPT-3 (Brown et al., 2020b) further demonstrated that the amount of knowledge is crucial and can be leveraged by the model to perform reasoning with a few examples presented at inference. T5 (Raffel et al., 2020) explored the limits of transfer learning with a unified text-to-text transformer, demonstrating the power of parametric encoder-decoder approaches in various natural language generation tasks. Roberts et al. (2020) looked deeper into the encoding of factual knowledge in model parameters, illustrating how neural language models can store and retrieve knowledge without external data sources. While the parametric approaches have shown amazing results, they have multiple drawbacks: i) The knowledge that is encoded in the neural network parameters needs to be available during training. ii) To encode a large amount of knowledge, the models need to be large which makes their training and inference expensive (Brown et al., 2020b). iii) Encoding the knowledge in the parameters often results in inaccurate recall of facts at inference (Petroni et al., 2019). To address these limitations, many works have developed approaches for augmenting the neural networks with additional retrieved knowledge at inference, without encoding it in their parameters, hence referring to the methods as Non-parametric.

**Non-parametric Knowledge Encoding** In contrast to encoding the knowledge implicitly, non-parametric knowledge encoding uses external information sources. In Chapter 3, we proposed to teach the reading comprehension model to leverage external commonsense knowledge at inference to improve performance. Work by Lample et al. (2019) on large memory layers with product keys contributed to the non-parametric approaches by providing an innovative method for storing and accessing large amounts of external data efficiently. The work of Lewis et al. (2020b) demonstrated the effectiveness of external sources on retrieval-augmented generation for knowledge-intensive natural language tasks. REALM (Guu et al., 2020a) introduced a pre-training method that incorporates retrieval mechanisms, highlighting the importance of external knowledge sources even at a larger scale. Izacard and Grave (2021) discussed enhancing generative models for open-domain question answering with passage retrieval, highlighting the benefits of external data. Fan et al. (2021) augmented Transformers

(Vaswani et al., 2017) with KNN-based composite memory for dialog, demonstrating the effectiveness of external knowledge memory in dialogue systems. Chen et al. (2023) explored augmenting pre-trained language models with question and answer memory for open-domain question answering, highlighting how external question-answer pairs can enhance model performance. While augmenting neural networks with external knowledge sources tackles many of the problems of parametric approaches mentioned above, it often reduces throughput at inference since it requires performing the retrieval from usually big sources of data. To tackle this in the context of generative large language models, Wu et al. (2022) combined the ideas of parametric and retrieval-augmented approaches into an Efficient Memory-Augmented Transformer. The proposed approach encodes knowledge into a key-value memory and introduces new pre-training tasks that help the model learn how to integrate the knowledge into the network.

To summarize, parametric methods that encode knowledge into the models are limited by their training data and although they are very good at reasoning, often produce inaccurate results. On the other hand, non-parametric models improve the accuracy of certain knowledge tasks with up-to-date information and make the process more trustworthy at the expense of inference throughput and search in large knowledge bases.

## 2.2 Machine Reading Comprehension

Originally the task of reading comprehension has been used for testing the understanding of school students. In terms of machine reading comprehension, the task was initially adapted by (Hirschman et al., 1999) who proposed using it to evaluate an automatic language comprehension model. This was extended by (Breck et al., 2001) who annotated 75 stories with natural language questions, whose answers were entire sentences in the story context. To better understand how the machine reading comprehension task evolved we will discuss the most widely used datasets in the field. We group the datasets by task formulation (how the answer is presented) or the requirements for additional (external) knowledge.

### 2.2.1 Task Formulations

The task of Machine Reading Comprehension has several formulations, implemented by recent datasets, based on the answer selection type.

**Reading Comprehension as Multi-Choice Question Answering** One of the first MRC datasets that got wider adoption was *MCTest* (Richardson et al., 2013) which used a scalable

crowdsourcing approach and annotated 500 stories with multiple **multi-choice questions**. The dataset consists of almost 600 children’s and everyday stories with four multi-choice questions each. The dataset aims to evaluate reasoning understanding of natural language and reasoning. An example of multi-choice MRC from the dataset is shown in Figure 2.2.

More recently Lai et al. (2017) built a large-scale machine comprehension dataset created by collecting reading comprehension exams for Chinese students learning English. By its structure, the questions and documents are similar to *MCTest* but the dataset is much larger and contains enough examples for training large neural models. Some of the examples are also framed as cloze-style questions.

In contrast to other multi-choice datasets, *MultiRC* (Khashabi et al., 2018a) contains questions that have multiple correct answers. This made the task even more challenging because a model that tried to solve this dataset had to determine how many of the answers were valid rather than just pick the most probable.

**Reading Comprehension with Cloze-style Evaluation** is another task presentation. Recently several large-scale, automatically generated datasets for cloze-style reading comprehension gained a lot of attention. These include the *CNN/Daily Mail* (Hermann et al., 2015), *WhoDidWhat* (Onishi et al., 2016) and the *Children’s Book Test (CBTest)* data set (Hill et al., 2016). Originally, cloze-style reading comprehension is a setting where the reader is presented with a passage with a randomly removed (missing) word, and it is required to fill the gap with a word from the context. In the Natural Language Processing community, this formulation is often anecdotally considered to be just ‘gap-filling’, instead of reading comprehension, although the task itself often requires looking at the broader context and often requires reasoning about it (Hill et al., 2016). Indeed cloze-style tests measuring reading comprehension skills of human subjects using cloze-style reading comprehension tests are considered to have a high correlation with conventional human-created datasets (Bormuth, 1968a) and multi-choice reading comprehension (Bormuth, 1968b). The earliest large-scale cloze-style dataset ‘CNN/Daily Mail dataset’ (Hermann et al., 2015) has been automatically collected from a large number of news articles from CNN and Daily Mail news websites. In this dataset, a news article is used as a document context and the questions are generated by removing an entity from the given summary highlights<sup>1</sup>. The main goal pursued with this dataset is to stimulate the development of models that memorize facts mentioned in the discourse concerning actual (but anonymized) named entities, which typically constitute the answer to the question. Later *Who-Did-What (WDW)* (Onishi et al., 2016) was

---

<sup>1</sup>See *Story Highlights* section in <http://edition.cnn.com/2017/03/08/politics/white-house-wikileaks-donald-trump-cia-documents/index.html> for an example.

**Story Context:** Once upon a time there a little girl named Ana. Ana was a smart girl. Everyone in Ana's school knew and liked her very much. She had a big dream of becoming spelling bee winner. Ana studied very hard to be the best she could be at spelling. Ana's best friend would help her study every day after school. By the time the spelling bee arrived Ana and her best friend were sure she would win. There were ten students in the spelling bee. This made Ana very nervous, but when she looked out and saw her dad cheering her on she knew she could do it. The spelling bee had five rounds and Ana made it through them all. She was now in the finals. During the final round James, the boy she was in the finals with, was given a really hard word and he spelled it wrong. All Ana had to do was spell this last word and she would be the winner. Ana stepped to the microphone, thought really hard and spelled the word. She waited and finally her teacher said "That is correct". Ana had won the spelling bee. Ana was so happy. She won a trophy. Ana also won a big yellow ribbon. The whole school was also happy, and everyone clapped for her. The whole school went outside. They had a picnic to celebrate Ana winning.

1: What made Ana very nervous?

- \*A) The other ten students
- B) Her best friend
- C) The bright lights
- D) The big stage

2: Where did the school have the picnic?

- A) The gym
- B) Ana's house
- \*C) Outside
- D) Ana's classroom

3: What was Ana's big dream?

- A) Becoming a ballerina
- B) Becoming a famous singer
- C) Becoming class president
- \*D) Becoming spelling bee winner

4: Who helped Ana study everyday?

- A) Her dad
- \*B) Her best friend
- C) Her mom
- D) Her sister

Fig. 2.1 Example context and multiple multi-choice questions from the MCTest dataset. \* indicates the correct answer.

created in a similar way to ‘CNN/Daily mail’. It was derived from the Gigaword English corpus and provides cloze-style questions whose answers are entities. In order to avoid some paraphrase matching biases, in contrast to the *CNN/DM* corpus, the cloze-style questions in *WDW* are derived from a different article than the context. The questions (around 200 000) are written by crowd workers. In addition, the questions in the dataset are filtered so they are not answerable by simple baselines. They are answerable by human annotators with 84% accuracy. *Children’s Book Test (CBTest)* (Hill et al., 2016) is a cloze-style dataset that examines how well a natural language understanding model captures the meaning in children’s books. The dataset contains subsets on guessing missing *Named Entities*, *Common Nouns*, *Verbs*, and *Prepositions* with the common noun setting being the most challenging. An example of the *Common Nouns* subset is shown in Figure 2.1.

In this thesis, we use *Common Nouns* and *Named Entities* subsets of this dataset for evaluating our method for incorporating external commonsense knowledge into a neural network model, presented in Chapter 3. When we conducted this research, The CB Test dataset seemed the most prominent large-scale dataset that allowed training an end-to-end neural model and contained queries that would benefit from commonsense knowledge.

**Reading Comprehension as Span-based Question Answering** Initially, cloze-style reading comprehension has been accepted by the community due to its cheap way of producing large-scale datasets. The power of end-to-end neural network models trained to answer such queries was impressive and it was overseen as a way for retrieving information from documents, in contrast to existing information retrieval models. However, in the real world, humans usually form the query for retrieving information, as a grammatical question ("Who is the chancellor of Germany?") rather than statements with a missing token ("The chancellor of Germany is XXXX?") and the correct answer can contain multiple tokens ('Angela Merkel'). Therefore, there was a need for *span-based reading comprehension* datasets to fill the gap.<sup>2</sup> In span-based MRC, a system is given a passage of text (context) and natural language question and is required to select an answer span in the context.

*SQuAD* (Rajpurkar et al., 2016) (Stanford Question Answering Dataset) was the first large-scale RC dataset (about 100 000 questions) that contains natural language questions and answers (See Figure 2.3). It has been created using crowd-sourced questions from paragraphs from about 500 Wikipedia articles. To generate questions, workers were presented with a paragraph as a context and were asked to come up with questions that have an answer that appears as a span in this context. Later, a new version of the dataset was released that contained also unanswerable questions Rajpurkar et al. (2018).

---

<sup>2</sup>Pun intended!



**Story Context:** “ How can you be so absurd ? ” cried the queen . “ How often must I tell you that there are no fairies ? And even if there were – but , no matter ; pray let us drop the subject . ” “ They are very old friends of our family , my dear , that ’s all , ” said the king timidly . “ Often and often they have been godmothers to us . One , in particular , was most kind and most serviceable to Cinderella I . , my own grandmother . ” “ Your grandmother ! ” interrupted her majesty . “ Fiddle-de-dee ! If anyone puts such nonsense into the head of my little Prigio – ” But here the baby was brought in by the nurse , and the queen almost devoured it with kisses . And so the fairies were not invited ! It was an extraordinary thing , but none of the nobles could come to the christening party when they learned that the fairies had not been asked . Some were abroad ; several were ill ; a few were in prison among the Saracens ; others were captives in the dens of ogres . The end of it was that the king and queen had to sit down alone , one at each end of a very long table , arrayed with plates and glasses for a hundred guests – for a hundred guests who never came ! “ Any soup , my dear ? ” shouted the king , through a speaking-trumpet ; when , suddenly , the air was filled with a sound like the rustling of the wings of birds . Flitter , flitter , flutter , went the noise ; and when the queen looked up , lo and behold ! on every seat was a lovely fairy , dressed in green , each with a most interesting-looking parcel in her hand . Do n’t you like opening parcels ?

**Cloze-style query:** The king did , and he was most friendly and polite to the XXXXX.

**Candidates:** grandmother majesty fairies baby king air others dens friends

**Answer:** fairies

Fig. 2.2 Example from the Common Noun subset of the cloze-style reading comprehension dataset CBTest. XXXXX is the question placeholder that has to be replaced with the correct choice.

**Context (Paragraph):** The Rhine (Romansh: Rein, German: Rhein, French: le Rhin, Dutch: Rijn) is a European river that begins in the Swiss canton of Graubünden in the southeastern Swiss Alps, forms part of the Swiss-Austrian, Swiss-Liechtenstein border, Swiss-German and then the Franco-German border, then flows through the Rhineland and eventually empties into the North Sea in the Netherlands. The biggest city on the river Rhine is Cologne, Germany with a population of more than 1,050,000 people. It is the second-longest river in Central and Western Europe (after the Danube), at about 1,230 km (760 mi),[note 2][note 1] with an average discharge of about 2,900 m<sup>3</sup>/s (100,000 cu ft/s).

**Questions:**

**Q:** What is the largest city the Rhine runs through?

**Ground Truth Answers:** Cologne, Germany | Cologne, Germany | Cologne

**Q (SQuAD2.0):** In what country does the Danube empty?

**Ground Truth Answers:** <No Answer>

Fig. 2.3 Example of a context and question from SQuAD 1.0 ((Rajpurkar et al., 2016)) and unanswerable question from SQuAD 2.0 (Rajpurkar et al., 2018) (Q2). The spans containing the answer are underlined. If multiple ground truth spans overlap, the underline is on the combined text span.

Trischler et al. (2017) created *NewsQA* in a similar way to SQuAD from more than 10,000 news articles from CNN. The questions are collected using a multi-stage crowdsourcing process including collecting questions, finding their answers, and validation. The process ensures that the answering of the generated questions requires reasoning beyond word matching and paraphrasing. To make the dataset closer to a real-world setting, this is the first dataset that also presents unanswerable questions - questions that do not contain an answer in the presented context.

Several span-prediction machine reading comprehension datasets have been later constructed, inspired by open domain question answering <sup>3</sup> including TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), NaturalQA (Kwiatkowski et al., 2019). Instead of using a single coherent context like previous work, these datasets provide the system with a collection of documents collected from an external source or search engine.

*TriviaQA* (Joshi et al., 2017) collects around 95k questions written by trivia enthusiasts. The questions are long compositional questions that are supposed to require reasoning from multiple sentences or background knowledge. Another difference from previous datasets is that the questions are created independently from the context. Moreover, the context

<sup>3</sup>TREC open question answering challenge <https://trec.nist.gov/data/qamain.html>

is constructed from several evidence paragraphs extracted from sources like Wikipedia or Bing.com web search. Similarly to *TriviaQA*, *SearchQA* (Dunn et al., 2017) collects questions from trivia questions - in this case, Jeopardy question-answer pairs from j-archive.com. To collect evidence snippets for the context (Dunn et al., 2017) uses the Google search engine to retrieve relevant text, given the question as a query.

Most recently Kwiatkowski et al. (2019) introduced *Natural Questions* - a machine reading comprehension dataset with questions generated from real users on the Google search engine. In contrast to others, instead of having as a context only paragraphs (parts of stories, etc), it provides entire Wikipedia pages as a context and contains more than 300k questions.

**Reading Comprehension as Question Answering with Open Answers** Finding an answer in a context through span-selection is convenient for training a machine learning model but it does not always represent how a human would answer a given question. Therefore, several datasets were created with the answer as a free text, independent of the context. Nguyen et al. (2016) used more than 1 million questions and answers from the Bing search engine to create *MS Marco*. The given context is a set of passages retrieved from relevant to the query documents. The answer is generated by first selecting the passages that contain an answer and then summarizing in a single human-written answer. The dataset also provides a sub-task of ranking relevant passages which allows the evaluation of information retrieval techniques as well. *NarrativeQA* (Kociský et al., 2017) is a reading comprehension dataset about narrative stories. The dataset has two settings: *summary* - which requires answering questions about a summary of 400 to 1200 tokens and *full* where the questions are generated by looking at the summary but the given context is a very long text (up to 50k tokens) from an actual book or a movie script. The dataset is interesting for evaluating semantic and discourse phenomena. Its stories are (as the title suggests) narratives and answering questions requires reasoning between character relations in the discourse such as ‘Why did Jericho replace Picard as captain of Enterprise?’. We use this dataset to evaluate our approach for combining semantic and discourse knowledge with a self-attention mechanism in a neural network model for machine reading comprehension, presented in Chapter 5.

**Multi-hop Reading Comprehension** Since it has been shown (Chen et al., 2016a,b; Weissenborn et al., 2017b) that many of the MRC formulations and datasets mentioned above contain a high number of questions that can be answered easily by word or paraphrase matching, there was a need for more complex and challenging datasets that require reasoning beyond a single sentence.

*WikiHop* and *MedHop* (Welbl et al., 2018) are reading comprehension datasets that were constructed to evaluate complex multi-hop reasoning. They require combining information from multiple pieces of evidence (paragraphs) to reach the answer. The context consists of multiple paragraphs. The answers to the questions should be selected by choosing an entity from the entities in the context. The proposed method for the dataset construction starts with a fact  $(s, r, o)$  from a knowledge graph and forms a question  $q = (s, r, ?)$  and answer  $a = o$ . The context paragraphs are then selected so that the entities in them connect to the answer through at least one other paragraph. The method is used for constructing datasets from Wikipedia (*WikiHop*) and the domain of molecular biology (*MedHop*). *HotpotQA* (Yang et al., 2018) is another multi-hop reading comprehension dataset that requires reasoning across multiple sentences. The questions in the dataset are from various topics. The context is presented as a set of sentences and to reach the answer a model should collect information from a chain of them. In addition, the exact sentences required for the reasoning are annotated so the models can be trained for reasoning and interpretability.

Although many of the reading comprehension datasets are designed to be answered only with the context, they would benefit from external background or commonsense knowledge, in addition to the given context.

**Machine Reading Comprehension with External Knowledge** In Chapter 3, we focus on knowledge-enhanced reading comprehension where additional knowledge is used to complement the document context. At the time our work in this area was conducted, there were no reading comprehension datasets that were designed to use external knowledge, so we extended the *Children’s Book Test* (Hill et al., 2016) with knowledge from ConceptNet. Later, several machine comprehension datasets focused on this setting where a context is provided but in order to answer a question, a system could or should use some external knowledge.

*MCScript* (Ostermann et al., 2018) is a dataset that was built especially with the aim of collecting questions that require both the usage of a context and additional script knowledge. The examples in the dataset contain a context which is an everyday story about a specific event such as ‘going to a restaurant’ or ‘preparing food’ and the questions provided can be answered using the context alone or combining a statement from the context and external script knowledge.

In a similar fashion, *ProPara* (Mishra et al., 2018) requires answering questions about changing specific procedural states in processes such as producing electricity, photosynthesis, etc. In order to answer a question the model should understand the process states and constraints and include additional commonsense knowledge.

*OpenBookQA* (Mihaylov et al., 2018) is a dataset modeled after open book exams for assessing human understanding of a subject and most questions require multi-hop reasoning over knowledge facts. The almost 6000 multi-choice questions set comes with an open book (common context) of 1329 elementary-level science facts. The questions probe an understanding of these facts and their application to novel situations in a multi-hop fashion. This requires combining an open book fact (e.g., metals conduct electricity) with broad common knowledge (e.g., a suit of armor is made of metal) obtained from other sources. While existing QA datasets over documents or knowledge bases focus mainly on linguistic understanding, *OpenBookQA* probes a deeper understanding of the language in the context of common and commonsense knowledge. Therefore the dataset can also be seen as machine comprehension over a ‘partial context and external knowledge’ since the open book that is provided is not sufficient for answering the questions.

This dataset was used for evaluation of the commonsense knowledge-enhanced neural model, presented in Chapter 3.

## 2.2.2 Related Natural Language Processing Tasks

Sugawara et al. (2017); Chen et al. (2016a); Rajpurkar et al. (2016) shown that the *Reading Comprehension* task, represented by existing Machine Reading Comprehension datasets requires a set of natural language ‘skills’ such as paraphrase detection, recognition of named entities, natural language inference, and understanding of the discourse. In this thesis, we explore approaches to teach neural networks to perform several of these skills and combine them to solve the more complex Machine Reading Comprehension task. The natural language skills referred to above are often resembled by datasets constructed for building *Natural Language Processing* systems.

Here we briefly describe several natural language ‘skill’ tasks important for this work and their formulation. More details about the dataset and the usage of these tasks are presented in Chapter 4 and 5.

**Question Answering** is a task that requires a system to answer natural language questions. Typically the questions can be answered by different sources - knowledge bases, text corpus, etc. Recently QA systems are built using a two-stage system: an Information Retrieval module to retrieve documents from a large corpus (such as Wikipedia) and an MRC model to read through selected documents and find the answer of the question.

**Named Entity Recognition** (NER) is a task that aims at recognizing entities and their types in natural language text. The task is often formed as a token annotation task (Tjong

Kim Sang, 2002) that requires assigning a label to a span from the text. An example is *[Starfleet]<sub>ORG</sub> assigned [Captain Jellico]<sub>PER</sub> as [Picard]<sub>PER</sub>'s replacement*, where *Starfleet* is an entity with type Organization and the rest of the entities are of type *Person*.

**Shallow Discourse Parsing** (SDP) Xue et al. (2016b) is a task that aims at identifying two discourse arguments in the text (text spans or entire sentences) and recognizing the discourse relation between them. The main types of discourse relations, in terms of their presentation of the discourse, are *Explicit* and *Implicit*.

*Explicit* discourse relations are usually present when the two discourse arguments (Arg1 and Arg2) are connected with an *explicit connective* such as *but*, *because*, *when*, etc. *Implicit* discourse relations are identified when there is no explicit connective that characterizes the relation between the arguments.

**Semantic Role Labeling** Semantic Role Labeling (Palmer et al., 2005) is a task that recognizes relations between predicates and their arguments in a sentence. An example is *[Starfleet]<sub>A0</sub> [assigned]<sub>V</sub> [Jellico]<sub>A1</sub> [as Picard's replacement]<sub>A2</sub>*. Similarly to NER, this task is formulated as a token sequence labeling task.

**Coreference Resolution** Coreference Resolution (Hobbs, 1978) aims at identifying the occurrence of references to the same entity in a given text. For example in the text: *Captain [Picard]<sub>C1</sub> went on a mission. [Cardassians]<sub>C2</sub> took [him]<sub>C1</sub> as a prisoner. [They]<sub>C2</sub> accused [him]<sub>C1</sub> in terrorism*, *Picard* and *Cardassians* are two entities that have references (*they*, *him*) in multiple sentences.

**Natural Language Inference** Entailment or Natural Language Inference [cite] detects if two two phrases or sentences (usually referred as *Premise* and *Hypothesis*) are in entailment relations (see Table 2.1).

Premise	Relation	Hypothesis
Animals need food to survive.	Entailment	Dogs need food every day.
The dog is man's best friend.	Neutral	Cats like playing with kids.
Everyone is staying at home.	Contradiction	Streets are full of people.

Table 2.1 Examples of different natural language inference relations

**Text Classification** Many natural language processing tasks are based on text classification where a model is required to assign a label to a given text. In this work, we use the tasks of

*Question Type Classification* and *DBPedia text classification*. *Question Type Classification* (Li and Roth, 2002a) was created with the idea that models that are able to perform well on the task can be used as a component in an automatic QA system. Questions are classified in 6 high-level classes (*Abbreviation*, *Entity*, *Description*, *Human*, *Location*, and *Numeric Value*) and 50 fine-grained classes. *DBPedia text classification* is a dataset that challenges systems to classify the types of entities, given either their name (ex. *President Obama*) or short description (ex. *Barack Obama served as the 44th president of the United State of America.*) available in the DBPedia (Auer et al., 2007) knowledge base.

**Language Modeling** (LM) is an artificial task that evaluates to what extent a machine learning model can ‘guess’ the next word in a continuous text. To do that statistical and neural models are trained to estimate the probability distribution  $P(w_i|w_{i-(n-1)}..w_{i-1})$  for each word  $w_i$  in a text, given the previous  $n - 1$  words.

## 2.3 Neural Network Approaches for Machine Reading Comprehension

### 2.3.1 Common Architecture

Depending on the target dataset and the required information for solving it, the architecture of a Machine Reading Comprehension system may vary. However, given the specificity of the MRC tasks, the majority of proposed models based on neural networks used a general architecture which we summarize in Figure 2.4. Most neural network-based systems read an input and pass it to a **Neural Reader**. This module is usually an end-to-end trained model that outputs the answer to a given query. Optionally, if the given context  $D_{1..N}$  is a multi-paragraph or a very long context, too big to be processed as a whole by the Neural Reader, only a specified number of paragraphs can be retrieved by a **Document Retriever** module. Another module that could be integrated is a **Feature Extraction** module. It extracts features based on external annotations and heuristics that are usually used as additional input by the Neural Reader module.

After the Neural Reader module is been trained, the final answers predicted for a question can be re-ranked by combining multiple external rules or models (ex. TF-IDF, PMI, Language Model, etc.), or simply by using an ensemble (See Section 2.3.3) of the predictions of multiple instances of the model trained with different initialization.

In Chapter 3, we propose to use a **Knowledge Retrieval** module to enhance the original MRC task input ( $D_{1..N}$ ,  $Q$ ,  $C_{1..M}$ ) with an external set of background or commonsense

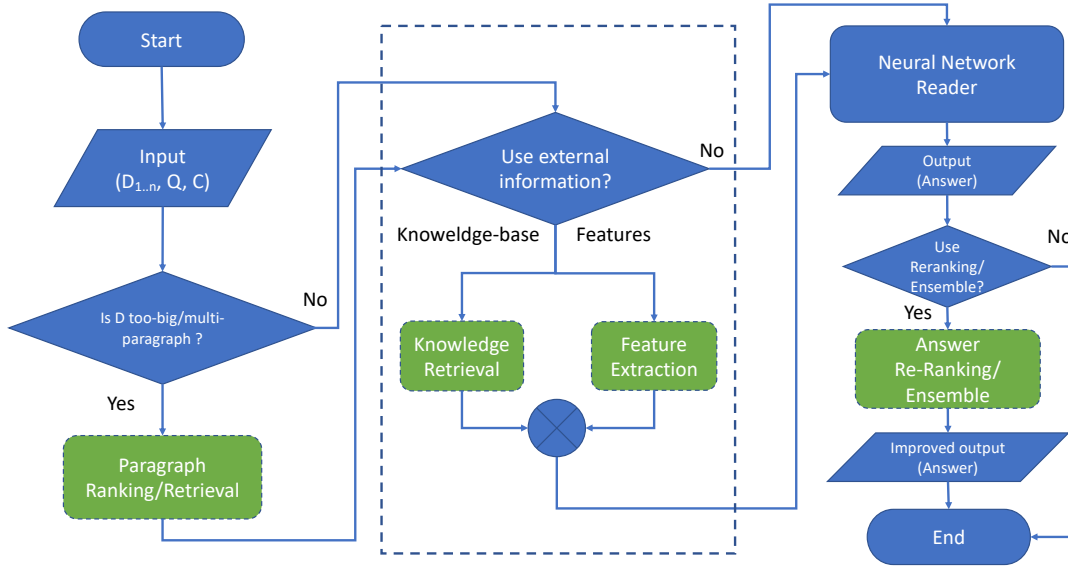


Fig. 2.4 Flowchart of general architecture of a MRC/QA system.  $D_{1..N}$  are one or more paragraph documents,  $Q$  is a question and  $C_{1..M}$  are number of candidates in terms of multi-choice answer candidates. The blue modules (solid line) are common for all neural network-based systems. Modules in green (punctuated line) are optional. Modules in orange (punctuated dotted line) are proposed in our contributions.

knowledge facts  $K$ , retrieved from an external knowledge-base such as ConceptNet, and used by the Neural Reader.

### 2.3.2 Neural Reader

Figure 2.5 shows a common neural architecture for MRC with an optional knowledge integration module. The neural reader is a system that is trained end-to-end on the MRC task. It usually reads the input document (or passage) and the given question and tries to select the answer, depending on the task formulation (span detection, generation, multi-choice, etc.). It has several layers that take care of different parts of the process.

**Embedding Layer** The embedding layer takes the input tokens  $w_0..w_n$  and generates outputs vector representations  $e_0..e_n$  with size  $d_e$ :

$$e_0..e_n = \text{EmbeddingLayer}(w_0..w_n) \in \mathbb{R}^{n \times d_e} \quad (2.1)$$

$$(2.2)$$



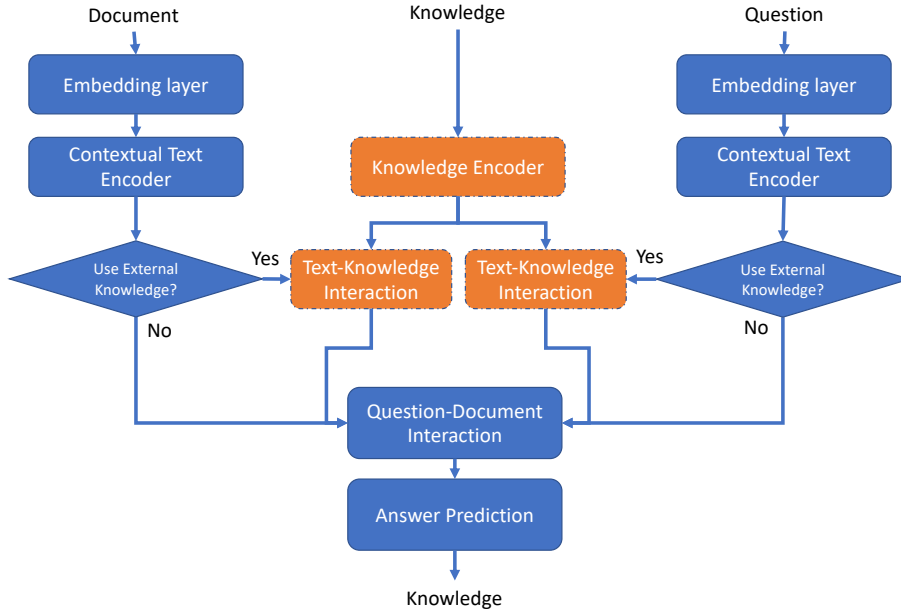


Fig. 2.5 Flowchart of common neural reader model of a MRC system. The blue modules (solid line) are common for all neural network-based systems. Modules in orange (punctuated dotted line) are proposed in this work.

The goal is to convert the input text tokens to numerical representations that can be read in the neural network models. The embedding layer is applied to both the document and the question tokens.

There are several ways to embed the word tokens into an embedding vector that have varied success based on the task and dataset.

**Lookup Word Embedding** Initial works in the field initially have used pre-trained word embeddings where each word token (ex. *dog*) maps to exactly one embedding vector:

$$e_i^{lookup} = LookupEmbedder(w_i, E) \in \mathbb{R}^d \quad (2.3)$$

$E$  is an embedding matrix with size  $n_v \times d$ , where  $n_v$  is the number of unique tokens in the vocabulary and  $d$  is the size of the embedding.  $E$  can be initialized with pre-trained word embeddings like GLOVE (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013b) or randomly. The different missing tokens in the pre-trained model are either initialized randomly with different values or are replaced with the same vector corresponding to a single *UNK* token.

Different pre-trained embedding models have different impacts on the performance of the model, and this often depends on the reading comprehension task and the domain (Dhingra

et al., 2017a; Mihaylov and Frank, 2017a). The size of the lookup word embedding vectors can vary and it is often treated as a hyper-parameter when the embeddings are initialized randomly. While using pre-trained embedding might be often beneficial, early work (Kadlec et al., 2016) has also found that for training end-to-end neural models on large datasets, it is also sufficient to initialize the embedding vectors randomly for all tokens in the vocabulary.

**Character-based Embeddings** Another type of representing tokens is using character-based embedding. In contrast to lookup embedding where each word is encoded with a corresponding vector, here each token is encoded as a sequence of characters. Each character in the token is embedded with a different randomly initialized vector with a relatively smaller size ( $d_c = 20, 50, \text{etc.}$ ) than the word-level embeddings. Then, the word-level character embeddings are obtained by processing the set character with a convolutional neural network (Zhang et al., 2015b) or recurrent neural network (Luong and Manning, 2016; Ling et al., 2015) and outputs a single vector for each word token. Character-based encoding  $e_i^{char}$  of a single word token  $w_i$  with characters  $w_0^c..w_{n_c}^c$  can be formalized as:

$$e_i^{char} = CharEncoder(LookupEmbedder(w_0^c..w_{n_c}^c)), \quad (2.4)$$

where *CharEncoder* can be *CNN*, *RNN*, or *Transformer* encoder. The benefit of character-based word embedding is that tokens that are morphologically close such as *buy*, *buying*, and *buyer* could be represented in similar vector spaces. Character-level embedding of the words can also help with out-of-vocabulary words. For example, unseen words in the evaluation such as *Anna* will have a similar representation to others that might be seen during training (*Ana*) or share a property that the model can learn (ex. person names start with a capital letter). Character-based word embeddings and lookup embeddings from a pre-trained model are often used together by simply concatenating the output of both:  $EmbeddingLayer(x): concat(CharEncoder, LookupEmbedder(w))$  or using more sophisticated mechanisms (Yang et al., 2017a).

**Pre-trained Contextualized Word Representations** Recently Peters et al. (2018) introduced ELMO (Embeddings from Language Models), deep contextualized embeddings derived from pre-trained bi-directional language models. These have been shown to work great as a replacement for conventional word embedding models and improve the state-of-the-art on multiple tasks (Peters et al., 2018). Other pre-trained contextualized models have been proposed such as GPT (Radford et al., 2018b), ULMFit(Howard and Ruder, 2018a), and most recently BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al.,

2019a). When used as a replacement for standard lookup embeddings or character-based embeddings alone are improving greatly the state-of-the art in multiple tasks.

The pre-trained contextualized embeddings can be used as a fixed encoding layer or fine-tuned during the learning of the RC task. This depends on many factors such as the size of the pre-trained model, the hidden size of the embeddings, and the training data size can often be used as a hyper-parameter (tune vs. not to tune). When fine-tuned these embeddings can be used as a replacement for the standard context encoder that is a fundamental part of most neural readers.

**Context Encoding** After the input tokens of the document and question are converted to vector representations by the embedding layer, the neural models usually include an additional *context encoder* layer. The purpose of this layer is to learn contextualized representation of the input, that is trained especially for the task. That is in contrast to the embedding layer which (when not fine-tuned) can provide general word representations. We process the output of the embedding layer with an *Encoder* to obtain context-encoded representations for document ( $c_{d_{1..n}}^{ctx}$ ) and question ( $c_{q_{1..m}}^{ctx}$ ) encoding:

$$c_{d_{1..n}}^{ctx} = \text{Encoder}^{ctx}(e_{d_{1..n}}) \in \mathbb{R}^{n \times h} \quad (2.5)$$

$$c_{q_{1..m}}^{ctx} = \text{Encoder}^{ctx}(e_{q_{1..m}}) \in \mathbb{R}^{m \times h}, \quad (2.6)$$

where  $d_i$  and  $q_i$  denote the  $i$ th token of a text sequence  $d$  (document) and  $q$  (question), respectively,  $n$  and  $m$  is the size of  $d$  and  $q$ , and  $h$  is the vector (hidden) size of the output. The most used architectures for contextual encoding in the machine reading comprehension literature are Recurrent Neural Networks and Transformers.

**Recurrent Neural Network (RNN) Encoders** Until the invention of self-attention transformer models Vaswani et al. (2017) RNN models such as LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Chung et al., 2014) were de-facto standard for building contextual representations due to their ability to learn contextual dependencies and ease of use in existing deep learning frameworks (Abadi et al., 2015; Paszke et al., 2019).

**Transformers** Since RNN-based models require a lot of time for execution, due to their recurrent nature, Vaswani et al. (2017) introduced the Transformer encoder that uses multi-layer multi-head self-attention to build efficient contextual representations. *QANet* Yu et al. (2018) was the first to use the *Multi-Head Self-Attention* model as a context encoder to speed-up computation and allow training RC models at scale. Since 2018 this became a standard in context encoding for machine reading comprehension models. In Chapter 5 we

discuss the architecture and limitations of a standard self-attention architecture and propose a method to augment it with linguistic knowledge to improve its performance for machine reading comprehension.

**Question-Context Interaction** The contextual representation of document and question are often encoded with separate encoders (Chen et al., 2016a; Kadlec et al., 2016; Dhingra et al., 2017b). This is motivated by the assumption that the question usually has different grammar than the context and there is a need to better learn the alignment to the document tokens (Chen et al., 2016a; Weissenborn et al., 2017b). An important part of the neural reader in machine reading comprehension architectures is the question-context interaction between the contextually encoded representations of question and context. The goal of this module is to create neural representations of the context (Hermann et al., 2015), based on the question and/or context-aware representation of the question, usually using neural attention methods (Dzmitry Bahdanau et al., 2014).

Depending on the number of interactions between the question and context representations, the interaction layers can be classified as *single-turn* and *multi-turn*.<sup>4</sup> Early work introduced simple but good *single-turn models* (Hermann et al., 2015; Kadlec et al., 2016; Chen et al., 2016a), that read the document once with the question representation ‘in mind’ and select an answer from a given set of candidates. In these models, the context-question interaction layer and the answer selection layer are combined.

Initially Hermann et al. (2015) introduced the *Attentive Reader* as a simple but good baseline for the task of cloze-style reading comprehension. In their work Hermann et al. (2015) used the final states from a bi-directional LSTM encoder to obtain a contextualized representation of the question as a single vector  $r_q^{ctx} \in \mathbb{R}^{2h}$ . This was then used to retrieve weighted representation from the encoded document context and combine it with a non-linear layer to predict the answer from the vocabulary.

Chen et al. (2016a) introduced the *Stanford Reader* which used a higher number of parameters and more complex attention functions to achieve competitive results.

*Stanford Reader* and *Attentive Reader* can be classified as *aggregation* models since they obtain single aggregated representations of the context, given the question, and then infer the answer. The early aggregation-based models were soon outperformed by models that represent the interaction between question and documents tokens explicitly for cloze-style reading comprehension.

---

<sup>4</sup>*Turn* refers to the process of going from a contextual representation to question-aware contextual representation

Kadlec et al. (2016) introduced the *Attentive Sum Reader (ASR)* which instead of building an attention-weighted sum of the context and inferring the answer, *points* to the answer in the text. Additionally, they sum the attention of all occurrences of every candidate answer in the context. At the same time, Kobayashi et al. (2016) proposed a *Dynamic Entity Representation* model that similarly to Kadlec et al. (2016) keeps track of the entity states in order to infer the desired answer.

More complex models (Weston et al., 2015c; Dhingra et al., 2017b; Cui et al., 2017; Munkhdalai and Yu, 2016; Sordoni et al., 2016) perform *multi-turn reading* of the story context and the question, before inferring the correct answer. Weston et al. (2015c); Sukhbaatar et al. (2015) proposed *Memory Networks* that aggregate the information from consecutive sentences into long-term memory. These perform multiple steps of ‘reasoning’ over the memory blocks before inferring the answer. The proposed models were shown to perform well on several synthetic reading comprehension tasks *bAbI* (Weston et al., 2015a).

Dhingra et al. (2017c) used a model called *Gated-Attention Reader*, which has a multi-layer architecture and uses different GRU parameters for encoding the question and the context. The *Attention-over-Attention Reader* Cui et al. (2017) uses different attention over the question and context and then attention over these attention representations.

A study on some of the early models for cloze-style reading comprehension argues (Wang et al., 2016a) that all the proposed readers and pointer architectures are driven by a logical structure in the vector representations that they learn. They perform experiments on *CNN/Daily Mail*, *Who-did-What(WDW)*, and *Children’s Book Test(CBT)* datasets. Wang et al. (2016a) presented experiments supporting the existence of logical structure in the hidden state vectors of *aggregation readers* such as the *Attentive Reader* and *Stanford Reader*. They have shown that the logical structure of aggregation readers reflects the architecture of *pointer readers* such as the *Attention-Sum Reader* (Kadlec et al., 2016), the *Gated Attention Reader* (Dhingra et al., 2017b) and the *Attention-over-Attention Reader* (Cui et al., 2017).

Performing multiple hops and *modeling a deeper relation between question and document* was further developed by several models (Seo et al., 2017; Xiong et al., 2016; Wang et al., 2016b, 2017; Shen et al., 2016) on the newer generation of span-based RC datasets, e.g. SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), and TriviaQA (Joshi et al., 2017).

**Answer Prediction** The last module of a standard reading comprehension neural model is usually the *answer prediction* module. It takes the output representations from the context-question interaction representations and outputs values that can be used for decoding the answers. The prediction of the answer depends on the task formulation. In Section 2.2 we

described several task formulations in terms of answer representation. These usually have different answer-prediction modules.

In a **cloze-style** reading comprehension setting the answer is usually presented as a single token. If the original token is not in the source document used for reading, a single representation is obtained from the context and question, and the answer is inferred from the full vocabulary (Chu et al., 2016)]. If the target token is in the text, we usually use a pointer module similar to *Attention Sum Reader*. In both cases, an attention function  $Att$  is employed to select the answer:

$$P(a_i|q, d) = \text{softmax}(Att(r_q^{ctx}, c_{d_j}^{ctx})), \quad (2.7)$$

where  $r_q^{ctx}$  is a query representation and  $m_j$  is the  $j$ -th vector representation from the vocabulary or the encoded context. The neural network is trained using cross-entropy loss between the correct answer token index and  $P(a_i|q, d)$ .

In **span selection** tasks (Rajpurkar et al., 2016; Kwiatkowski et al., 2019) the answer is selected as a token span from the text document. The first span prediction dataset that took off in the community was SQuAD (Rajpurkar et al., 2016) and the proposed techniques of span prediction and evaluation have been used for several datasets (Joshi et al., 2017; Kwiatkowski et al., 2019). In this setting, the answer is represented as the indices of the *start* ( $a_{start}$ ) and *end* ( $a_{end}$ ) tokens.

To train the model we minimize the sum of the cross-entropy losses:

$$L = CE(a_{start}^{pred}, a_{start}) + CE(a_{end}^{pred}, a_{end}). \quad (2.8)$$

During inference, dynamic programming is used to select the answer with the indices  $i_{start}$  and  $i_{end}$ , such that  $i_{end} > i_{start}$ ,  $i_{end} - i_{start} \leq l_{max}^{ans}$ , and  $i_{end} * i_{start} = \max$ .

In **generative question answering**, the answer prediction requires generating a sequence of text tokens that are not necessarily in the source document. In this case, the answer prediction is executed as a sequence-to-sequence architecture, usually combining the inference from a vocabulary and a copy-mechanism (Gu et al., 2016; See et al., 2017) pointing to the source document. While some datasets such as NarrativeQA (Kociský et al., 2017) require answer generation, previous work (Hu et al., 2018b) has shown that mapping the task to span selection can yield better results than others that train the model as answer generation (Bauer et al., 2018; Wang and Jiang, 2016).

### 2.3.3 Answer Re-Ranking and Ensemble

Neural readers for reading comprehension are often evaluated based on the performance of the best model trained with this architecture. In order to measure the maximum possible performance for a given task, the neural model is often combined across multiple initializations of the same model (*Ensemble*) or re-ranked using external information about the task.

**Ensemble** Several ensemble methods have been adopted in the community. Kadlec et al. (2016) have tried different ensemble methods for cloze-style reading comprehension. They used average ensembling - the selected answer is determined by averaging the probabilities of the same neural models with different initializations. Another ensemble method that they adopted is to take the average of only the top 20% best performing on the *Dev* set, or greedy ensemble, by using a combination of models that yields the highest performance on *Dev*. However, the average ensemble worked best for their AS Reader. This has also been adopted for many other models for cloze-style RC (Cui et al., 2017; Chen et al., 2016a; Onishi et al., 2016). Simple averaging of the prediction probabilities of the output model has worked well for span-based RC where *start* and *end* probabilities are used. This was adopted by the majority of works that worked on SQuAD (Rajpurkar et al., 2016).

**Answer Re-Ranking** Another way to improve the overall performance, on top of a trained single neural model or ensemble is by using another source of information that contains prior knowledge relevant to the task. In their work Cui et al. (2017) proposed the *n-best re-ranking strategy* that includes selecting the top  $N$  answers selected by the model and using additional methods for re-scoring. For each example, the cloze query was filled with each of the  $N$  answer choices to form  $N$  sentences.

They used a combination of a Global N-gram model trained on the train data documents, a local N-gram model, trained only on the current document, and a Word-class LM (Och, 1999) to score each of the sentences. This improved the performance with an additional 4%.

## 2.4 Neural Machine Reading Comprehension with Prior Knowledge

Most work in machine reading comprehension before the work conducted for this thesis has focused on training end-to-end neural networks on large-scale datasets, without explicitly adding new data to the task. In this thesis, we focus on using additional knowledge to tackle

the reading comprehension task. Below we review existing methods for augmenting neural network models with various types of external knowledge.

### 2.4.1 Token Embeddings

While many large-scale datasets can be used to train the neural network from scratch, many have used pre-trained word embeddings (Chen et al., 2016a; Seo et al., 2017; Kumar et al., 2016; Chen et al., 2016b; Dhingra et al., 2017a; Mihaylov and Frank, 2017a) . These embeddings are usually obtained using distributional approaches for pre-training such as *Word2Vec* (Mikolov et al., 2013b) and *Glove* (Pennington et al., 2014) or additionally trained with structured knowledge (Speer and Chin, 2016).

### 2.4.2 Features

Several models have used heuristic features to improve Machine Reading Comprehension (Weissenborn et al., 2017b; Chen et al., 2016b). *FastQA* proposed by Weissenborn et al. (2017b) used *word in question* binary features combined with Bi-LSTM to reach very good performance on the SQuAD (Rajpurkar et al., 2016) dataset. For each word in the context, such *word in question* feature indicated if the word token is contained in the question. The assumption is that the answers to some questions are often found in a similar context to a question paraphrase. These features were also used in DrQA (Chen et al., 2016b). They also used an additional soft similarity feature, computed using a cosine similarity between the *Glove* word embeddings of each word token in the context and each token in the question. Chen et al. (2016b) also augments the word embedding layer with token-based features such as part-of-speech (POS) and named entity labels. These were obtained by annotating the text with existing natural language processing tools (Manning et al., 2014). Independent work by Wang et al. (2016a) also confirmed that the addition of linguistics features to the input of strong neural readers significantly boosts their performance on multiple datasets.

### 2.4.3 Combining Text and External Knowledge Sources

In Chapter 3 we propose employing knowledge from an external commonsense knowledge base to improve machine reading comprehension. Below we discuss some related neural approaches, published before our work, and applied to machine reading comprehension and related natural language processing tasks.



**Knowledge Base with External Text** The alignment of text and knowledge bases is initially explored in the context of relation extraction and semantic parsing by a wide range of work (Bunescu and Mooney, 2007; Mintz et al., 2009; Yao et al., 2010; Riedel et al., 2010). Riedel et al. (2013) proposed modeling these alignments jointly in a single model representation trained end-to-end. They proposed using a *Universal Schema* obtained by relations from structured knowledge bases and schema-free relations extracted from text. This approach improved significantly the accuracy of knowledge-base completion. Das et al. (2017a) later used the alignment from the *Universal Schema* to enhance a knowledge-base question answering neural model with textual knowledge.

In Chapter 3 we use a similar approach in the opposite setting - we propose a text-first model and enhance it with structured commonsense knowledge for cloze-style reading comprehension.

**Reading Comprehension with External Knowledge** Work similar to ours from Chapter 3 is by Long et al. (2017), who have introduced a new task of Rare Entity Prediction. The task is to read a paragraph from WikiLinks (Singh et al., 2012) and to fill a blank field in place of a missing entity. Each missing entity is characterized with a short description derived from Freebase, and the system needs to choose one from a set of pre-selected candidates to fill the field. While the task is superficially similar to cloze-style reading comprehension, it differs considerably: first, when considering the text without the externally provided entity information, it is clearly ambiguous. In fact, the task is more similar to Entity Linking tasks in the Knowledge Base Population (KBP) tracks at TAC 2013-2017, which aim at detecting specific entities from Freebase. Our work, by contrast, examines the impact of injecting external knowledge in reading comprehension, or NLU task, where the knowledge is drawn from a commonsense knowledge base, ConceptNet in our case. Another difference is that in their setup, the reference knowledge for the candidates is explicitly provided as a single, fixed set of knowledge facts (the entity description), encoded in a single representation. In our work, we are retrieving (typically) distinct sets of knowledge facts that might (or might not) be relevant for understanding the story and answering the question. Thus, in our setup, we crucially depend on the ability of the attention mechanism to retrieve relevant pieces of knowledge. Our aim is to examine to what extent commonsense knowledge can contribute to and improve the cloze-style RC task, which in principle is supposed to be solvable without explicitly giving additional knowledge. We show that by integrating external commonsense knowledge we achieve clear improvements in reading comprehension performance over a strong baseline, and thus we can speculate that humans, when solving this RC task, are similarly using commonsense knowledge as implicitly understood background knowledge.

Another work from Weissenborn et al. (2017a) was driven by similar intentions of using commonsense knowledge for natural language understanding tasks. The authors exploit knowledge from ConceptNet to improve the performance of a reading comprehension model, experimenting on the recent SQuAD (Rajpurkar et al., 2016) and TriviaQA (Joshi et al., 2017) datasets. While the source of the background knowledge is the same that we use, the way of integrating this knowledge into the model and task is completely different. (i) In our work from Chapter 3 we are using attention to select unordered facts triples using key-value retrieval and (ii) we integrate the knowledge that is considered relevant explicitly for each token in the context. The model of Weissenborn et al. (2017a), by contrast, reads the acquired additional knowledge sequentially after reading the document and question, but transfers the background knowledge implicitly, by refining the word embeddings of the words in the document and the question with the words from the supporting knowledge that share the same lemma. In contrast to the implicit knowledge transfer of Weissenborn et al. (2017a), our explicit attention to external knowledge facts can deliver insights about the used knowledge and how it interacts with specific context tokens.

**Neural Models with External Knowledge for Other NLP Tasks** Kiddon et al. (2016) integrated knowledge about food ingredients in a *neural-checklist model* to enhance text generation of recipes. They copied words from a list of ingredients instead of inferring the word from a global vocabulary. Ahn et al. (2016) proposed a language model that copies fact attributes from a topic knowledge memory. The model predicts a fact in the knowledge memory using a gating mechanism and given this fact, the next word to be selected is copied from the fact attributes. The knowledge facts are encoded using embeddings obtained using *TransE* (Bordes et al., 2013). Yang et al. (2017b) extended a *sequence-to-sequence* model with attention to embedded facts for dialogue and recipe generation and a co-reference resolution-aware language model. A similar model was adopted by He et al. (2017c) for answer generation in dialogue. Incorporating external knowledge in a neural model has proven beneficial for several other tasks: Yang and Mitchell (2017) incorporated knowledge directly into the *LSTM* cell state to improve event and entity extraction. They used knowledge embeddings trained on WordNet (Miller et al., 1990) and NELL (Mitchell et al., 2015) using the *BILINEAR* (Yang et al., 2014) model.

#### 2.4.4 Neural Transfer Learning for Machine Reading Comprehension

Above we discussed that most recent neural models use a form of transfer learning by incorporating word embeddings (Section 2.4.1) and features (Section 2.4.2).

**Transferring Knowledge Between Machine Reading Comprehension Datasets** Neural transfer learning between the same or very similar tasks in different domains often works well (Ruder, 2017). This was shown to also work well for Machine Reading Comprehension on multiple occasions. Initially Kadlec et al. examined transfer learning using neural models trained on a source MRC dataset and evaluated on a target dataset. Golub et al. (2017) proposed a novel two-stage transfer learning approach to perform out-of-domain question answering without explicit neural representation transfer learning. Instead, they first trained modules for the generation of question-answer pairs trained on *SQuAD* (Rajpurkar et al., 2016) and used the method to generate question-answer for the *NewsQA* dataset documents (Trischler et al., 2017). The generated pairs are later used for training a neural network model that performed well on the *NewsQA*. Min et al. (2017) showed supervised transfer learning from *SQuAD* to *WikiQA* (Yang et al., 2015) and community question answering (Nakov et al., 2016a) is beneficial compared to the standard training setup. Chung et al. (2018) evaluated supervised transfer learning and self-supervised iterative labeling technique and reported improvements across all target datasets. More recently, to examine the generalization capabilities of MRC models Fisch et al. (2019) has organized the MRQA 2019 Shared Task on *Evaluating Generalization in Reading Comprehension* combining several datasets for training and evaluation. The generalization evaluation has been ensured by allowing the participating teams to train on a limited set of training datasets and evaluating out-of-domain target datasets. Talmor and Berant (2019) examined the transferability and generalization between 10 datasets for machine reading comprehension using several models. They have shown that the MRC tasks greatly benefit from each other and provided a fine-grained analysis of the transfer relations between multiple datasets. The transfer between various question answering and machine reading comprehension datasets was shown to further benefit from unifying the format of the tasks (Khashabi et al., 2020).

**Supervised Transfer from Natural Language Processing Tasks** In contrast to using transfer learning between datasets and similar question answering and machine reading comprehension datasets, inspired the ability of humans to transfer skills across tasks, in Chapter 4 we examine how different lower-level linguistic tasks contribute to the machine reading comprehension task.

In natural language processing supervised transfer learning with neural models was proposed initially by (Collobert and Weston, 2008). It was encouraged (Bengio, 2011) as a way of sharing representations between tasks and is now widely adopted in the community. Supervised knowledge transfer using neural representations can be performed jointly on multiple tasks (Ruder, 2017), by learning linguistic information in a hierarchical fashion

(Søgaard and Goldberg, 2016), and on many levels (Hashimoto et al., 2016). It was also examined between tasks from different modalities including text, images, and audio (Kaiser et al., 2017b).

Concurrently to our work from Chapter 4 (Mihaylov et al., 2017), direct transfer of less related supervised tasks to machine reading comprehension was performed by McCann et al. (2017). They used it as initialization for the DCN (Xiong et al., 2016) and improved the result on SQuAD. By contrast in Chapter 4, we examined a generic and modular **supervised** approach to learning a set of diverse language tasks (referred to as ‘skills’) and analyzed their performance on the task of machine reading comprehension.

**Unsupervised Pre-trained Language Models** While many target tasks benefit from supervised transfer learning, it is usually expensive to create supervised datasets for multiple tasks. Therefore many new works focused on obtaining knowledge from natural language by training unsupervised models on large text corpora.

Initially Dai and Le (2015) proposed unsupervised training on natural language text to improve the stability of training for LSTM models and improve text classification. The LSTM recurrent neural network models were very new to the field and this method remained overlooked. Later Peters et al. (2017) and Liu et al. (2018) revisited the approach and used weights from pre-trained language models to improve sequence labeling tasks. Building on their previous work (Peters et al., 2017), Peters et al. (2018) proposed *ELMo* (*Embeddings from Language Models*) - a bi-directional LSTM language model pre-trained on a large portion of text. *ELMo* was used as a replacement for traditional word embeddings and improved the performance on a wide range of tasks, including machine reading comprehension (Peters et al., 2018). Concurrently to Peters et al. (2018), Howard and Ruder (2018b) successfully adopted a similar approach to Dai and Le (2015) and improved text classification across various datasets. Radford et al. (2018a) pre-trained a Transformer-based uni-directional language model (*GPT1*) on an even bigger amount of natural language text and also proposed a new method of fine-tuning for multi-choice question answering task on RACE (Lai et al., 2017). This was followed by Devlin et al. (2019a)’s *BERT* (Bidirectional Encoder Representations from Transformers) and *RoBERTa* (Liu et al., 2019) that matched the human annotators’ performance on SQuAD and improved the performance of several other MRC benchmarks. In the next chapters, we will not focus on these approaches since they emerged after the work from the last chapter was published (Mihaylov and Frank, 2019).

## 2.5 Summary

Machine Reading Comprehension is a complex task that aims at evaluating text content comprehension by a natural language processing agent. Recent work in the area has focused on using neural approaches to machine reading comprehension, hence we refer to the recent approaches to solve the task as Neural Machine Reading Comprehension. The task has different formulations captured in several large-scale datasets used for the training and evaluation of the systems. Most neural approaches to machine reading comprehension have focused on learning the task solely from the data provided by task- and domain-specific datasets. The interest in the task moved the field forward by producing multiple improvements on common MRC model architectures.

New approaches, including those described in this thesis, have focused instead on using different sources of external knowledge to enrich the representations of common MRC neural architectures. They showed promising directions in improving the MRC models in terms of overall performance and robustness.

# Chapter 3

## Neural Machine Reading Comprehension using External Declarative Knowledge

### 3.1 Motivation

In contrast to many previous models (Weston et al., 2015c; Dhingra et al., 2017b; Cui et al., 2017; Munkhdalai and Yu, 2016; Sordoni et al., 2016) that perform a reading of *only the story and a question* before inferring the correct answer, we aim to tackle the cloze-style RC task by additionally using background knowledge. We develop a neural model for RC that can successfully deal with tasks where some of the information to infer answers from is given in the document (story), but where additional information is needed to predict the answer, which can be retrieved from a knowledge base and added to the context representations explicitly.<sup>1</sup> An illustration is given in Figure 3.1.

In general, such knowledge may be *commonsense knowledge* or *factual background knowledge about entities and events* that is not explicitly expressed in the text but can be found in a knowledge base such as ConceptNet (Speer et al., 2017), BabelNet (Navigli and Ponzetto, 2012), Freebase (Tanon et al., 2016) or domain-specific knowledge bases collected with Information Extraction approaches (Fader et al., 2011; Mausam et al., 2012; Bhutani et al., 2016). Thus, we aim to define a neural model that encodes pre-selected knowledge in memory, and that learns to include the relevant information as an enrichment to the context representation.

The main difference between our model and prior state-of-the-art is that instead of relying only on document-to-question interaction or discrete features while performing multiple hops over the document, our model (i) *attends to relevant selected external knowledge* and (ii)

---

<sup>1</sup> ‘Context representation’ refers to a vector representation computed from textual information only (i.e., document (story) or question).

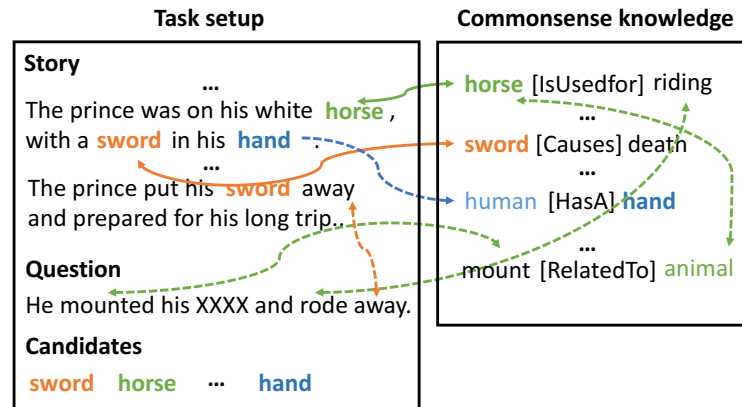


Fig. 3.1 Cloze-style reading comprehension with external commonsense knowledge. **Task setup** shows the official task formulation. **Commonsense knowledge** is external knowledge that is presented to the model. Connecting lines show some examples of desired attention between context and knowledge: solid - a higher attention score, punctuated - a lower attention score.

*combines this knowledge with the context representation before inferring the answer, in a single hop. This allows the model to explicitly imply knowledge that is not stated in the text but is relevant for inferring the answer, and that can be found in an external knowledge source. Moreover, by including knowledge explicitly, our model provides evidence and insight about the used knowledge in the RC.*

## 3.2 Data and Task Descriptions

We experiment with knowledge-enhanced cloze-style reading comprehension on the *Common Nouns* and *Named Entities* partitions of the Children’s Book Test (CBT) dataset (Hill et al., 2016). We also evaluated the proposed approach to multi-choice open book question answering on OpenBookQA (Mihaylov et al., 2018).

### 3.2.1 Cloze-style Reading Comprehension with External Commonsense Knowledge

**Dataset Description** In the CBT cloze-style task (Hill et al., 2016). a system is asked to read a children’s story context of 20 sentences. The following 21<sup>st</sup> sentence involves a placeholder token that the system needs to predict, by choosing from a given set of 10 candidate words from the document. An example with suggested external knowledge facts is given in Figure 3.1. While in its *Common Nouns* setup, the task can be considered as

CBT Statistics		
	Common Nouns (CN)	Named Entities (NE)
<b>Train</b>	120,769 / 470	108,719 / 433
<b>Dev</b>	2,000 / 448	2,000 / 412
<b>Test</b>	2,500 / 461	2,500 / 424
<b>Vocab</b>	53,185	53,063

Table 3.1 Characteristics of Children Book Test datasets. CN: *Common Nouns*, NE: *Named Entities*. Cells for *Train*, *Dev*, *Test* show overall numbers of examples and average story size in tokens.

a language modeling task, Hill et al. (2016) shows that humans can answer the questions without the full context with an accuracy of only 64.4% and a language model alone with 57.7%. By contrast, the human performance when given the full context is at 81.6%. Since the best neural model (Munkhdalai and Yu, 2016) achieves only 72.0% on the task, we hypothesize that the task itself can benefit from external knowledge. The characteristics of the data are shown in Table 3.1.

Other popular cloze-style datasets such as CNN/Daily Mail (Hermann et al., 2015) or WhoDidWhat (Onishi et al., 2016) are mainly focused on finding *Named Entities* where the benefit of adding commonsense knowledge (as we show for the *NE* part of CBT) would be more limited.

**Knowledge Source** As a source of common-sense knowledge we use the *Open Mind Common Sense* part of ConceptNet 5.0 that contains 630k fact triples. We refer to this entire source as *CN5All*. We hypothesize that some of the lexical relations that resemble semantic similarity can be learned in the training so we also conduct experiments with subparts of this data: *CN5WN3* which is the WordNet 3 part of *CN5All* (213k triples) and *CN5Sel*, which excludes the following WordNet relations: *RelatedTo*, *IsA*, *Synonym*, *SimilarTo*, *HasContext*.

### 3.2.2 Open Book Question Answering with External Background and Commonsense Knowledge

**Dataset Description** OpenBookQA is a question answering modeled after open book exams for assessing human understanding of a subject. The open book that comes with our questions is a set of 1326 elementary-level science facts. Roughly 6000 questions probe an understanding of these facts and their application to novel situations. This requires combining an open book fact (e.g., metals conduct electricity) with broad common knowledge (e.g., a suit of armor is made of metal) obtained from other sources. While existing QA datasets



<b>OpenBookQA Statistics</b>	
<b>Total # of questions</b>	5957
<b>Train # of questions</b>	4957
<b>Dev # of questions</b>	500
<b>Test # of questions</b>	500
<b># of choices per question</b>	4
<b>Avg. question sentences</b>	1.08 (6)
<b>Avg. question tokens</b>	11.46 (76)
<b>Avg. choice tokens</b>	2.89 (23)
<b>Avg. science fact tokens</b>	9.38 (28)
<b>Vocabulary size (q+c)</b>	11855
<b>Vocabulary size (q+c+f)</b>	12839
<b>Answer is the longest choice</b>	1108 (18.6%)
<b>Answer is the shortest choice</b>	216 (3.6%)

Table 3.2 Statistics for full OpenBookQA dataset. Parenthetical numbers next to each average are the *max*.

over documents or knowledge bases, being generally self-contained, focus on linguistic understanding, OpenBookQA probes a deeper understanding of both the topic—in the context of common knowledge—and the language it is expressed in. Human performance on OpenBookQA is close to 92%, but many state-of-the-art pre-trained QA methods perform surprisingly poorly, worse than several simple neural baselines we develop. OpenBookQA consists of 5957 questions, with 4957/500/500 in the Train/Dev/Test splits. Table 3.2 summarizes some statistics about the full dataset. Each question has exactly four answer choices and one associated fact used in the creation process. We report the average length of questions, candidate choices, and associated facts, as well as how often is the longest/shortest choice the correct one.

**Knowledge Sources** The dataset comes with an *open book* containing 1326 elementary level science facts. Each question in the dataset is accompanied by an *oracle fact* that bridges it with a fact from the *open book*. The selected *oracle fact* is not intended for use in the evaluation but can be used for computing an upper bound of a knowledge-enhanced model. Instead, for experimentation with knowledge, we consider the ‘open book’ set of facts  $\mathcal{F}$  in conjunction with two sources of common knowledge: the Open Mind Common Sense (Singh et al., 2002) part of ConceptNet (Speer et al., 2017), and its WordNet (Miller, 1995) subset.

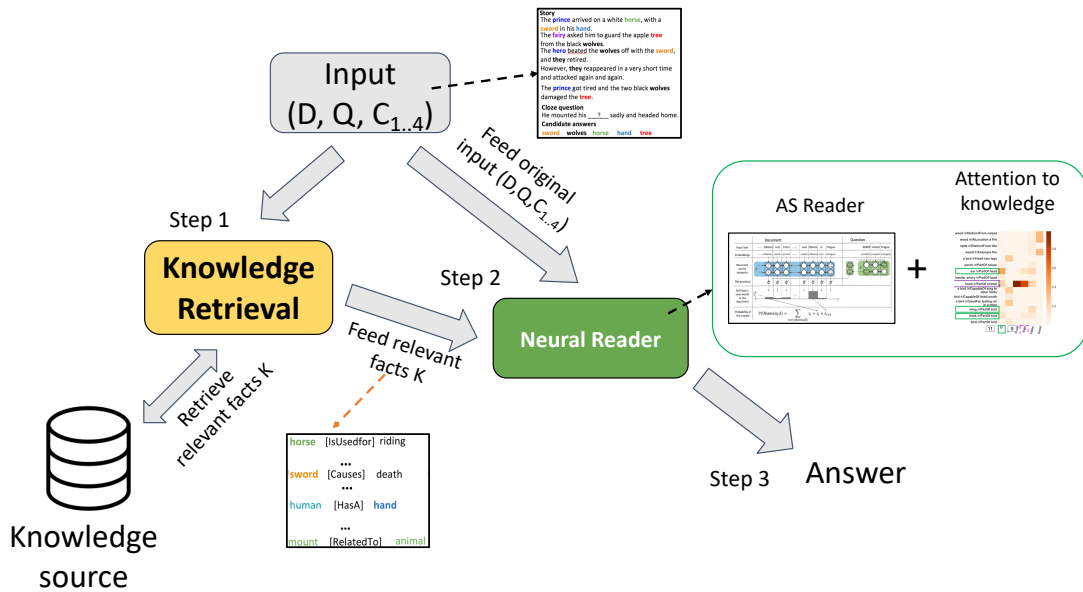


Fig. 3.2 Full architecture of our approach.

### 3.3 Cloze-style Reading Comprehension with Background Knowledge Sources

In this work, we examine the impact of using external knowledge as supporting information for the task of *cloze style* reading comprehension.

We build a system with two modules (Figure 3.2). The first, *Knowledge Retrieval*, performs *fact retrieval* and selects a number of facts  $f_1, \dots, f_p$  that might be relevant for connecting story, question, and candidate answers. The second, main module, *the Knowledgeable Reader*, is a knowledge-enhanced neural module. It uses the input of the story context tokens  $d_{1..m}$ , the question tokens  $q_{1..n}$ , the set of answer candidates  $a_{1..k}$  and a set of ‘relevant’ background knowledge facts  $f_{1..p}$  in order to select the right answer. To include external knowledge for the RC task, we encode each fact  $f_{1..p}$  and use attention to select the most relevant among them for each token in the story and question. We expect that enriching the text with additional knowledge about the mentioned concepts will improve the prediction of correct answers in a strong *single-pass* system. See Figure 3.1 for illustration.

#### 3.3.1 Knowledge Retrieval

In our experiments we use knowledge from the *Open Mind Common Sense (OMCS)* (Singh et al. (2002)) part of ConceptNet, a crowd-sourced resource of commonsense knowledge with

a total of  $\sim 630k$  facts. Each fact  $f_i$  is represented as a triple  $f_i=(subject, relation, object)$ , where *subject* and *object* can be multi-word expressions and *relation* is a relation type. An example is: (*[bow]*<sub>subj</sub>, *[IsUsedFor]*<sub>rel</sub>, *[hunt, animals]*<sub>obj</sub>)

We experiment with three set-ups: using (i) all facts from OMCS that pertain to ConceptNet, referred to as *CN5All*, (ii) using all facts from *CN5All* excluding some WordNet relations referred to as *CN5Selected* (see Section 3.2.1), and using (iii) facts from OMCS that have *source* set to *WordNet* (*CN5WN3*).

To address our cloze-style setup where our answer is a single token, we employ a heuristic approach for selecting relevant facts from our knowledge source. For each instance ( $D, Q, A_{1..10}$ ) we retrieve relevant commonsense background facts. We first retrieve facts that contain lemmas that can be looked up via tokens contained in any  $D$ (ocument),  $Q$ (uestion) or  $A$ (nswer candidates). We add a weight value for each node: 4, if it contains a lemma of a candidate token from  $A$ ; 3, if it contains a lemma from the tokens of  $Q$ ; and 2 if it contains a lemma from the tokens of  $D$ . The selected weights are chosen heuristically such that they model relative fact importance in different interactions as  $A+A > A+Q > A+D > D+Q > D+D$ . We weight the fact triples that contain these lemmas as nodes, by summing the weights of the subject and object arguments. Next, we sort the knowledge triples by this overall weight value. To limit the memory of our model, we run experiments with different sizes of the top number of facts ( $P$ ) selected from all instance fact candidates,  $P \in \{50, 100, 200\}$ . As an additional retrieval limitation, we force the number of facts per answer candidate to be the same, to avoid a frequency bias for an answer candidate that appears more often in the knowledge source. Thus, if we select the maximum 100 facts for each task instance and we have 10 answer candidates  $a_{i=1..10}$ , we retrieve the top 10 facts for each candidate  $a_i$  that has either a subject or an object lemma for a token in  $a_i$ . If the same fact contains lemmas of two candidates  $a_i$  and  $a_j$  ( $j > i$ ), we add the fact once for  $a_i$  and do not add the same fact again for  $a_j$ . If several facts have the same weight, we take the first in the order of the list<sup>2</sup>, i.e., the order of retrieval from the database. If one candidate has less than 10 facts, the overall fact candidates for the sample will be less than the maximum (100).

### 3.3.2 Knowledgeable Reader: Neural Reader with Explicit Knowledge Memory

We implement our *Knowledgeable Reader* (*KnReader*) using as a basis the *Attention Sum Reader* (Kadlec et al., 2016) as one of the strongest core models for single-hop RC. We

<sup>2</sup>We also experimented with re-ranking the facts with the same weight sums using tf-idf but we did not notice a difference in performance.

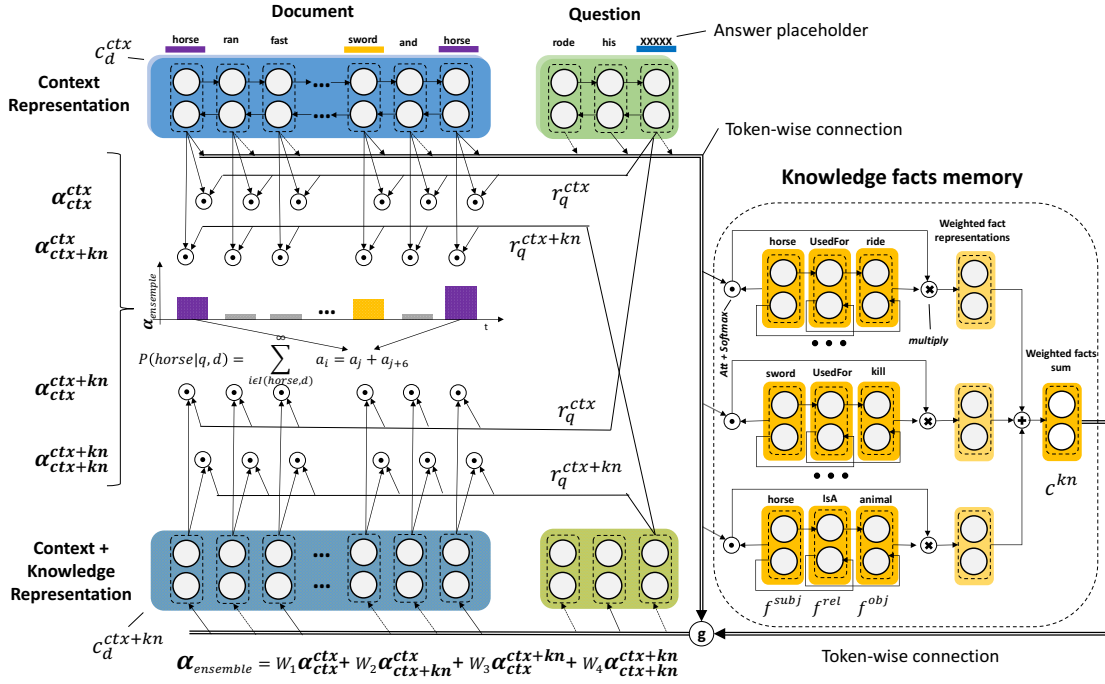


Fig. 3.3 The Knowledgeable Reader combines plain *context* & *enhanced (context + knowledge)* representations of  $D$  and  $Q$  and retrieved knowledge from the explicit memory with the *Key-Value* approach.

extend it with a knowledge fact memory that is filled with pre-selected facts. Our aim is to examine how adding commonsense knowledge to a simple yet effective model can improve the RC process and to show some evidence of that by attending to the incorporated knowledge facts. The model architecture is shown in Figure 3.3.

**Base Model** Our model for RC is based on the *Attention-Sum Reader* (Kadlec et al., 2016). It reads the input of story tokens  $d_{1..n}$ , the question tokens  $q_{1..m}$ , and the set of candidates  $a_{1..10}$  that occur in the story text. The model calculates the attention between the question representation  $r_q$  and the story token context encodings of the candidate tokens  $a_{1..10}$  and sums the attention scores for the candidates that appear multiple times in the story. The model selects as answer the candidate that has the highest attention score.

**Word Embeddings Layer** We represent input document and question tokens  $w$  by looking up their embedding representations  $e_i = Emb(w_i)$ , where  $Emb$  is an embedding lookup function. We apply dropout (Srivastava et al., 2014) with keep probability  $p = 0.8$  to the output of the embeddings lookup layer.

**Context Representations** To represent the document and question contexts, we first encode the tokens with a Bi-directional GRU (Gated Recurrent Unit) (Chung et al., 2014) to obtain context-encoded representations for document ( $c_{d_{1..n}}^{ctx}$ ) and question ( $c_{q_{1..m}}^{ctx}$ ) encoding:

$$c_{d_{1..n}}^{ctx} = BiGRU^{ctx}(e_{d_{1..n}}) \in \mathbb{R}^{n \times 2h} \quad (3.1)$$

$$c_{q_{1..m}}^{ctx} = BiGRU^{ctx}(e_{q_{1..m}}) \in \mathbb{R}^{m \times 2h}, \quad (3.2)$$

where  $d_i$  and  $q_i$  denote the  $i$ th token of a text sequence  $d$  (document) and  $q$  (question), respectively,  $n$  and  $m$  is the size of  $d$  and  $q$  and  $h$  the output hidden size (256) of a single GRU unit.  $BiGRU$  is defined in (3.3), with  $e_i$  a word embedding vector

$$BiGRU^{ctx}(e_i, h_{i_{prev}}) = \begin{bmatrix} \overrightarrow{GRU}(e_i, \overrightarrow{h_{i_{prev}}}), \\ \overleftarrow{GRU}(e_i, \overleftarrow{h_{i_{prev}}}) \end{bmatrix}, \quad (3.3)$$

where  $h_{i_{prev}} = [\overrightarrow{h_{i_{prev}}}, \overleftarrow{h_{i_{prev}}}]$ , and  $\overrightarrow{h_{i_{prev}}}$  and  $\overleftarrow{h_{i_{prev}}}$  are the previous hidden states of the forward and backward layers. Below we use  $BiGRU^{ctx}(e_i)$  without the hidden state, for short.

**Question Query Representation** For the question we construct a single vector representation  $r_q^{ctx}$  by retrieving the token representation at the placeholder (XXXX) index  $pl$  (cf. Figure 3.3):

$$r_q^{ctx} = c_{q_{1..m}}^{ctx}[pl] \in \mathbb{R}^{2h}, \quad (3.4)$$

where  $[pl]$  is an element pickup operation.

Our question vector representation is different from the original *AS Reader* that builds the question by concatenating the *last states* of a forward and backward layer  $[\overrightarrow{GRU}(e_m), \overleftarrow{GRU}(e_1)]$ . We changed the original representation as we observed some very long questions and in this way aim to prevent the context encoder from 'forgetting' where the placeholder is.

**Answer Prediction:  $Q^{ctx}$  to  $D^{ctx}$  Attention** In order to predict the correct answer to the given question, we rank the given answer candidates  $a_1..a_L$  according to the normalized attention sum score between the context ( $ctx$ ) representation of the question placeholder  $r_q^{ctx}$

and the representation of the candidate tokens in the document:

$$P(a_i|q, d) = \text{softmax}(\sum \alpha_{i_j}) \quad (3.5)$$

$$\alpha_{i_j} = \text{Att}(r_q^{ctx}, c_{d_j}^{ctx}), i \in [1..L], \quad (3.6)$$

where  $j$  is an index pointer from the list of indices that point to the candidate  $a_i$  token occurrences in the document context representation  $c_d$ .  $\text{Att}$  is a dot product.

**Enriching Context Representations with Knowledge (Context+Knowledge)** To enhance the representation of the context, we add knowledge, retrieved as a set of knowledge facts.

**Knowledge Encoding** For each instance in the dataset, we retrieve a number of relevant facts (cf. Section 3.3.1). Each retrieved fact is represented as a triple  $f = (w_{1..L_{subj}}^{subj}, w_0^{rel}, w_{1..L_{obj}}^{obj})$ , where  $w_{1..L_{subj}}^{subj}$  and  $w_{1..L_{obj}}^{obj}$  are a multi-word expressions representing the *subject* and *object* with sequence lengths  $L_{subj}$  and  $L_{obj}$ , and  $w_0^{rel}$  is a word token corresponding to a relation.<sup>3</sup> As a result of fact encoding, we obtain a separate knowledge memory for each instance in the data.

To encode the knowledge we use the same *BiGRU* weights used for the context and question to encode the triple argument tokens into the following context-encoded representations:

$$f_{last}^{subj} = \text{BiGRU}(\text{Emb}(w_{1..L_{subj}}^{subj}), 0) \quad (3.7)$$

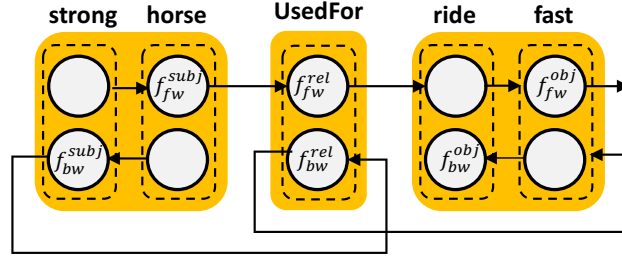
$$f_{last}^{rel} = \text{BiGRU}(\text{Emb}(w_0^{rel}), f_{last}^{subj}) \quad (3.8)$$

$$f_{last}^{obj} = \text{BiGRU}(\text{Emb}(w_{1..L_{obj}}^{obj}), f_{last}^{rel}), \quad (3.9)$$

where  $f_{last}^{subj}$ ,  $f_{last}^{rel}$ ,  $f_{last}^{obj}$  are the final hidden states of the context encoder *BiGRU*, that are also used as initial representations for the encoding of the next triple attribute in left-to-right order. Comprehensive visualization of the facts encoding and the unrolled token states is shown on Figure 3.4.

The motivation behind this encoding is: (i) using the same *BiGRU* weights used for context encoding, we encode the knowledge fact attributes in the same vector space as the plain tokens; (ii) we preserve the triple’s directionality; (iii) we use the relation type as a way of filtering the *subject* information to initialize the *object*.

<sup>3</sup>The 0 in  $w_0^{rel}$  indicates that we encode the relation as a single *relation type* word. Ex. */r/IsUsedFor*.



$$\begin{aligned}
 f^{subj} &= [GRU^{fw}([strong, horse]), GRU^{bw}([horse, strong])] \\
 f^{rel} &= [GRU^{fw}([strong, horse, UsedFor]), GRU^{bw}([horse, strong, UsedFor])] \\
 f^{obj} &= [GRU^{fw}([strong, horse, UsedFor, ride, fast]), GRU^{bw}([horse, strong, UsedFor, fast, ride])]
 \end{aligned}$$

Fig. 3.4 Encoding the knowledge triple using BiGRU.

**Querying the Knowledge Memory** To enrich the context representation of the document and question tokens with the facts collected in the knowledge memory, we select a single *sum of weighted fact representations* for each token using Key-Value retrieval (Miller et al., 2016). In our model the *key*  $M_i^{k(ey)}$  can be either  $f_{last}^{subj}$  or  $f_{last}^{obj}$  and the *value*  $M_i^{v(alue)}$  is  $f_{last}^{obj}$ .

For each context-encoded token  $c_{s_i}^{ctx}$  ( $s = d, q; i$  the token index) we attend to all knowledge memory keys  $M_i^k$  in the retrieved  $P$  knowledge facts. We use an attention function  $Att$ , scale the scalar attention value using  $softmax$ , multiply it with the value representation  $M_i^v$  and sum the result into a single vector value representation  $c_{s_i}^{kn}$ :

$$c_{s_i}^{kn} = \sum softmax(Att(c_{s_i}^{ctx}, M_{1..P}^k))^T M_{1..P}^v \quad (3.10)$$

$Att$  is a dot product, but it can be replaced with another attention function. As a result of this operation, the context token representation  $c_{s_i}^{ctx}$  and the corresponding retrieved knowledge  $c_{s_i}^{kn}$  are in the same vector space  $\in \mathbb{R}^{2h}$ .

**Combine Context and Knowledge** ( $ctx + kn$ ) We combine the original context token representation  $c_{s_i}^{ctx}$ , with the acquired knowledge representation  $c_{s_i}^{kn}$  to obtain  $c_{s_i}^{ctx+kn}$ :

$$c_{s_i}^{ctx+kn} = \gamma c_{s_i}^{ctx} + (1 - \gamma) c_{s_i}^{kn}, \quad (3.11)$$

where  $\gamma = 0.5$ . We keep  $\gamma$  static but it can be replaced with a gating function.

**Answer Prediction:**  $Q^{ctx(+kn)}$  to  $D^{ctx(+kn)}$  To rank answer candidates  $a_1..a_L$  we use attention sum similar to Eq.3.5 over an attention  $\alpha_{i_j}^{ensemble}$  that combines attentions between

context ( $ctx$ ) and context+knowledge ( $ctx + kn$ ) representations of the question ( $r_q^{ctx(+kn)}$ ) and candidate token occurrences  $a_{i_j}$  in the document  $c_{d_j}^{ctx(+kn)}$ :

$$P(a_i|q, d) = softmax(\sum \alpha_{i_j}^{ensemble}) \quad (3.12)$$

$$\begin{aligned} \alpha_{i_j}^{ensemble} = & W_1 Att(r_q^{ctx}, c_{d_j}^{ctx}) \\ & + W_2 Att(r_q^{ctx}, c_{d_j}^{ctx+kn}) \\ & + W_3 Att(r_q^{ctx+kn}, c_{d_j}^{ctx}) \\ & + W_4 Att(r_q^{ctx+kn}, c_{d_j}^{ctx+kn}), \end{aligned} \quad (3.13)$$

where  $j$  is an index pointer from the list of indices that point to the candidate  $a_i$  token occurrences in the document context representation  $c_d^{ctx(+kn)}$ .  $W_{1..4}$  are scalar weights initialized with 1.0 and optimized during training.<sup>4</sup> We propose the combination of  $ctx$  and  $ctx + kn$  attentions because our task does not provide supervision on whether the knowledge is needed. Having the combinations of knowledge and context interaction, we can examine the attention values at inference and validate if the knowledge was needed to answer the question correctly.

### 3.3.3 Technical Details

We implement our model in *TensorFlow 0.12* (Abadi et al., 2015). Below we report pre-processing steps and hyper-parameters required for reproducing the model.

**Dataset** We perform experiments on the *Common Nouns* and *Named Entities* parts of the Children’s Book Test (CBT) (Hill et al., 2016).<sup>5</sup>

**Pre-processing** For each instance of the dataset (21 sentences, 20 for the story and 1 for question), we remove the line number, which is originally presented in the text as a first token of the sentence and split the tokens using *str.split()* in *Python 2.7*. We then concatenate the tokens for the sentences in the story into a single list of story tokens  $d_{1..m}$ .

**Knowledge Source** We use knowledge from the Open Mind Common Sense (OMCS, Singh et al. (2002)) part of ConceptNet (Speer et al., 2017), a crowd-sourced resource of commonsense knowledge with a total of  $\sim 630k$  facts.<sup>6</sup>

<sup>4</sup>An example for learned  $W_{1..4}$  is (1.84, 1.41, 2.13, 1.49) in setting (CBT CN, CN5Sel, Subj-Obj as k-v, 50 facts).

<sup>5</sup>The dataset can be downloaded from: <http://www.thespermwhale.com/jaseweston/babi/CBTest.tgz>

<sup>6</sup>ConceptNet 5 github page: <https://github.com/commonsense/conceptnet5>.



**Vocabulary** To build the vocabulary we select the words that occur at least 5 times in the training set. We extend the vocabulary with all words retrieved from the knowledge source. All words are lowercased. Following Kadlec et al. (2016) we use multiple unknown tokens ( $UNK_1, UNK_2, \dots, UNK_{100}$ ). In each example, for each unknown word, we pick randomly an unknown token from the list and use it for all occurrences of the word in the document (story) and question.

**Word Embeddings** We use Glove 100D<sup>7</sup> word embeddings pre-trained on 6B tokens from Wikipedia and Gigaword5. We initialize the out-of-vocabulary words by sampling from a uniform distribution in range  $[-0.1, 0.1]$ . We optimize all word embeddings in the first 8000 training steps.

**Encoder Hidden Size.** We use a hidden size of 256 for the *GRU* encoder states (512 output for our bi-directional encoding). This setting has been shown to perform well for the Attention Sum Reader (Kadlec et al., 2016).

**Batching, Learning rate, Sampling** We sort the data examples in the training set by document length and create batches with 64 examples. For each training step we pick batches randomly. After every 1000 training steps we evaluate the models on the validation *Dev* set. We train for 60 epochs and pick the model with the highest validation accuracy to make the predictions for *Test*.

**Optimization** We use cross entropy loss on the predicted scores for each answer candidate. We use Adam (Kingma and Ba, 2015) optimizer with initial learning rate of *0.001* and clip the gradients in the range  $[-10, 10]$ .

### 3.4 Open Book Question Answering with External Knowledge Sources

In addition to reading comprehension, which is the main focus of this thesis, we evaluate our *Knowledgeable Reader* model to open book question answering on OpenBookQA (Mihaylov et al., 2018). OpenBookQA is a multi-choice dataset that requires answering scientific questions given scientific knowledge (open book) and requiring external common or commonsense knowledge. An example is shown in Figure 3.5.

---

<sup>7</sup>The embeddings can be downloaded from: <http://nlp.stanford.edu/data/glove.6B.zip>

**Question:**  
Which of these would let the most heat travel through?

A) a new pair of jeans.  
B) a steel spoon in a cafeteria.  
C) a cotton candy at a store.  
D) a calvin klein cotton hat.

**Science Fact:**  
Metal is a thermal conductor.

**Common Knowledge:**  
Steel is made of metal.  
Heat travels through a thermal conductor.

Fig. 3.5 An example for a question with a given set of choices and supporting facts.

To tackle this dataset we implement a two-stage model for incorporating external common knowledge,  $K$ . The first module performs information retrieval on  $K$  to select a fixed size subset of potentially relevant facts  $K_{Q,C}$  for each instance in the dataset. The second module is a neural network that takes  $(Q, C, K_{Q,C})$  as input to predict the answer  $a_{q,c}$  to a question  $Q$  from the set of choices  $C$ .

### 3.4.1 Knowledge Retrieval

This module is the first part of a two-stage approach for incorporating knowledge from an external source  $K$ .

For the information retrieval, we use TfidfVectorizer<sup>8</sup> to build vector representations  $\vec{q}_{\text{tfidf}}$ ,  $\vec{c}_{\text{tfidf}}^i$  and  $\vec{k}_{\text{tfidf}}$  for the question  $q$ , choice  $c_i \in C$ , and fact  $k \in K$  based on the tokens in the training set. We then calculate similarity scores  $t_{q,k}$  and  $t_{q,c_i,k}$  between  $q$  and  $c_i$ , resp., and each of the external facts in  $k \in K$ :

$$t_{q,k} = 1 - \text{sim}(\vec{q}_{\text{tfidf}}, \vec{k}_{\text{tfidf}})$$

$$t_{q,c_i,k} = 1 - \text{sim}(\vec{c}_{\text{tfidf}}^i, \vec{k}_{\text{tfidf}}) \cdot t_{q,k},$$

where  $\text{sim}$  is implemented as cosine distance. Based on these similarity scores, we obtain a set  $K_{q,C}$  of facts for each  $(q, C, K)$  as  $K_q \cup \bigcup_i K_{q,c_i}$ , where  $K_q$  and  $K_{q,c_i}$  are the top  $N_k$  facts each with highest similarity  $t_{q,k}$  and  $t_{q,c_i,k}$ , respectively.  $N_k$  is a hyper-parameter chosen from  $\{5, 10, 20\}$  so as to yield the best Dev set performance.

<sup>8</sup>Term frequency, Inverse document frequency based vectorizer from *scikit-learn* (Pedregosa et al., 2011).

For experimentation with knowledge, we consider the ‘open book’ set of facts  $\mathcal{F}$  in conjunction with two sources of common knowledge: the Open Mind Common Sense (Singh et al., 2002) part of ConceptNet (Speer et al., 2017), and its WordNet (Miller, 1995) subset.

### 3.4.2 Knowledgeable Reader for Multi-Choice Question Answering

**Base Model: BiLSTM Max-Out** As a base neural model, we adapt *BiLSTM max-out* model (Conneau et al., 2017a) to our QA task. That is, we first encode the question tokens and choice tokens  $w_{1..n_s}^s$ , independently with a bi-directional context encoder (LSTM) to obtain a context (ctx) representation  $h_{s_{1..n_s}}^{\text{ctx}} = \text{BiLSTM}(e_{1..n_s}^s) \in \mathbb{R}^{n_s \times 2h}$ . Next, we perform an element-wise aggregation operation  $\max$  on the encoded representations  $h_{s_{1..n_s}}^{\text{ctx}}$  to construct a single vector:

$$r_s^{\text{ctx}} = \max(h_{s_{1..n_s}}^{\text{ctx}}) \in \mathbb{R}^{2h}. \quad (3.14)$$

Given the contextual representations for each token sequence, we experiment with three configurations for using these representations for QA:

**(a) Plausible Answer Detector.** This baseline goes to the extreme of completely ignoring  $q$  and trying to learn how plausible it is for  $c_i$  to be the correct answer to *some* question in this domain. This captures the fact that certain choices like ‘a magical place’ or ‘flying cats’ are highly unlikely to be the correct answer to a science question without negation (which is the case for OpenBookQA).

We implement a plausible answer detector using a *choice-only* model for predicting the answer by obtaining a score  $\alpha_{c_i}$  as:  $\alpha_{c_i} = W_c^T r_{c_i}^{\text{ctx}} \in \mathbb{R}^1$ , where  $W_c^T \in \mathbb{R}^{2h}$  is a weights vector optimized during training,  $i = \{1..4\}$  is the index of the choice. To obtain the answer choice from the set of choice scores  $\alpha_{c_{1..4}}$  using  $\arg \max(\text{softmax}(\alpha_{c_{1..4}}))$ , where  $\text{softmax}(\alpha_{c_i}) = \frac{\exp(\alpha_{c_i})}{\sum_{j=1}^4 \exp(\alpha_{c_j})}$  as usual.

This model tries to predict which choice best matches the question (Nakov et al., 2016b), without relying on external knowledge. To achieve that, we compute an attention score  $\alpha_{q,c_i}$  between  $q$  and each of the choices  $q_i$  as  $\alpha_{q,c_i} = \text{Att}(r_q^{\text{ctx}}, r_{c_i}^{\text{ctx}})$ , and select the one with the highest score. We also experiment with a model where  $r_q^{\text{ctx}}$  and  $r_{c_i}^{\text{ctx}}$  are obtained using token-wise interaction proposed in ESIM (Chen et al., 2017).

**Knowledgeable Reader for Multi-choice Question Answering** As a base knowledge-aware model, we use a variant of the model of Mihaylov and Frank (2018), implemented

by extending our **BiLSTM max-out**. For each instance the model reads the question  $q$  and answers  $c_{1..4}$  independently and attends to the set of retrieved external knowledge facts  $K_{Q,C}$ . We encode each fact  $k_j$  from  $K_{Q,C} = k_{1..N_k}$  ( $N_k$  is the number of facts) with the same BiLSTM as used for  $q$  and  $c_{1..4}$  and construct a single vector  $r_{k_j}^{\text{ctx}} \in \mathbb{R}^{2h}$  using Eq. 3.14. Having such representations for each  $k_j$  results in knowledge memory matrix  $M_k = r_{k_{1..N_k}}^{\text{ctx}} \in \mathbb{R}^{N_k \times 2h}$ . Note that  $M_k$  is dynamic memory, specific for each instance in the batch, and is encoded in each step during training. This memory is used to calculate a knowledge-aware representation,  $r_s^{\text{kn}} = \sum((M_k^T r_s^{\text{ctx}}) \cdot M_k) \in \mathbb{R}^{2h}$ . Each context (ctx) representation  $r_s^{\text{ctx}}$  ( $s \in \mathcal{S}$ ) is combined with  $r_s^{\text{kn}}$  to obtain a knowledge-enhanced representation  $r_s^{\text{ctx+kn}} = (r_s^{\text{ctx}} + r_s^{\text{kn}})/2$ . We then model the knowledge-enhanced attention  $\alpha_{q,c_i}^{\text{kn}}$  between  $q$  and  $c_i$  as a linear combination of the ctx, kn and ctx + kn representations as

$$\begin{aligned} \alpha_{q,c_i} = W^T & [\text{Att}(r_s^{\text{ctx}}, r_{c_i}^{\text{ctx}}); \text{Att}(r_s^{\text{kn}}, r_{c_i}^{\text{kn}}); \\ & \text{Att}(r_s^{\text{ctx+kn}}, r_{c_i}^{\text{ctx}}); \text{Att}(r_s^{\text{ctx}}, r_{c_i}^{\text{ctx+kn}}); \\ & \text{Att}(r_s^{\text{ctx}}, r_{c_i}^{\text{kn}}); \text{Att}(r_s^{\text{kn}}, r_{c_i}^{\text{ctx}}); \\ & \text{Att}(r_s^{\text{ctx+kn}}, r_{c_i}^{\text{kn}}); \text{Att}(r_s^{\text{kn}}, r_{c_i}^{\text{ctx+kn}}); \\ & \text{Att}(r_s^{\text{ctx+kn}}, r_{c_i}^{\text{ctx+kn}})], \end{aligned}$$

where  $W \in \mathbb{R}^9$  is a weight vector initialized with the *ones* vector and optimized during training. We then select the answer  $c_i$  with the highest score.

### 3.4.3 Baseline Models

We evaluate the performance of several baseline systems on the Dev and Test subsets of OpenBookQA. For each question, a solver receives 1 point towards this score if it chooses the correct answer, and  $1/k$  if it reports a  $k$ -way tie that includes the correct answer. The ‘‘Guess All’’ baseline, which always outputs a 4-way tie, thus achieves a score of 25%, the same as the expected performance of a uniform random baseline.

#### 3.4.3.1 No Training, External Knowledge Only

Since OpenBookQA is a set of elementary-level science questions, one natural baseline category is existing systems that have proven to be effective on elementary- and middle-school level science exams. These pre-trained systems, however, rely only on their background knowledge and do not take the set  $\mathcal{F}$  of core facts into account. Further, their knowledge sources and retrieval mechanism are close to those used by the IR solver that, by design, is guaranteed to fail on OpenBookQA. These two aspects place a natural limit on the

effectiveness of these solvers on OpenBookQA, despite their excellent fit for the domain of multiple-choice science questions. We consider four such solvers.

**PMI** (Clark et al., 2016) uses pointwise mutual information (PMI) to score each answer choice using statistics based on a corpus of 280 GB of plain text. It extracts unigrams, bigrams, trigrams, and skip-bigrams from the question  $q$  and each answer choice  $c_i$ . Each answer choice is scored based on the average PMI across all pairs of question and answer n-grams.

**TableILP** (Khashabi et al., 2016) is an Integer Linear Programming (ILP) based reasoning system designed for science questions. It operates over semi-structured relational tables of knowledge. It scores each answer choice based on the optimal (as defined by the ILP objective) “support graph” connecting the question to that answer through table rows. The small set of these knowledge tables, however, often results in missing knowledge, making TableILP not answer 24% of the OpenBookQA questions at all.

**TupleInference** (Khot et al., 2017a), also an ILP-based QA system, uses Open IE tuples (Banko et al., 2007) as its semi-structured representation. It builds these subject-verb-object tuples *on-the-fly* by retrieving text for each question from a large corpus. It then defines an ILP program to combine evidence from multiple tuples.

**DGEM** (Khot et al., 2018) is a neural entailment model that also uses Open IE to produce a semi-structured representation. We use the adaptation of this model to multiple-choice question answering proposed by Clark et al. (2018), which works as follows: (1) convert  $q$  and each  $c_i$  into a hypothesis,  $h_i$ , and each retrieved fact into a premise  $p_j$ ; and (2) return the answer choice with the highest entailment score,  $\arg \max_i e(p_j, h_i)$ .

### 3.4.3.2 No Training; $\mathcal{F}$ and External Knowledge

We also consider providing the set  $\mathcal{F}$  of core facts to two existing solvers: the **IR** solver of Clark et al. (2016) (to assess how far simple word-overlap can get), and the **TupleInference** solver.

### 3.4.3.3 Trained Models, No Knowledge

We consider several neural baseline models that are trained using Train set of OpenBookQA. For ease of explanation, we first define the notation used in our models. For a given question  $q_{mc} = (q, \{c_1, c_2, c_3, c_4\})$ , we define the set of token sequences,  $\mathcal{S} = \{q, c_1, c_2, c_3, c_4\}$ . For each token sequence  $s \in \mathcal{S}$ ,  $w_j^s$  is the  $j^{th}$  and  $e_j^s = \text{Emb}(w_j^s)$  is the embedding for this token. We use  $n_s$  to indicate the number of tokens in  $s$  and  $d$  for the dimensionality of the

embeddings.<sup>9</sup> We model multiple-choice QA as multi-class classification: Given  $q_{mc}$ , predict one of four class labels  $L = \{1, 2, 3, 4\}$ , where the true label is the correct answer index.

**Embeddings + Similarities as Features.** We first experiment with a simple logistic regression model (Mihaylov and Nakov, 2016; Mihaylov and Frank, 2016a, 2017b) that uses centroid vectors  $r_s^{\text{emb}}$  of the word embeddings of tokens in  $s$ , and then computes the cosine similarities between the question and each answer choice,  $r_{q,c_i}^{\text{cos}}$ :

$$r_s^{\text{emb}} = \frac{1}{n_s} \sum_{j=1}^{n_s} e_{s_j} \in \mathbb{R}^d$$

$$r_{q,c_i}^{\text{cos}} = \cos(r_q^{\text{emb}}, r_{c_i}^{\text{emb}}) \in \mathbb{R}^1$$

For each training instance, we build a feature representation  $\vec{f}$  by concatenating these vectors and train an  $L2$  logistic regression classifier:

$$\vec{f} = [r_q^{\text{emb}}; r_{c_{1..4}}^{\text{emb}}; r_{q,c_{1..4}}^{\text{cos}}] \in \mathbb{R}^{5d+4}$$

### 3.4.4 Technical Details

Our neural models are implemented with *AllenNLP*<sup>10</sup> (Gardner et al., 2017) and *PyTorch*<sup>11</sup> (Paszke et al., 2017).

We use *cross-entropy* loss and the *Adam* optimizer (Kingma and Ba, 2015) with an initial learning rate of 0.001.

**Training** For the neural models *without* external knowledge, we typically train the model with a maximum of 30 epochs and stop training early if the Dev set accuracy does not improve for 10 consecutive epochs. We also halve the learning rate if there is no Dev set improvement for 5 epochs. For the neural models *with* external knowledge, we typically train for 60 epochs with patience of 20 epochs.

**Encoder Hidden Size** For most of our neural models, we use  $h = 128$  as the *LSTM* hidden layer size.

<sup>9</sup>For all experiments we use  $d = 300$  *GloVe* (Pennington et al., 2014) embeddings pre-trained on 840B tokens from *Common Crawl* (<https://nlp.stanford.edu/projects/glove/>).

<sup>10</sup><https://allennlp.org>

<sup>11</sup><https://pytorch.org>

**Embeddings** We use Glove 100D word embeddings. The embedding dropout rate is chosen from  $\{0.1, 0.2, 0.5\}$ , again based on the best Dev set performance.

**Reporting Results** For each model configuration, we perform 5 experiments with different random seeds. For each run, we take the model with the best performance on Dev and evaluate on Test. We report the average accuracy for the best Dev score and the average of the corresponding Test score  $\pm$  the standard deviation across the 5 random seeds.

The code for the models and the configuration files required for reproducing the results are available at <http://data.allenai.org/OpenBookQA>.

## 3.5 Experiments and Results

We perform quantitative analysis through experiments. We study the impact of the used knowledge and different model components that employ external knowledge for cloze-style reading comprehension and open book question answering.

### 3.5.1 Cloze-style Reading Comprehension

We perform experiments on knowledge-enhanced cloze-style question answering on the CBTest dataset. Some of the experiments below focus only on the *Common Nouns (CN)* dataset, as it has been shown to be more challenging than *Named Entities (NE)* in prior work.

#### 3.5.1.1 Model Parameters

We experiment with different model parameters.

**Number of facts.** We explore different sizes of knowledge memories, in terms of the number of acquired facts. If not stated otherwise, we use 50 facts per example.

**Key-Value Selection Strategy.** We use two strategies for defining key and value (Key/Value): *Subj/Obj* and *Obj/Obj*, where *Subj* and *Obj* are the subject and object attributes in the fact triples and they are selected as *Key* and *Value* for the KV memory (see Section 3.3.2, *Querying the Knowledge Memory*). If not stated otherwise, we use the *Subj/Obj* strategy.

**Answer Selection Components.** If not stated otherwise, we use ensemble attention  $\alpha_{ensemble}$  (combinations of *ctx* and *ctx+kn*) to rank the answers. We call this our *Full model* (see Sec. 3.3.2).

Source	Dev	Test
CN5All	71.40	66.72
CN5WN3 (WN3)	70.70	68.48
CN5Sel(ected)	<b>71.85</b>	67.64

Table 3.3 Results with different knowledge sources, for CBT-CN (Full model, 50 facts).

# facts	50	100	200	500
Dev	<b>71.85</b>	71.35	71.40	71.20
Test	67.64	67.44	<b>68.12</b>	67.24

Table 3.4 Results for CBT (CN) with different numbers of facts. (Full model, CN5Sel)

**Hyper-parameters.** For our experiments we use pre-trained Glove (Pennington et al., 2014) embeddings, *BiGRU* with hidden size 256, batch size of 64, and learning rate of 0.001 as they were shown (Kadlec et al., 2016) to perform good on the AS Reader.

### 3.5.1.2 Empirical Results

We perform experiments with the different model parameters described above. We report accuracy on the *Dev* and *Test* and use the results on *Dev* set for pruning the experiments.

**Knowledge Sources.** We experiment with different configurations of *ConceptNet* facts (see Section 3.2.1). Results on the *CBT CN* dataset are shown in Table 3.3. *CN5Sel* works best on the *Dev* set but *CN5WN3* works much better on *Test*. Further experiments use the *CN5Sel* setup.

**Number of facts.** We further experiment with different numbers of facts on the *Common Nouns* dataset (Table 3.4). The best result on the *Dev* set is for 50 facts so we use it for further experiments.

**Component ablations.** We ensemble the attention from different combinations of the interaction between the question and document *context* (*ctx*) representations and *context+knowledge* (*ctx+kn*) representations in order to infer the right answer (see Section 3.3.2, Answer Ranking).

Table 3.5 shows that the combination of different interactions between *ctx* and *ctx+kn* representations leads to clear improvement over the *w/o knowledge* setup, in particular for the *Common Nouns* dataset.



$D_{repr}$ to $Q_{repr}$ interaction	Named Entities (NE)		Common Nouns (CN)	
	Dev	Test	Dev	Test
$D_{ctx}, Q_{ctx}$ (w/o know)	75.50	70.30	68.20	64.80
$D_{ctx+kn}, Q_{ctx+kn}$	76.45	69.68	70.85	66.32
$D_{ctx}, Q_{ctx+kn}$	<b>77.10</b>	69.72	70.80	66.32
$D_{ctx+kn}, Q_{ctx}$	75.65	<b>70.88</b>	71.20	<b>67.96</b>
Full model	76.80	70.24	<b>71.85</b>	67.64
w/o $D_{ctx}, Q_{ctx}$	75.95	70.24	70.65	67.12
w/o $D_{ctx+kn}, Q_{ctx+kn}$	76.20	69.80	70.75	67.00
w/o $D_{ctx}, Q_{ctx+kn}$	76.55	70.52	71.75	66.32
w/o $D_{ctx+kn}, Q_{ctx}$	76.05	70.84	70.80	66.80

Table 3.5 Results for different combinations of interactions between document (D) and question (Q) *context* ( $ctx$ ) and *context + knowledge* ( $ctx+kn$ ) representations. (CN5Sel, 50 facts)

Key/Value	Named Entities (NE)		Common Nouns (CN)	
	Dev	Test	Dev	Test
Subj/Obj	76.65	71.52	71.85	67.64
Obj/Obj	76.70	71.28	71.25	67.48

Table 3.6 Results for key-value knowledge retrieval and integration. (CN5Sel, 50 facts). *Subj/Obj* means: we attend over the fact subject (Key) and take the weighted fact object as value (Value).

**Key-Value Selection Strategy.** Table 3.6 shows that for the *NE* dataset, the two strategies perform equally well on the *Dev* set, whereas the *Subj/Obj* strategy works slightly better on the *Test* set. For *Common Nouns*, *Subj/Obj* is better.

**Results for Ensemble Models.** For each dataset, we combine our best 11 runs and use majority voting to predict the answer for our *Ensemble* model.

In Table 3.8 we show the comparison of our approach with multi-hop models. We report *Accuracy* on the *Dev* and *Test* sets, rounded to the first decimal point as done in previous work. The *AoA Reader* (Cui et al., 2017) uses re-ranking as a post-processing step and the other neural models are not directly comparable. Our ensemble model is comparable to the performance of most multi-hop models that do not use re-ranking.

**Comparison to Previous Work.** Table 3.7 compares our model (*Knowledgeable Reader*) to previous work on the CBT datasets. We show the results of our model with the settings that performed best on the *Dev* sets of the two datasets *NE* and *CN*: for *NE*, ( $D_{ctx+kn}, Q_{ctx}$ ) with 100 facts; for *CN* the *Full model* with 50 facts, both with *CN5Sel*.

Models	Named Entities (NE)		Common Nouns (CN)	
	dev	test	dev	test
Human (ctx + q)	-	81.6	-	81.6
Single interaction				
LSTMs (ctx + q) (Hill et al., 2016)	51.2	41.8	62.6	56.0
AS Reader	73.8	68.6	68.8	63.4
AS Reader (our impl)	75.5	70.3	68.2	64.8
KnReader (ours)	77.4	71.4	71.8	67.6
Multiple interactions				
MemNNs (Weston et al., 2015c)	70.4	66.6	64.2	63.0
EpiReader (Trischler et al., 2016)	74.9	69.0	71.5	67.4
GA Reader (Dhingra et al., 2017b)	77.2	71.4	71.6	68.0
IAA Reader (Sordoni et al., 2016)	75.3	69.7	72.1	69.2
AoA Reader (Cui et al., 2017)	75.2	68.6	72.2	69.4
GA Reader (+feat)	77.8	72.0	74.4	70.7
NSE (Munkhdalai and Yu, 2016)	77.0	71.4	74.3	71.9

Table 3.7 Comparison of KnReader to existing end-to-end neural models on the benchmark datasets.

Models	Named Entities (NE)		Common Nouns (CN)	
	dev	test	dev	test
Human (ctx + q)	-	81.6	-	81.6
Ensemble				
AS Reader (Kadlec et al., 2016)	74.5	70.6	71.1	68.9
KnReader (ours)	78.0	73.3	72.2	70.6
EpiReader (Trischler et al., 2016)	76.6	71.8	73.6	70.6
IAA Reader (Sordoni et al., 2016)	76.9	72.0	74.1	71.0
AoA Reader (Cui et al., 2017)	78.9	74.5	74.7	70.8
Re-ranking				
AoA Reader (re-ranking)	79.6	74.0	75.7	73.1
AoA Reader (ens + re-rank)	80.3	75.6	77.0	74.1

Table 3.8 Comparison of *KnReader* to existing ensemble models and models that use re-ranking.

Note that our work focuses on the impact of external knowledge and employs a *single interaction (single-hop)* between the document context and the question so we primarily compare to and aim at improving over similar models. *KnReader* clearly outperforms prior single-hop models on both datasets. While we do not improve over the state of the art, our model stands well among other models that perform multiple hops.

Solver	Dev	Test
Human solver	89.3	91.7
Guess All (“random”)	25.0	25.0
NO TRAINING, KB ONLY (§3.4.3.1)		
TupleInference	15.9	17.9
PMI (Waterloo corpus)	19.7	21.2
TableILP	20.0	23.4
DGEM	27.4	<b>24.4</b>
NO TRAINING, KB + $\mathcal{F}$ (§3.4.3.2)		
IR with $\mathcal{F}$	25.5	24.8
TupleInference with $\mathcal{F}$	23.6	<b>26.6</b>
DGEM with $\mathcal{F}$	28.2	24.6
TRAINED MODELS, NO $\mathcal{F}$ OR KB (§3.4.3.3)		
Embedd+Sim	44.6	41.8
ESIM	53.9±0.4	48.9±1.1
Question Match (BiLSTM Max-Out)	54.6±1.2	<b>50.2±0.9</b>
KNOWLEDGABLE READER, ORACLE SETUP, $\mathcal{F}$ AND/OR KB (§3.4.2)		
$f$	63.0±2.3	55.8±2.3
$f$ + WordNet	57.6±1.4	56.3±1.3
$f$ + ConceptNet	57.0±1.6	53.7±1.5
$f$ + $k$	80.2±1.1	<b>76.9±0.7</b>
KNOWLEDGABLE READER, KB (§3.4.2)		
ConceptNet only (cn5omcs)	54.0±0.6	51.1±2.1
Wordnet only (cn5wordnet)	54.9±0.4	49.4±1.5
OpenBook + ConceptNet	53.8±1.0	<b>51.2±1.1</b>
OpenBook + Wordnet	53.3±0.7	50.6±0.6

Table 3.9 Scores obtained by various solvers on OpenBookQA, reported as a percentage  $\pm$  the standard deviation across 5 runs with different random seeds. Other baselines are described in the corresponding referenced section. For oracle evaluation, we use the gold science fact  $f$  associated with each question, and optionally the additional fact  $k$  provided by the question author. Bold denotes the best Test score in each category.

### 3.5.2 Open Book Question Answering

Here, we report the results for open book question answering. The results for various baseline models are summarized in Table 3.9, grouped by method category. We make a few observations:

First, the performance of crowd-workers is 92%, whereas the random guess is 25%.

The **second group** shows that pre-trained retrieval solvers (Clark et al., 2018) for multiple-choice science questions perform poorly. One explanation is their correlation with the IR method used for question filtering (Mihaylov et al., 2018).

The **third group** of results suggests that adding  $\mathcal{F}$  to pre-trained models has a mixed effect, improving TupleInference by 8.7% but not changing DGEM.<sup>12</sup> Unlike DGEM, TupleInference relies on brittle word-overlap similarity measures very similar to the ones used by IR. Since IR (KB) gets 0% by design, TupleInference (KB) also has poor performance, and adding  $\mathcal{F}$  helps it find better support despite the brittle measures.

The **fourth group** demonstrates that carefully designed trainable neural models — even if simplistic and knowledge-free — can be surprisingly powerful. The “question match” solver, which simply compares the BiLSTM max-out encoding of the question with that of various answer choices, also achieves 50.2%.<sup>13</sup>

Interestingly, all of these neural knowledge-free baselines simultaneously succeed on 34.4% of the Dev questions and simultaneously fail on 23.6%. For **Question Match** and **ESIM** we also experiment with ELMo (Peters et al., 2018) which improved their score on Test with 0.4% and 1.8%.

The **fifth group** demonstrates the need for external knowledge and deeper reasoning. When the “oracle” science fact  $f$  used by the question author is provided to the knowledge-enhanced reader, it improves over the knowledge-less models by about 5%. However, there is still a large gap, showing that the core fact is insufficient to answer the question. When we also include facts retrieved from WordNet (Miller et al., 1990), the score improves by about 0.5%. Unlike the WordNet gain, adding ConceptNet (Speer et al., 2017) introduces a distraction and reduces the score. This suggests that ConceptNet is either not a good source of knowledge for our task, or only a subset of its relations should be considered. Overall, external knowledge helps, although retrieving the right bits of knowledge remains difficult. In the fourth row of the fifth group of Table 3.9, we use the oracle core fact along with the question author’s interpretation of the additional fact  $k$ . This increases the scores substantially, to about 76%. This big jump shows that improved knowledge retrieval should help with this task. At the same time, we are still not close to the human performance level of 92% due to various reasons: (a) the additional fact needed can be subjective, as hinted at by our earlier analysis; (b) the authored facts  $\mathcal{K}$  tend to be noisy (incomplete, over-complete, or only distantly related), also as mentioned earlier; and (b) even given the true gold facts, performing reliable “reasoning” to link them properly remains a challenge.

In the **last group** we show the results of the *knowledgeable reader* with ConceptNet or WordNet as sources of external knowledge as well as their combination with the OpenBook.

<sup>12</sup>By design, IR with its default corpus gets 0% on OpenBookQA. Hence we don’t consider the effect of adding  $\mathcal{F}$ , which appears artificially magnified.

<sup>13</sup>At the time (April 2018) this model also achieved the best score, 33.87%, on the ARC Reasoning Challenge (Clark et al., 2018). When adapted for the textual entailment task by comparing BiLSTM max-out encodings of premise and hypothesis, it achieves 85% on the SciTail dataset (Khot et al., 2018).

We observe a peak of the performance for the experiments where ConceptNet is used, compared to the baseline trained models without knowledge. While WordNet does not perform better alone, it improves the Question Match (no-knowledge) baseline when combined with the OpenBook.

## 3.6 Discussion and Analysis

In this section, we analyze the results for our *Knowledgeable Reader*. Here we focus our analysis mostly on the cloze-style reading comprehension task which is the main scope of this thesis.

### 3.6.1 Analysis of the empirical results.

Our experiments examined key parameters of the KnReader. As expected, injection of background knowledge yields only small improvements over the baseline model for *Named Entities*. However, on this dataset, our single-hop model is competitive to most multi-hop neural architectures.

The integration of knowledge clearly helps for the *Common Nouns* task. The impact of knowledge sources (Table 3.3) is different on the *Dev* and *Test* sets which indicates that either the model or the data subsets are sensitive to different knowledge types and retrieved knowledge. Table 3.6 shows that attending over the *Subj* of the knowledge triple is slightly better than *Obj*. This shows that using a *Key-Value* memory is valuable. A reason for the lower performance of *Obj/Obj* is that the model picks facts that are similar to the candidate tokens, not adding much new information. From the empirical results we see that training and evaluation with fewer facts is slightly better. We hypothesize that this is related to the lack of supervision on the retrieved and attended knowledge.

### 3.6.2 Interpreting Component Importance

Figure 3.6 shows the impact on prediction accuracy of individual components of the *Full model*, including the interaction between  $D$  and  $Q$  with  $ctx$  or  $ctx + kn$  (w/o  $ctx$ -only). The values for each component are obtained from the attention weights, without retraining the model. The difference between blue (left) and orange (right) values indicates how much the module contributes to the model. Interestingly, the ranking of the contribution ( $D_{ctx}, Q_{ctx+kn} > D_{ctx+kn}, Q_{ctx} > D_{ctx+kn}, Q_{ctx+kn}$ ) corresponds to the component importance ablation on the *Dev* set, lines 5-8, Table 3.5.

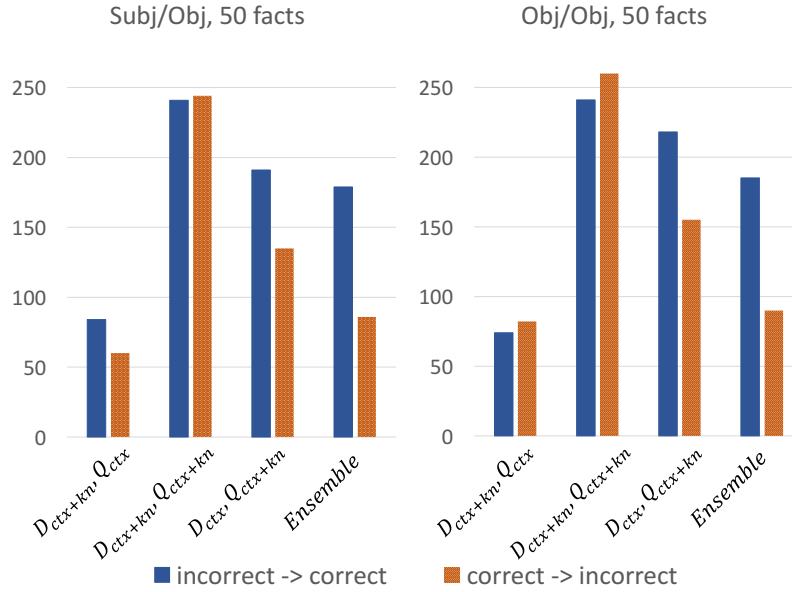


Fig. 3.6 # of items with reversed prediction ( $\pm$ correct) for each combination of  $(ctx+kn, ctx)$  for Q and D. We report the number of *wrong*  $\rightarrow$  *correct* (blue) and *correct*  $\rightarrow$  *wrong* (orange) changes when switching from score w/o knowledge to score w/ knowledge. The best model type is *Ensemble*. (Full model w/o  $D_{ctx}, Q_{ctx}$ ).

### 3.6.3 Qualitative Data Investigation

We will use the attention values of the interactions between  $D_{ctx(+kn)}$  and  $Q_{ctx(+kn)}$  and attentions to facts from each candidate token and the question placeholder to interpret how knowledge is employed to make a prediction for a single example.

**Method: Interpreting Model Components.** We manually inspect examples from the evaluation sets where *KnReader* improves prediction (blue (left) category, Fig. 3.6) or makes the prediction worse (orange (right) category, Fig. 3.6). Figure 3.8 shows the question with a placeholder, followed by answer candidates and their associated attention weights as assigned by the model *w/o knowledge*. The matrix shows selected facts and their assigned weights for the question and the candidate tokens. Finally, we show the attention weights determined by the knowledge-enhanced D to Q interactions. The attention to the correct answer (*head*) is low when the model considers the text alone (*w/o knowledge*). When adding retrieved knowledge to the Q only (row  $ctx, ctx + kn$ ) and to both Q and D (row  $ctx + kn, ctx + kn$ ) the score improves, while when adding knowledge to D alone (row  $ctx + kn, ctx$ ) the score remains ambiguous. The combined score *Ensemble* (see Eq. 3.13) then takes the final decision for the answer. In this example, the question can be answered without the story. The model tries to find knowledge that is related to *eyes*. The fact *eyes /r/PartOf head* is not contained in the retrieved knowledge but instead the model selects the fact *ear /r/PartOf*

*head* which receives the highest attention from  $Q$ . The weighted *Obj* representation (*head*) is added to the question with the highest weight, together with *animal* and *bird* from the next highly weighted facts. This results in a high score for the  $Q_{ctx}$  to  $D_{ctx+kn}$  interaction with candidate *head*.

Using the method described above, we analyze several example cases that highlight different aspects of our model:

**Case 1** We provide an extended illustration of the example discussed in the main paper in Figure 3.8. We manually inspect examples from the evaluation sets where *KnReader* improves prediction or makes the prediction worse. Figure 3.8 shows the question with placeholder, followed by answer candidates and their associated attention weights as assigned by the model *w/o knowledge*. The matrix shows selected facts and their learned weights for the question and the candidate tokens. Finally, we show the attention weights determined by the knowledge-enhanced D to Q interactions.

The attention to the correct answer (*head*) is low when the model considers the text alone (*w/o knowledge*). When adding retrieved knowledge to the  $Q$  only (row  $ctx, ctx + kn$ ) and to both  $Q$  and  $D$  (row  $ctx + kn, ctx + kn$ ) the score improves, while when adding knowledge to  $D$  alone (row  $ctx + kn, ctx$ ) the score remains ambiguous. The combined score *Ensemble* then takes the final decision for the answer. In this example, the question can be answered without the story. The model tries to find knowledge that is related to *eyes*. The fact *eyes /r/PartOf head* is not contained in the retrieved knowledge but instead the model selects the fact *ear /r/PartOf head* which receives the highest attention from  $Q$ . The weighted *Obj* representation (*head*) is added to the question with the highest weight, together with *animal* and *bird* from the next highly weighted facts. This results in a high score for the  $Q_{ctx}$  to  $D_{ctx+kn}$  interaction with candidate *head*.

**Case 2** Figure 3.9 shows another interesting example. The document is part of the *The kings new clothes* by Hans Christian Andersen. While, given the story, many of the choices are plausible (*cloth, clothes, nothing, air, cloak*) the model without knowledge selects *cloth* as the most probable answer. Adding the knowledge facts reverts the answer. We can speculate that the reason is the fact *clothes /r/Antonym undressed* retrieved by the answer candidate token *clothes* which has multiple occurrences in the text, and since the updated representation combines well with the phrase *put on* which is antonym to undressed *clothes /r/Antonym undressed* and *clothes /r/Antonym naked*. A reason for this could also be the high frequency of clothes in the story. However, the example cannot be answered using the story context alone, as it talks about the imaginary, not existing (*air, nothing*) new clothes of the king.

The example also shows what kind of knowledge is missing in our currently used resources: ideally, the question can be answered using information from the question alone,

by analyzing the meaning of the phrases *take off your clothes* and *then we will put on the new XXXX*. If they were available, the model could exploit the knowledge that *taking off (clothes)* and *putting on (clothes)* are actions often performed in temporal sequence.

**Case 3** In Figure 3.10 we have an example where the model overcomes the frequency bias of the story (*magician* occurs 4 times) to select a more plausible example (*father*) using the fact *father /r/Antonym son*.

**Case 4** Figure 3.11 shows an example where a correct initial prediction obtained without knowledge is reversed and a clearly wrong answer is selected instead. Although a relevant fact is selected (*people /r/UsedFor help you*), apparently, the model misses the information that *brothers are people* and can't combine the acquired concept *help you* with the question context *and with their help dragged ...*, and thus, the correct answer is not sufficiently promoted.

**Case 5** The example in Figure 3.12 illustrates the lack of knowledge about locations. The context of *Q* talks about *climbing up* and while the text-only module selects the right answer *cliff*, the available knowledge modifies the representation and reverses the answer to *sea* which is *usually* on lower level. Here the association is made with a *cliff* and *sea* by the fact *inlet /r/PartOf sea* and *beach /r/PartOf shore*). That is, the context-only neural representation guesses that the plausible answer is similar to *cliff* (*inlet and shores are usually associated with cliff*). Again, we are missing knowledge of actions, e.g., that *climbing* is done to move up steep locations such as hills, or cliffs. In future work we plan to experiment with sources that offer more information about events.

Here we summarize our observations:

**(i.) Answer prediction from Q or Q+D.** In both human and machine RC, questions can be answered based on the question alone (Figure 3.8) or jointly with the story context (Case 2, *Suppl.*). We show that empirically, enriching the question with knowledge is crucial for the first type, while enrichment of Q and D is required for the second.

**(ii.) Overcoming frequency bias..** We show that when appropriate knowledge is available and selected, the model is able to correct a frequency bias towards an incorrect answer (Cases 1 and 3).

**(iii.) Providing appropriate knowledge.** We observe a lack of knowledge regarding events (e.g. *take off* vs. *put on clothes*, Case 2; *climb up*, Case 5). Nevertheless relevant knowledge from CN5 can help predicting infrequent candidates (Case 2).

**(iv.) Knowledge, Q and D encoding.** The context encoding of facts allows the model to detect knowledge that is semantically related, but not surface near to phrases in Q and D (Case 2). The model finds facts to non-trivial paraphrases (e.g. *undressed–naked*, Case 2).



Q: UNK\_59 did not say anything ; but when the other two had passed on she bent down to the bird , brushed aside the feathers from his xxxxx , and kissed his closed eyes gently

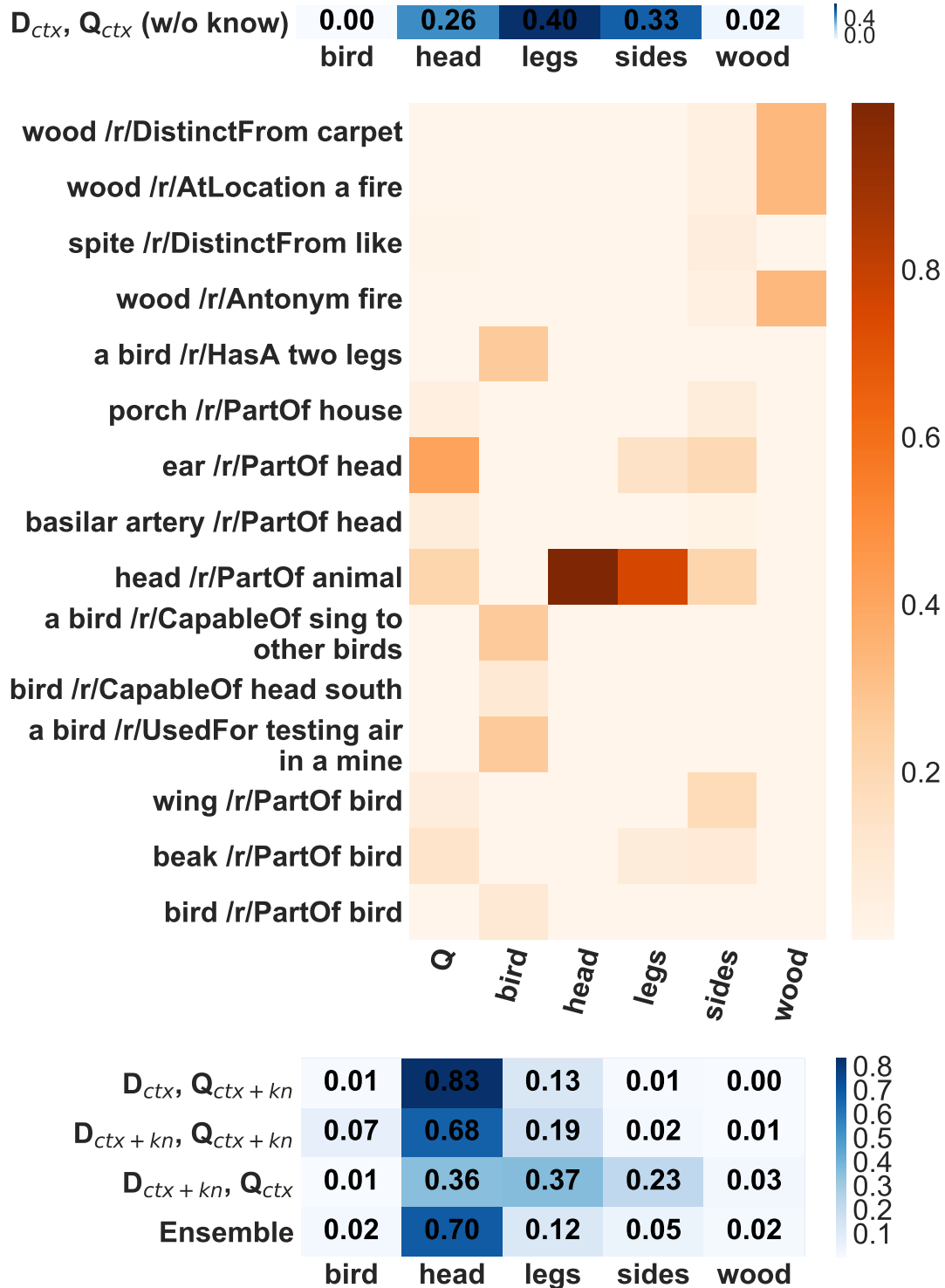


Fig. 3.7 Interpreting the components of *KnReader*. Adding knowledge to *Q* and *D* increases the score for the correct answer. Results for top 5 candidates are shown. (Full model, CN data, CN5Sel, Subj/Obj, 50 facts)

**Story:** ... ‘ what has a bird , in spite of all his singing , in the winter-time ? he must starve and freeze , and that must be very pleasant for him , i must say ! ’

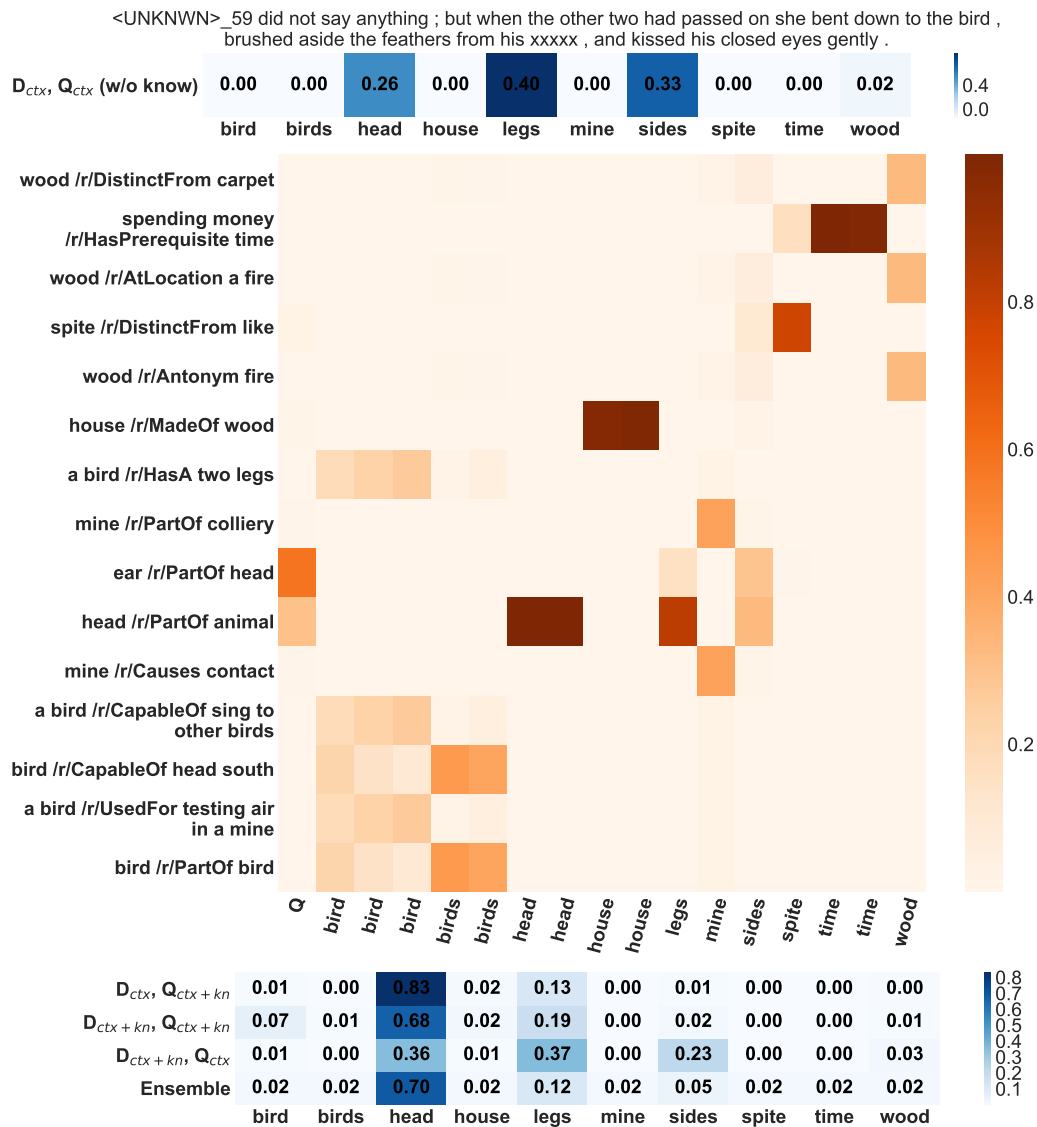


Fig. 3.8 **Case 1:** Interpreting the components of *KnReader* (Full model). Adding retrieved knowledge to *Q* and *D* helps the model to increase the score for the correct answer. Results for top 5 candidates are shown. (Subj/Obj as key-value memory, 50 facts, CN5Sel) (Item #357)

**Story:** ... they pretended they were taking the cloth from the loom , cut with huge scissors in the air , sewed with needles without thread , and then said at last , ‘ now the clothes are finished ! ’ the emperor came himself with his most distinguished knights , and each impostor held up his arm just as if he were holding something , and said , ‘ see ! here are the breeches ! here is the coat ! here the cloak ! ’ and so on .  
 ‘ spun clothes are so comfortable that one would imagine one had nothing on at all ; but that is the beauty of it ! ’ ‘ yes , ’ said all the knights , but they could see nothing , for there was nothing there .

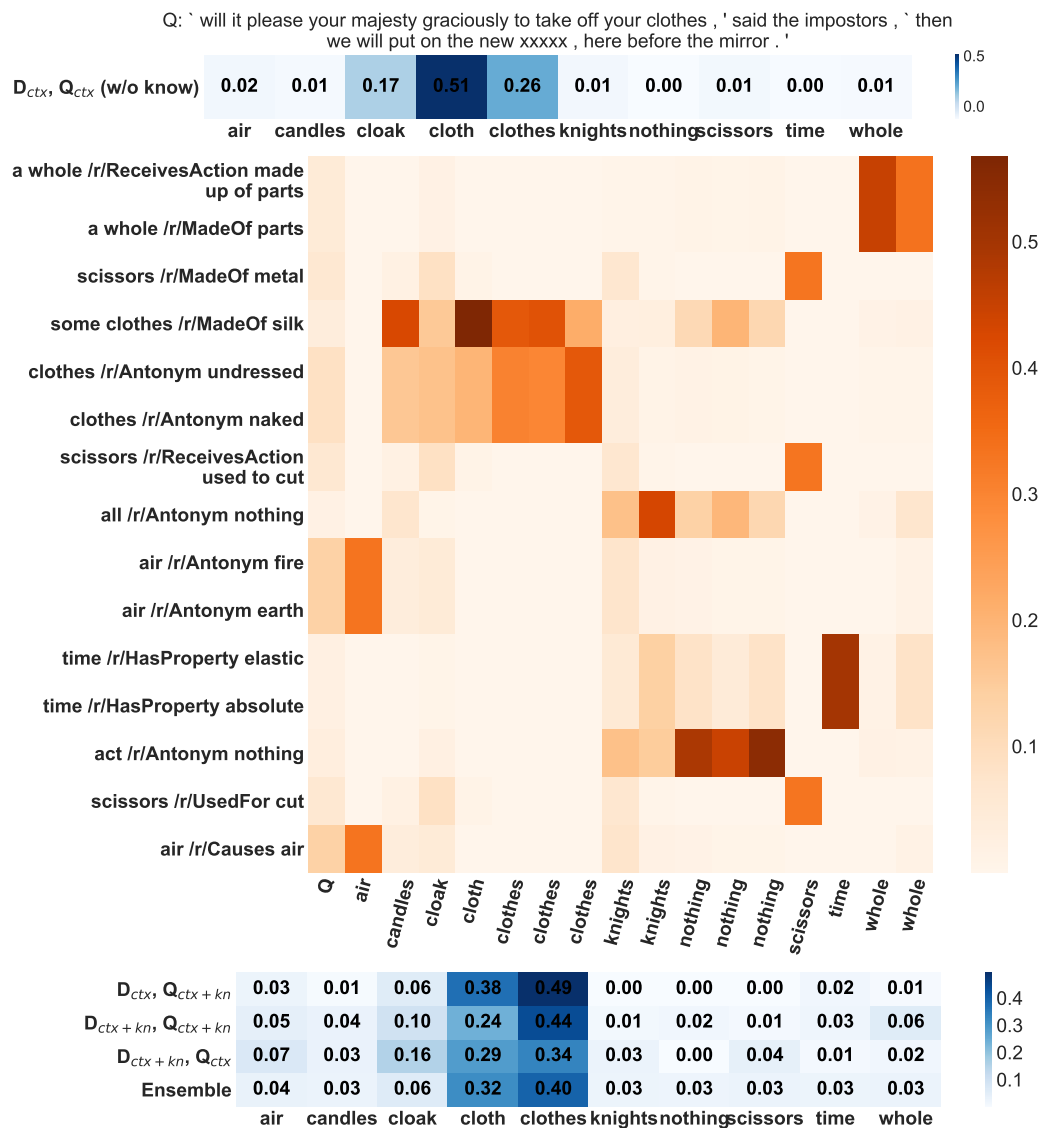


Fig. 3.9 Case 2: Interpreting the components of *KnReader* (Full model). Adding retrieved knowledge to *Q* and *D* helps the model to increase the score for the correct answer. Results for top 5 candidates are shown. (Subj/Obj as key-value memory, 50 facts, CN5Sel) (Item #52)

**Story:** ... a celebrated magician , who had given the seed to my father , promised him that they would grow into the three finest trees the world had ever seen .  
 ‘ after this i had the beautiful fruit of these trees carefully guarded by my most faithful servants ; but every year , on this very night , the fruit was plucked and stolen by an invisible hand , and next morning not a single apple remained on the trees .  
 for some time past i have given up even having the trees watched .

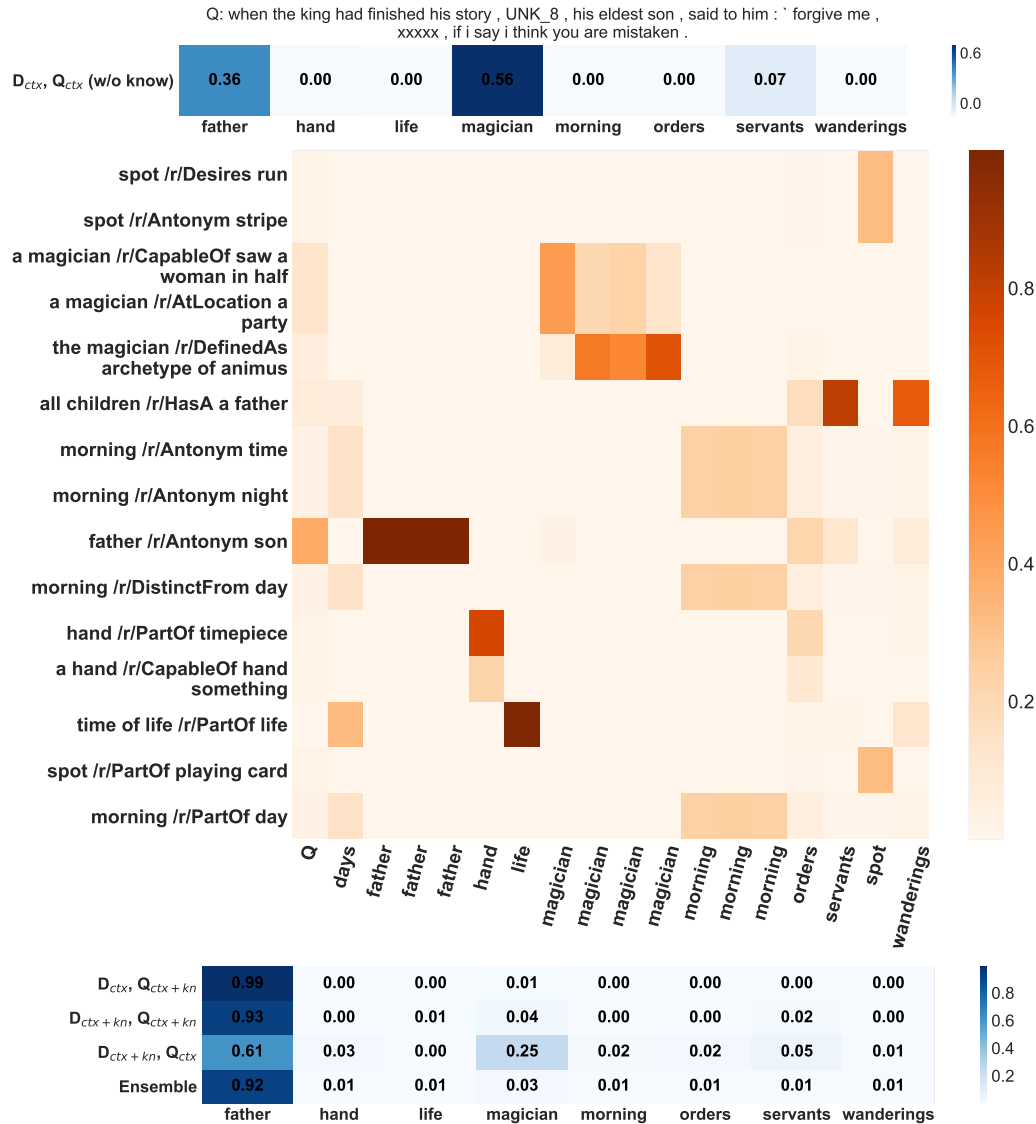


Fig. 3.10 **Case 3:** Interpreting the components of *KnReader (Full model)*. Adding retrieved knowledge to *Q* and *D* helps the model to increase the score for the correct answer. Results for top 5 candidates are shown. (Subj/Obj as key-value memory, 50 facts, CN5Sel) (Item #240)

**Story:** ... in the same village there lived three brothers , who were all determined to kill the mischievous hawk . ... his eyelids closed , and his head sank on his shoulders , but the thorns ran into him and were so painful that he awoke at once . the hawk fell heavily under a big stone , severely wounded in its right wing . the youth ran to look at it , and saw that a huge abyss had opened below the stone .

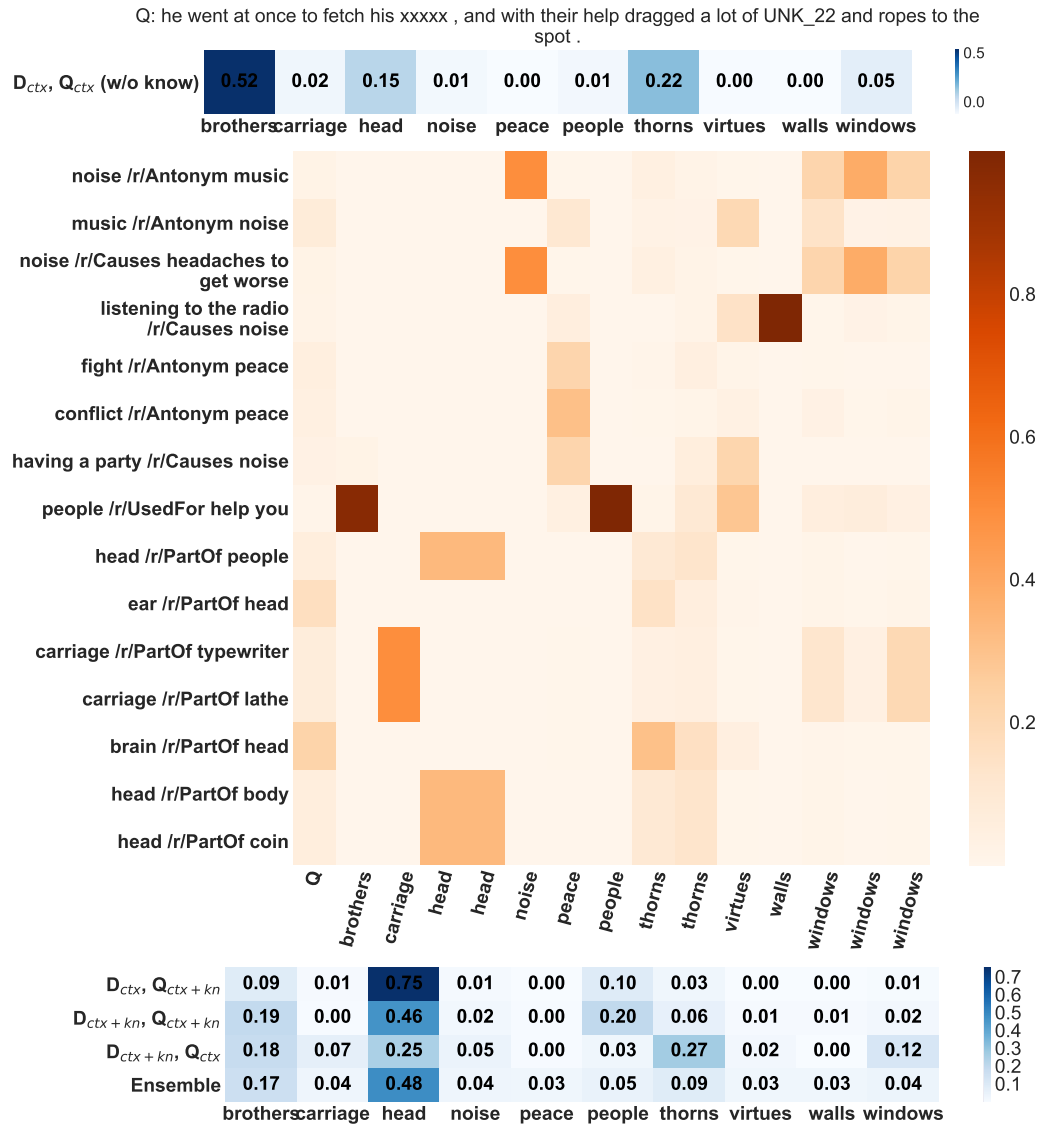


Fig. 3.11 **Case 4:** Interpreting the components of *KnReader* (Full model). Adding retrieved knowledge to *Q* and *D* confuses the model and decreases the score for the correct answer. Results for top 5 candidates are shown. (Subj/Obj as key-value memory, 50 facts, CN5Sel) (Item #172)

**Story:** ... i also lay this belt beside you , to put on when you awaken ; it will keep you from growing faint with hunger .  
 the woman now disappeared , and unk\_98 woke , and saw that all her dream had been true .  
 the rope hung down from the cliff , and the clew and belt lay beside her .

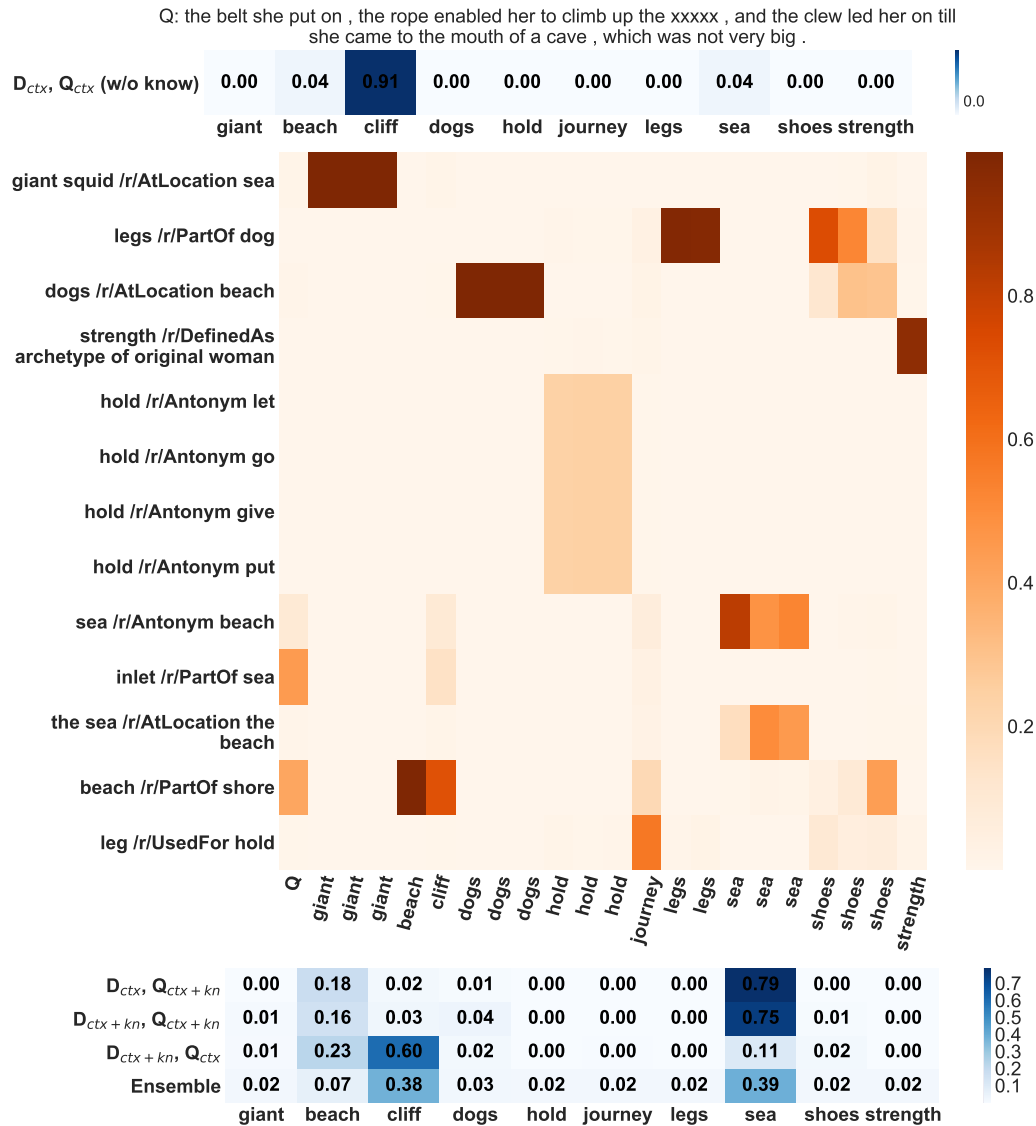


Fig. 3.12 **Case 5:** Interpreting the components of *KnReader* (Full model). Adding retrieved knowledge to *Q* and *D* confuses the model and decreases the score for the correct answer. Results for top 5 candidates are shown. (Subj/Obj as key-value memory, 50 facts, CN5Sel) (Item #187)

### 3.6.4 Success and Failure Examples for Open Book QA

We give some examples of questions that were answered correctly/incorrectly by various groups of models on the OpenBookQA(Mihaylov et al., 2018) dataset. We include here the first three questions in each case.

**Neural Baseline Successes** We begin with three examples of questions that all neural models without external knowledge (namely Question Match, Plausible Answer, One-Odd-Out, and ESIM from the fourth group in Table 3.10) predicted correctly.

---

A body may find its temperature to be lowered after (A) water is heated up (B) **fluid spreads from pores** (C) the air becomes arid (D) the sky stays bright

---

Oil is a non-renewable resource which tells us that when (A) it can be remade (B) it can be found in other places (C) there is an endless supply (D) **the final barrel is gone, there supply is finished**

---

Magma contains (A) **particles of iron** (B) Loads of leaves (C) Soda (D) Silly Putty

---

Table 3.10 Sample questions predicted **correctly** (172/500) by all trained neural models without external knowledge.

In these examples, we observe that the correct answer usually contains a word that is semantically closer (than words in other answer choices) to an important word from the question: *pores* to *body*; *non-renewable* (negative sentiment) to *gone, finished* (also negative sentiment); *iron* to *magma (liquid rock)*.

---

Frilled sharks and angler fish live far beneath the surface of the ocean, which is why they are known as (A) **Deep sea animals** (B) fish (C) Long Sea Fish (D) Far Sea Animals. **Oracle facts:** (*f*) deep sea animals live deep in the ocean. (*k*) Examples of deep sea animals are angler fish and frilled sharks.

---

Gas can fill any container it is given, and liquid (A) is standard weight and size (B) is the opposite of variable (C) only needs a few (D) **uses what it needs**. **Oracle facts:** (*f*) Matter in the liquid phase has definite volume. (*k*) liquid cannot spread endlessly.

---

When birds migrate south for the winter, they do it because (A) **they are genetically called to** (B) their children ask for them to (C) it is important to their happiness (D) they decide to each year. **Oracle facts:** (*f*) migration is an instinctive behavior. (*k*) instinctive is genetic.

---

Table 3.11 Sample questions predicted **correctly** by the  $f + k$  Oracle model (405/500) but were predicted **incorrectly** by all of the 4 neural models without knowledge (total of 69 out of 405).

---

An example of data collection is: (A - 0.9977) **Deleting case files on the computer**, (B - 0.0000) Touching evidence without gloves, (C - 0.0004) **speaking with a witness**, (D - 0.0019) Throwing documents in the trash. **Oracle facts:** ( $f$ ) An example of collecting data is measuring. ( $k$ ) Interviews are used to collect data.

If a farmland up the hill gets rainfall, what could happen to lower lands? (A - 0.0005) **all of these**, (B - 0.0245) they could get fertilizer washed to them, (C - 0.9542) **they could experience unfavorable chemical change in their lands**, (D - 0.0208) they could have their lands poisoned. **Oracle facts:** ( $f$ ) runoff contains fertilizer from cropland. ( $k$ ) fertilizers for certain crops could poison other crops or soil types.

Layers of the earth include all but: (A - 0.0429) mantle, (B - 0.0059) **center**, (C - 0.0334) crust, (D - 0.9177) **inner core**. **Oracle facts:** ( $f$ ) the crust is a layer of the Earth. ( $k$ ) the last layer is the outer core.

---

Table 3.12 Sample **questions** predicted **incorrectly** by all models w/o knowledge, as well as the  $f + k$  Oracle model, even though the Oracle model has confidence higher than 0.90.

**Neural Baseline Failures, Oracle Success** Table 3.11 shows example questions (with the Oracle facts) from the Dev set that were predicted correctly by the  $f + k$  Oracle model (405/500) but incorrectly by all of the 4 neural models without knowledge (69/405). In contrast to Table 3.10, a simple semantic similarity is insufficient. The questions require chaining multiple facts in order to arrive at the correct answer.

**Neural Baseline and Oracle Failures** 42/500 questions in the Dev set were predicted incorrectly by all models without external knowledge, as well as by the Oracle  $f + k$  model. In Table 3.12 we show 3 such questions. In all cases, the Oracle  $f + k$  model made an incorrect prediction with confidence higher than 0.9.

As noted earlier, there are several broad reasons why even this so-called oracle model fails on certain questions in OpenBookQA. In some cases, the core fact  $f$  associated with a question  $q$  isn't actually helpful in answering  $q$ . In many other cases, the corresponding second fact  $k$  is noisy, incomplete, or only distantly related to  $q$ . Finally, even if  $f$  and  $k$  are sufficient to answer  $q$ , it is quite possible for this simple model to be unable to perform the reasoning that's necessary to combine these two pieces of textual information in order to arrive at the correct answer.

In the shown examples, the first question falls outside the domain of *Science* where most of the core facts come from. The scientific fact “( $f$ ) An example of collecting data is measuring” is transformed into a question related to the law and judicial domain of *collecting data for a (court) case*. This is an indication that the model trained on the Train set does not perform well on distant domains, even if the core facts are provided.



In the second question, we have an option *all of these*. Indeed, the selected answer seems the most relevant (a generalized version of the other two), but the model did not know that if we have an option *all of these* and all answers are plausible, it should decide if all answers are correct and not pick the “most likely” individual answer.

The third question again requires the model to select a special type of aggregate answer (“all but xyz”), but the related Oracle facts are pointing to a specific answer.

### 3.7 Summary and Conclusions

In this chapter, we proposed a neural model that incorporates external commonsense knowledge, building on a single-turn neural model for cloze-style reading comprehension and open book question answering. Incorporating external knowledge improves its results for cloze-style reading comprehension with a relative error rate reduction of 9% on *Common Nouns*, thus the model can compete with more complex models. We show that the types of knowledge contained in ConceptNet are useful both for cloze-style reading comprehension and science question answering. For cloze-style reading comprehension, we experimented with knowledge encoded as directional triples, and for science question answering with OpenBookQA, we used natural language text since it was available as additional external resources with the dataset. We provide quantitative and qualitative evidence of the effectiveness of our model, which learns how to select relevant knowledge to improve reading comprehension. The attractiveness of our model lies in its *transparency and flexibility*: due to the attention mechanism, we can trace and analyze the facts considered in answering specific questions. This opens up a deeper investigation and future improvement of RC models in a targeted way, allowing us to investigate what knowledge sources are required for different data sets and domains. Since our model directly integrates background knowledge with the document and question context representations, it could be adapted to very different task settings where we have a pair of two arguments (i.e. *entailment, retrieval, etc.*).

# Chapter 4

## Neural Machine Reading Comprehension using Contextual Representations Pre-trained on Lower-Level Supervised Language Tasks

### 4.1 Introduction

Reading comprehension (RC) is a language understanding task, typically evaluated in a question answering setting, where a system reads a text passage (document  $D$ ) and answers questions ( $Q$ ) about it. Recently, work on novel datasets for machine reading comprehension gained a lot of attention: ‘CNN/Daily Mail’ (Hermann et al., 2015), Children Book Test (Hill et al., 2016), Who Did What (Onishi et al., 2016), bAbI (Weston et al., 2015b) and before that MCTest (Richardson et al., 2013). Most recently SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017) and TriviaQA (Joshi et al., 2017) were created using crowd-sourcing.

Sugawara et al. (2017); Chen et al. (2016a); Rajpurkar et al. (2016) have shown that Reading Comprehension requires a set of language skills such as paraphrase detection, recognition of named entities, natural language inference, etc. The common approach to tackling higher-level tasks such as Reading Comprehension is to build a complex neural model that reads a large-scale dataset and tries to learn all required skills at once.

We propose learning the ‘skills’ required for reading comprehension from existing supervised language tasks. We evaluate the performance of several learned lower-level ‘skills’ for MRC on SQuAD (Rajpurkar et al., 2016) by integrating them in a base neural model. This is in contrast to Conneau et al. (2017b) who learn sentence compression representations

from a large supervised corpus and transfer the learned knowledge to a set of lower-level tasks. Our approach is similar to McCann et al. (2017) who use weights pre-trained on machine translation to improve a strong RC system (Xiong et al., 2016).

We propose a simple RC model that allows us to combine learned ‘skill’ representations easily and analyze the learning behavior of this skill transfer model. We show that using such skills, learned from specialized corpora, boosts the performance of a good baseline RC system (i) early in training and (ii) when training on smaller portions (2, 5, 10, and 25 percent) of the original training data.

## 4.2 Method

We tackle the RC task using lower-level ‘skill’ tasks. To do that, we implement a baseline model to represent the relation between a given question and the story context and enrich the representation by reusing encoder weights from the chosen ‘skill’ tasks.

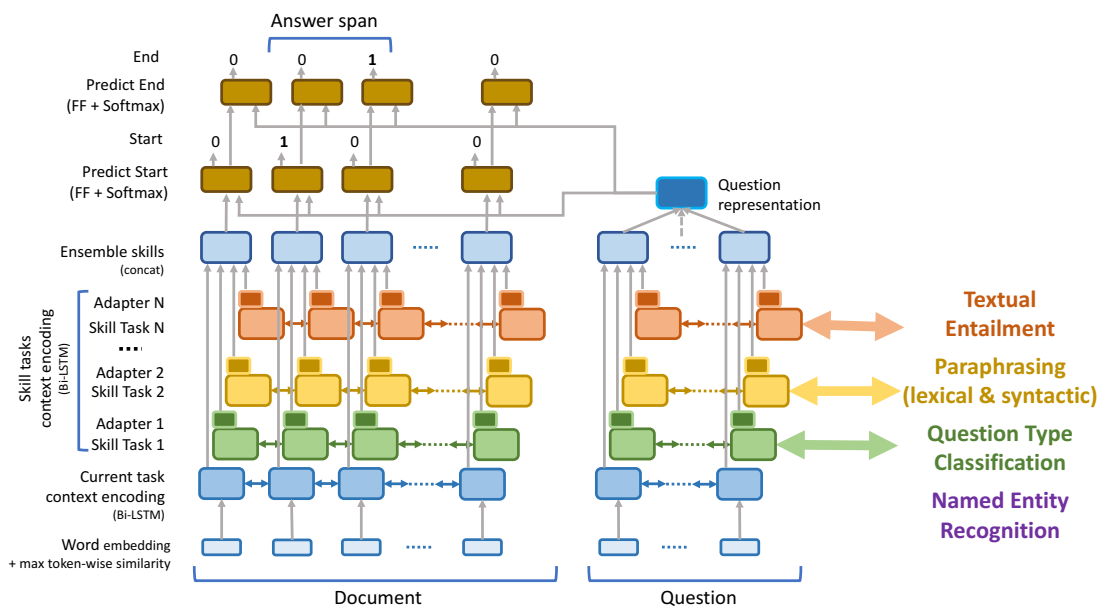


Fig. 4.1 Skillful Reader: Architecture for transferring knowledge from ‘skill’ language tasks to an RC model.

Our skill transfer model is visualized in Figure 4.1. It can be summarized in two main steps: (i) *Skill Learning*: Train context encoder-based (Bi-LSTM) models for several language ‘skill’ tasks and preserve the learned encoder weights; (ii) *Neural Skill Transfer*: Reuse the learned context encoder skill weights to encode and extend the text context of document and question, in a simple model for the target task (QA/RC).

Task	Task Type
Named Entity Recognition (NER), Semantic Role Labeling (SRL)	Sequence Labeling
Question Type Classification (TREC) Entity Description Classification (DbPedia)	Text Classification
Discourse Relation Sense Classification (CoNLL 16 ST) Natural Language Inference (SNLI, MNLI) Paraphrasing Detection (Quora, MRPC)	Relation Classification

Table 4.1 Skill tasks, datasets, and model types used in the experiments

Dataset	Model	Train size	Dev size	Test size	Vocab
NER (CoNLL 2012)	SeqLbl	59914	8217	8252	36090
SRL (CoNLL-05 ST)	SeqLbl	39832	1346	426	39549
QTC	TC/TC-TW	5452	500	-	8868
DbPedia	TC/TC-TW	560000	70000	70000	1016562
DR - Explicit (CoNLL 16 ST)	TC-P	14865	678	584	32049
DR - Non-Exp (CoNLL 16 ST)	Rel	17068	730	983	36664
SNLI	Rel	550152	10000	10000	36732
MNLI	Rel	392702	10000	10000	86330
SNLI+MNLI	Rel	942854	10000	10000	100220
Quora Question Paraphrasing	Rel	384290	10000	10000	117099
MRPC	Rel	4076	1725	1725	14801

Table 4.2 Skill datasets and the size of training, evaluation sets (number of examples), and vocabulary size (number of tokens).

This model can be considered similar to *progressive neural networks* (Rusu et al., 2016) without the notion of sequential learning of the tasks. In Table 4.1 we show the skill tasks used in this study grouped by task type: Sequence Labeling, Text Classification, or Relation Classification.

### 4.2.1 Skill Tasks

We transfer knowledge from several ‘skill’ tasks presented in Table 4.2) together with their data properties. We briefly describe each of the tasks as follows:

**Named Entity Recognition (NER)** (Pradhan et al., 2012; Weischedel et al., 2011) is a task that aims at automatically tagging entities in natural language text. We hypothesize that having a NER skill could be beneficial since many of the questions from machine reading datasets such as SQuAD (Rajpurkar et al., 2016) are about different entities (people, locations,

etc.). In our experiments, we employ NER from Ontonotes (Weischedel et al., 2011) to train our skill encoders in a sequence labeling setting (see Section 4.2.2).

**DbPedia Entity Type Classification** (Zhang et al., 2015c) is a dataset for text classification of entity descriptions extracted from the English DbPedia <sup>1</sup> in 15 entity classes such as *Company, Athlete, Artist, etc.*. With a similar intuition to using NER, we hypothesize that such a task would help a model to detect which tokens in a given paragraph are from the description of an entity with a specific type. Having such representations could help answer questions such as ‘What is Luis Armstrong famous for?’, which requires the model to focus on a relevant text similar to ‘was an American trumpeter, composer, vocalist, and actor’.

**Question Type Classification** (Li and Roth, 2002a) aims at classifying the type for a given question. The QTC dataset by Li and Roth (2002a) offers annotation on six coarse types (ABBR, DESC, ENTITY, HUM, NUM) and a total of 50 fine-grained types (ex. DESC:reason, ENTITY:animal, HUM:group, etc.). In this work, we train a model to classify questions in these fine-grained classes. For example ‘Why do heavier objects travel downhill faster?’ is of type ‘DESC:reason’ and the answer would be "gravity acceleration". Our motivation is that understanding the type of a question should be beneficial for a question answering model.

**Semantic Role Labeling (SRL)** (Carreras and Màrquez, 2005) represents the relations of ‘who did what, to whom’ - it decomposes a sentence into verbs (V) and it’s depending arguments (ARG0, ARG1, ARG2, etc.). If a simple neural network model is able to perform this task, it could be able to recognize events and their participants, when integrated into a MRC system. We use the CoNLL 2005 SRL dataset (Carreras and Màrquez, 2005) in a sequence labeling setting.

**Discourse Relation Sense Classification** (Xue et al., 2015, 2016a) is a task that aims at detecting the relation of text arguments (sentences or phrases) in discourse. Example for a relation in discourse with type *Contingency.Cause.Reason* between two arguments ARG1, ARG2, and discourse connective CONN is ‘[Warsaw gained the title of the ‘Phoenix City’]<sub>ARG0</sub> [because]<sub>Conn</sub> [it has survived many wars, conflicts, and invasions throughout its long history.]<sub>ARG1</sub>’. We hypothesize that understanding discourse relations would help in answering specific questions such as ‘Why was Warsaw called ‘Phoenix City‘?’. For pre-training we use the data from the CoNLL 2016 Shared Task on Discourse Relation Sense Classification (Xue et al., 2016b). We teach two separate neural network models to learn

---

<sup>1</sup>DbPedia - <https://wiki.dbpedia.org/>

detecting the fine-grained type of *Explicit* (arguments connected with a discourse connective) and *Non-Explicit* (the relation is not characterized by a given discourse connective) discourse relations.

**Natural Language Inference** is a task that aims at recognizing if a given hypothesis is in entailment, contradiction, or neutral relation to a given premise. We are inspired by work by Conneau et al. (2017b) showing that using representations trained on NLI can be helpful for sentence similarity tasks. In this work, we use two NLI datasets: SNLI (Bowman et al., 2015) which contains natural language inference examples from common knowledge, and MNLI (Williams et al., 2018) which covers a broad set of domains such as *Fiction, Travel, Government, etc.* We hypothesize that neural representations, trained on NLI data would help find answers which require inference beyond text matching. For example, if we have the question ‘Who is the usual captain of the Enterprise spaceship?’, and the sentence ‘Picard is temporarily replaced by Jellico from being captain of Enterprise.’ we need to understand it entails a hypothetical sentence ‘Picard was the regular captain of Enterprise’ so that we arrive at the correct answer ‘Picard’, rather selecting ‘Jellico’.

**Paraphrase Identification** is another skill which would be beneficial for answering questions (Sugawara et al., 2017). To ‘learn’ this skill we use two datasets different datasets for paraphrase detection. *Microsoft Research Paraphrase Corpus (MRPC)* (Dolan and Brockett, 2005) is a small dataset that contains high-quality pairs of sentences in English that are labeled as being paraphrases or not. With similar labels, *Quora Question Pairs* is a large-scale dataset that contains around 400k pairs of questions collected from quora.com.<sup>2</sup>

### 4.2.2 Skill Learning Architectures

For encoding the ‘skill’ knowledge from lower-level tasks we first implement simple context encoder models for each low-level learning setup. In this work, we implement three types of models for encoding language tasks: *Sequence Labeling, Text Classification, and Relation Classification.*

**Sequence Labeling** is applied for labeling each token in a text with a specific category. For this type of encoder model, we use a vanilla Bi-LSTM (Graves and Schmidhuber, 2005) architecture, that uses word embeddings as input and a label projection layer with a softmax layer to predict the sequence labels. While this does not lead to a supreme performance in

<sup>2</sup><https://www.kaggle.com/c/quora-question-pairs>

any sequence-labeling task, it is a reasonable unified baseline for our setup (Ma and Hovy, 2016; Lample et al., 2016; Chiu and Nichols, 2016). We hypothesize that by using a simple architecture for the skill learning model, we can encode the skill knowledge in the context layer. With the sequence labeling model (SeqLbl), we encode knowledge from the tasks of Named Entity Recognition (NER) based on the CoNLL 2012 NER dataset derived from Ontonotes (Pradhan et al., 2012) and the Semantic Role Labeling (Carreras and Màrquez, 2005) from CoNLL 2005 shared task. For both, we use the BIO schema for label encoding as shown in Figure 4.2.

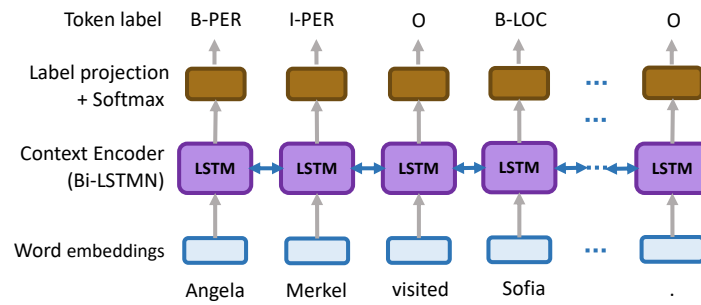


Fig. 4.2 Vanilla Bi-LSTM for sequence labeling (NER)

**Text Classification** is applied to categorize a given word token sequence. Since our RC task is cast as a QA problem, we propose to employ the skill of Question Type (QT) Classification, using the TREC Question Classification dataset (Li and Roth, 2002b). To ensure that we learn diverse question types we use the fine-grained classification with 50 classes for training. The task is to classify a given question according to the type of its answer. Since the answer is not given during training, using this task we learn a valuable skill to implicitly recognize what type of answer to look for in a text. We employ a simple model with a Bi-LSTM context encoder and label prediction layer. We have two modifications of the model depending on the label-prediction layer. We train the text classification using token-wise (TW) and sentence-wise supervision. Figure 4.3 A) shows sentence-wise supervision where the label is predicted from an aggregated representation from the context layer. In Figure 4.2 B) we have the token-wise label prediction. That is, instead of retrieving a single vector representation of the sentence (with avg- or max-pooling, etc.) and predicting the label, we project the token context representation  $c_{t_{1..n}}$  to the label space (50 classes)  $c_{t_{1..n}}^{lbl}$  and *sum* the soft label prediction for each token, to obtain the label for the sentence:

$$r_{sent}^{lbl} = softmax(\sum c_{t_{1..n}}^{lbl}).$$

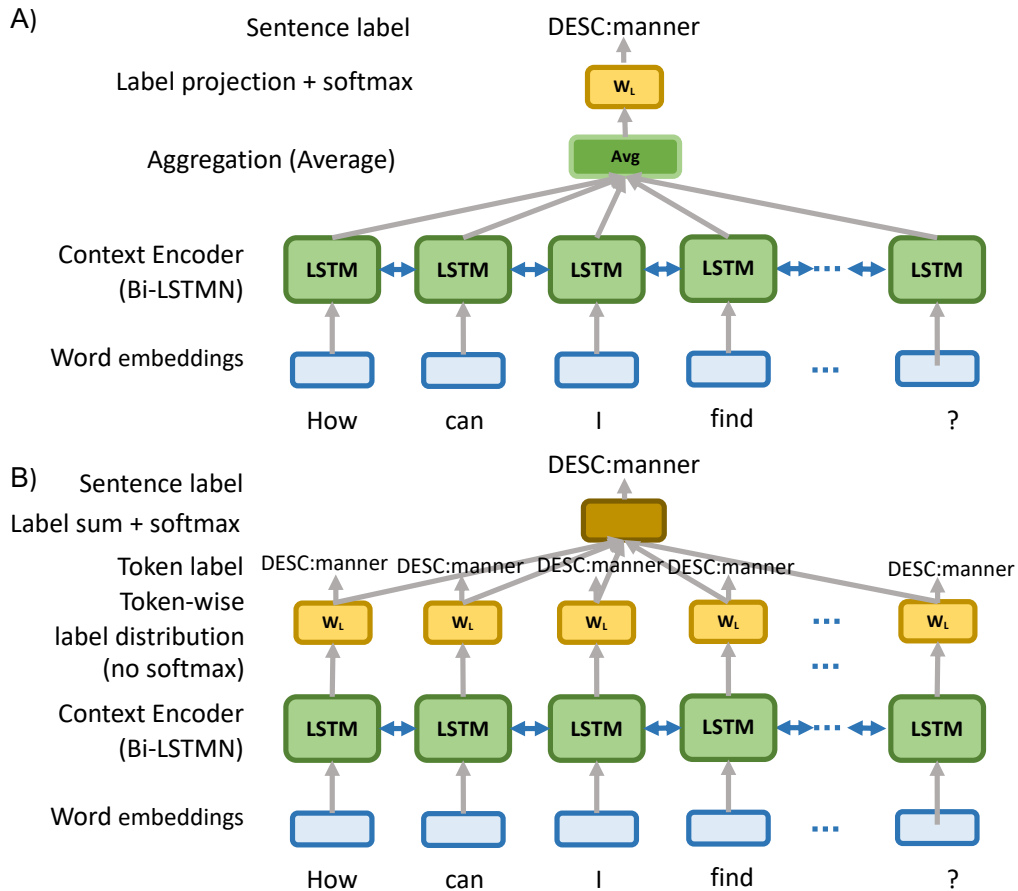


Fig. 4.3 Text classification (Question Type Classification) with Bi-LSTM context encoder and A) sentence-wise label prediction and B) token-wise label supervision.

We hypothesize that with lower-level label supervision, we can propagate the knowledge expressed by the label to the context representations of specific tokens. This is a form of deep supervision (Lee et al., 2015), similar to (Lipton et al., 2015).

**Relation Classification** is used to classify the relationship between two arguments represented as text. We implement relation classification skills following the exact *Bi-LSTM max-out* model from Conneau et al. (2017b) (Figure 4.4), which has been shown to be successful for learning sentence representations.

As relation classification skills we employ Natural Language Inference from SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018).

We also employ a specific relation classification model (Figure 4.5) for Explicit Discourse Relation Sense Classification. In this model, we encode the sentence containing two arguments (*Arg1*, *Arg2*) and the discourse connective (*Conn*) together using a contextual



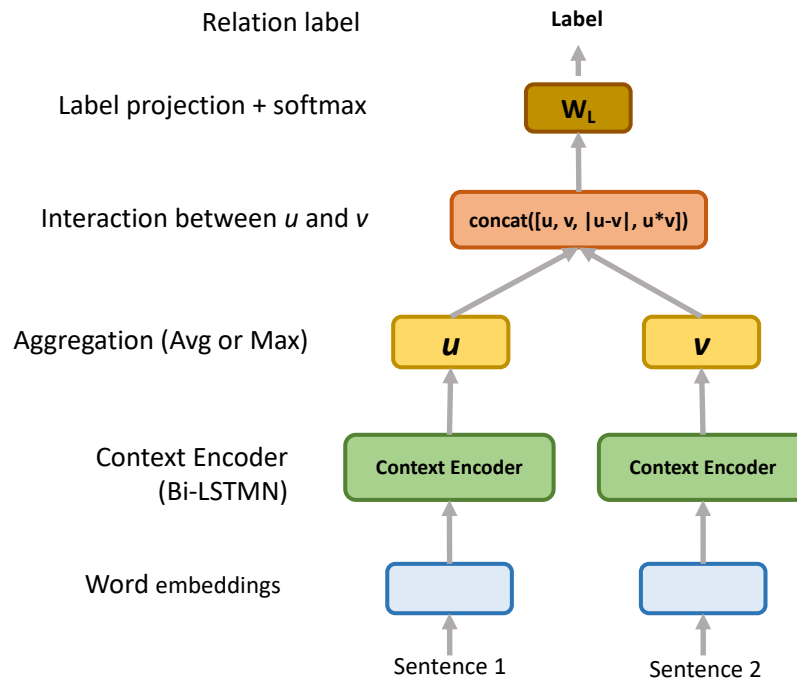


Fig. 4.4 General model for learning representations from relation classification tasks, proposed by (Conneau et al., 2017b)

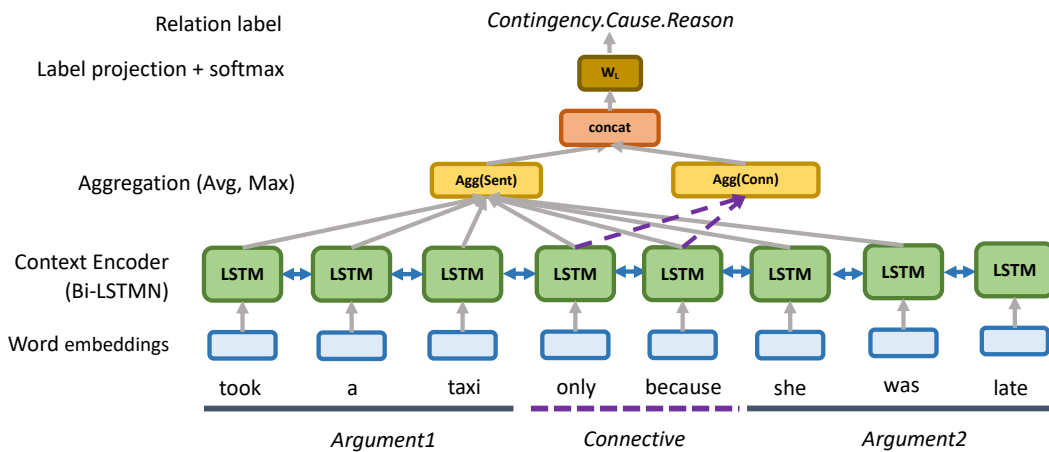


Fig. 4.5 Relation classification using pooled connective tokens for Explicit Discourse Relation Sense Classification.

representation built with *Bi-LSTM*. We then model the relation between the whole sentence and the discourse relation tokens only. This is done by aggregating the token information of i) all tokens and ii) pooled discourse relation tokens. We hypothesize that this is useful for Explicit discourse relations since discourse relation type is greatly dependent on the explicit

discourse connectives (Pitler et al., 2009; Xue et al., 2015). We argue that by combining the discourse connective tokens in the context of the whole sentence at the projection layer, we would help the **context encoder** to encode the main knowledge required for the task.

### 4.2.3 Skillful Reader: Reading Comprehension with Skill Representations

We build a simple neural model that uses pre-trained embeddings and word-matching features as input to a bi-directional LSTM context-encoder of document and question and two feed-forward (FF) layers for predicting the start and end of the answer span.

**Word Embedding Layer** As input to the neural model we use pre-trained 100D Glove (Pennington et al., 2014) word embeddings (WE). We also use two features for each token: the exact word matching feature (em) for words in the document in question (Weissenborn et al., 2017b; Chen et al., 2016b) and the maximum similarity between the word embeddings vector of each of the document tokens and each token in the question:

$$\text{maxsim}(w_i^d, w_{1..m}^q) = \max(\cos(w_i^d, w_{1..m}^q)).$$

We hypothesize that using these features, we would help the model to learn word-matching patterns easily and focus on extracting other more-important phenomena from the natural language tasks. For each token, we concatenate the WE and the two features:

$$w_{p1..N}^r = \text{concat}(w_{e_i}^p, \text{maxsim}, \text{em}),$$

$r$  means input representation,  $p$  is a token sequence that can be  $d$ (document) or  $q$ (question) and use them as input to the context-encoder. For the question, the two features above are set to 1 as in (Weissenborn et al., 2017b).

**Context Encoder** We use a Bi-LSTM context encoder represented as

$$c_{p1..N} = \text{BiLSTM}(w_{p1..N}^r)$$

and refer to it as a task-specific context-encoder  $Enc_{task}$ .

**Context Encoder for the Target Task** For the reading comprehension target task, we initialize an encoder  $Enc_{RC}$  with random weights and fine-tune them during training.

**Skill Task Context Encoders** For each skill task, we train a context-encoder model as described in Section 4.2.2. We use the learned weights to initialize the task-specific encoders  $Enc_{skill}$ . For some of the tasks that employ token label prediction (NER, and Question Type Classification), we also experiment with concatenating the soft label prediction vectors with the context encoder states:

$$Enc_{NER/QTC} = \text{concat}(c_{p1..N}, c_{p1..N}^{lbl})$$

**Adapted Representations** Each output from the skill context encoder is projected to a lower dimension using adapters (Rusu et al., 2016) as follows:

$$c_{task}^{1..n} = Enc_{task}(w_{1..n})A_{task} + b_{task}^a,$$

where  $A_{task}$  is a weight matrix for the current task (skill task or target task (RC)) and  $b_{task}^a$  is a bias vector.

**Ensemble Representation** For each token in the document  $d$  and question  $q$  we concatenate all adapted skill representations  $c_{task}$  to the main task representation  $c_{rc}$  to obtain the ensemble representation:

$$e_p = \text{concat}(c_{rc}, c_{ner}, c_{qtc}, c_{te}, c_{ppdb}),$$

where  $p$  is  $d$  or  $q$ . We represent the question by a weighted representation of its ensemble token vectors:  $r_q = \text{sum}(e_{q1..m} * \text{softmax}(e_{q1..m} W_{qw}))$ , where  $W_{qw}$  is a learnable weight matrix. We then model interaction between the question representation  $r_q$  and each document token  $e_{d_i}$  as

$$r_{d_i2q} = \text{concat}(e_{d_i}, r_q, e_{d_i} * r_q).$$

**Answer Spans Prediction** In our setup, we use extractive reading comprehension and we predict answer spans with their start and end tokens in the document. The probability of an answer span’s start and end tokens is presented as:

$$\begin{aligned} ans_i^{start} &= \text{softmax}(W_{start} FF(r_{d_i2q}) + b_{start}) \\ ans_i^{end} &= \text{softmax}(W_{end} FF(\text{concat}(r_{d_i2q}, ans_i^{start}, ans_i^{start} * e_{d_i})) + b_{end}) \end{aligned}$$

with  $W_{start}$  and  $W_{end}$  being a weight matrix and  $b_{start}$  and  $b_{end}$  bias vectors.

## 4.3 Learning Skill Encoders from Tasks

In this section, we describe the trained models used as skill encoder learners.

We first train our neural encoders with various configurations of the models for Sequence Labeling - (SeqLbl), Text Classification (TC), Text Classification with Token-wise label supervision (TC-TW), Relation Classification (Rel), and Relation Classification with Pooled representation (Rel-P). We ensure that all skill encoders have the same capacity as the trainable parameters. For all skill tasks and the RC task, we use pre-trained Glove word embeddings with size 100. For all tasks, including the target RC task, we train the bi-directional LSTM encoder with output size 256 (2 x 128). For the adaption layer, we use an output size of 100 for the skills and RC.

**Question Type Classification (QTC)** We train two models for question type classification using the QTC dataset by Li and Roth (2002a) in its fine-grained setup using 50 labels. The *QTC* is trained with the default text classification model and *QTC TW* uses Question Type Classification with token-wise label prediction. The assumption here is that the question type classifier can identify tokens in the context that are a good indicator for specific question types and would help to align it with the corresponding question.

**DbPedia Entity Type Classification** Similarly to the Question Type Classification pre-training we train two models for text classification on classifying the type of an entity, given its description. In our experiment *DbPd* is the standard text classification model. *DbPd TW* is trained with entity type classification with token-wise text classification model. We hypothesize that the entity type classifier can identify tokens in the context that are a good indicator for specific entity types and would help in answering specific questions about these types.

**Named Entity Recognition (NER)** We train a model with the standard sequence labeling setup with BIO label encoding as described in the previous section.

**Semantic Role Labeling (SRL)** Similar to NER we train a model in the standard sequence labeling model using BIO label encoding. With this experiment we wanted to introduce information from a semantic task, that would be valuable for finding answers to the questions "Who did what to whom?". Typically the task is evaluated with the F1 score of the full arguments which is calculated after decent prediction performance of the sequence BIO labeling. However, the performance of the trained model is low and the official evaluation

<b>Task Config</b>	<b>Measure</b>	<b>Dev</b>	<b>Test</b>
<b>Question Type Classification</b>			
QTC	Acc	0.8740	-
QTC TW	Acc	0.8620	-
<b>DbPedia Entity Type Classification</b>			
DbPd	Acc	0.9897	-
DbPd TW	Acc	0.9865	-
<b>Discourse Relations</b>			
DR (Exp) Avg	F1	85.43	79.03
DR (Exp) Max	F1	91.05	77.66
DR (Exp) Wiki Avg	F1	81.80	84.27
DR (Exp) Wiki Max	F1	82.31	83.28
DR (Non-Exp) Max	F1	42.86	37.02
DR (Non-Exp) Avg	F1	42.03	38.41
DR (Non-Exp) Wiki Avg	F1	36.05	36.14
DR (Non-Exp) Wiki Max	F1	33.72	36.05
<b>Named Entity Recognition</b>			
NER	Acc/F1	95.13/69.34	95.44/70.67
<b>Semantic Role Labeling</b>			
SRL	Acc - tokens - unofficial	42.00	67.00
<b>Natural Language Inference</b>			
MNLI Max	Acc	64.89	65.30
MNLI Avg	Acc	66.87	66.65
SNLI Max	Acc	80.01	79.48
SNLI Avg	Acc	80.10	78.81
SNLI + MNLI Max	Acc	66.39	67.82
SNLI + MNLI Avg	Acc	66.90	67.63
<b>Paraphrasing</b>			
Para MSR	Acc	73.39	-
Para Quora Q	Acc	85.40	84.52

Table 4.3 Results with different configurations of the source tasks.

produces inconsistent BIO labels so we report the token accuracy.<sup>3</sup> We hypothesize that the results of the trained SRL model are low due to its simplicity and the complexity of the task. For comparison, top-performing models that use LSTM-based architecture (He et al., 2017a) are much deeper and usually have a CRF layer on top together with a set of heuristic rules for closing open BIO tags, etc. In contrast, here we want to examine the performance of one-layer Bi-LSTM models as context encoders for neural transfer and we keep the architecture simple for all tasks.

**Discourse Relation Sense Classification** We train several model configurations for Explicit (DR (Exp)) and Non-Explicit (DR (Non-Exp)) Discourse Relation Sense Classification. We train the models with either average pooling (Avg) or max-pooling (Max) as aggregation strategies over the context representations of the underlying text fragments. The discourse relation sense disambiguation dataset that we use (Xue et al., 2016b) offers data sets in two domains. It has training, validation, and test sets from Penn Discourse Tree Bank (PDTB) (Prasad et al., 2008a) which contain documents from Wall Street Journal (WSJ) and a test set with instances from Wikipedia which is also the source of our target reading comprehension dataset SQuAD (Rajpurkar et al., 2016). For all DR models we use the same training set from the WSJ domain and we use configurations optimized for different validation sets: the original validation set or the Wikipedia evaluation set (Wiki). In column *Test* on Table 4.3 we display the results for the original WSJ test set. In column *Dev* we display the results for the original WSJ validation set or the Wiki set. That is, the models marked with Wiki are trained on WSJ documents, they are validated on the Wiki set and evaluated on the WSJ test set (Test column). The rest of the models are trained on WSJ documents, are validated on the WSJ dev set, and are evaluated on the WSJ test set (Test column). The experiments with models optimized on different domains aim at validating if the target domain is important for our setting. We hypothesize that we could improve the domain transfer from the DR task to RC by only optimizing the model on the target domain.<sup>4</sup> The results give us information about the tasks and the model architectures. The results for the Explicit discourse relations are much higher than those for Non-Explicit. The reason is that the arguments for the Non-Explicit DR are not connected with an explicit discourse connective which is usually a great indicator for the type. We also see that models trained with *Average* pooling versus *Max* pooling have higher test performance whereas the opposite is observed for the development set. This holds

---

<sup>3</sup>The results are oddly round but these are indeed the correct values, rounded to two digits after the decimal point!

<sup>4</sup>These experiments are only available for the DR task since it is the only dataset that provides validation set in the target domain.

both for Explicit and Non-explicit discourse relations and indicates that max-pooling helps to better fit the training data but does not generalize well when in Test.

We also perform experiments with the model trained on the original WSJ domain and validated on the Wiki domain instead. We see that for the best checkpoints on the Wiki validation set, the resulting relative performance on the Test set differs for *Exp* and *Non-Exp*. For the *Exp*, fine-tuning on the Wiki set leads to better results for the WSJ test. This might mean that the WSJ (in-domain) validation set is too similar to the training data and overfitting it would lead to poorer test performance with the LSTM model. We hypothesize that for DR (*Exp*) Wiki, we have learned better transferable representations and we expect them to perform better when evaluated on the RC task. For the Non-Explicit Discourse Relations, the best performance on the Wiki validation sets lead to worse performance on the WSJ test. We argue that the harder task of Non-Explicit discourse relation classification would generally require more in-domain knowledge due to the lack of a discourse connective to characterize the relations.

**Natural Language Inference** For the natural language inference task, we compare models by their training dataset and representation aggregation. The training datasets we use are SNLI (Bowman et al., 2015), MultiNLI (Williams et al., 2018), and their combination (MNLI+SNLI). We hypothesize that the multi-domain dataset would yield better results when transferred to an unseen domain or task. For the model trained on a combination of datasets, we evaluate on the MultiNLI. Similarly to other relation classification tasks, we experiment with average pooling and max-pooling as representation aggregation techniques. From the results in Table 4.3 we can see that getting more data (MNLI+SNLI vs MNLI) helps in learning the NLI skill. The aggregation technique does not seem to have a great impact on the in-domain skill performance.

**Paraphrasing** We also train models on two paraphrasing tasks which are very different from each other. Both are framed as binary relation classification but they have significantly different domains and sizes of the training data.

In this section, we presented the results for several different tasks and models. We further evaluate the transfer performance on the target machine reading comprehension task.

## 4.4 Neural Transfer to Machine Reading Comprehension

Here, we report results on evaluating the pre-trained skill models as part of a model for machine reading comprehension on SQuAD.

### 4.4.1 Training Details

We compare multiple models based on the pre-trained representations transferred from the supervised tasks. Since we want to evaluate the knowledge that is transferred from the pre-training tasks, we fix the ‘skill’ representation encoder when used on the target task. For our target task, we use a base contextual encoder, that is fine-tuned during training and a single ‘skill’ encoder or a combination of ‘skill’ encoders in an ablation setting. For each configuration, we evaluate the importance of the ‘skill’ encoder by comparing it to the same model architecture with randomly initialized parameters. This way we keep the number of learnable parameters fixed for all groups of experiments.

Since our computational resources are limited, we run all experiments with the same fixed hyper-parameters. The parameters are selected by using recommended or widely used parameters from previous work, and they fit an 8G GPU with a batch size of 32 for our task:

- Batch size - 32.
- LSTM hidden size - 128.
- Pre-trained Glove 100 embeddings.
- Embedding dropout rate - 0.2.
- Adapter output size for the transferred representations - 100.

Each setting for the reading comprehension task is evaluated using at least three runs with different random seeds and the reported results are the average of all runs. In the experiments below we also look at the evaluation curves compared to the number of training steps. For each evaluation step, we report the average performance from the same step for multiple runs.

### 4.4.2 Experiments and Results

We evaluate the pre-trained skill representations when used on the SQuAD reading comprehension dataset. Our aims here are to better understand the way the ‘skill’ representations help for learning the reading comprehension task, rather than maximizing the performance on this task. We are comparing the skill encoder with transferred representations to such with random initialization of the Bi-LSTM.



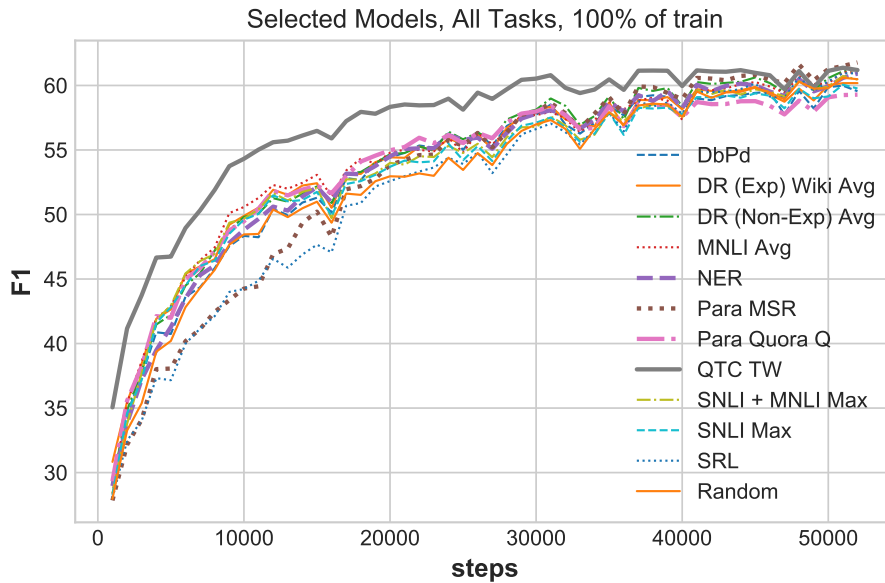


Fig. 4.6 Experiments with selected single representations on the full training data.

#### 4.4.2.1 Overall Results

For our experiments, we are reporting the F1 validation performance for each step of the training. Figure 4.6 shows the training performance of several representations trained on our skill tasks with the reading comprehension model.

In Figure 4.6 we can see that using pre-trained skill representations improves the performance on the reading comprehension task, early in training and the performance gap gets smaller when the models are trained with more iterations. To investigate further the behavior of different tasks we run experiments with different sizes of data and focus on the different stages of training based on the number of training steps.

#### 4.4.2.2 Limited Data and Training Stages

Since using pre-trained representations is helpful early in training we hypothesize that we would also see better gains with these representations when we train with less data.

We run the same experiments with different portions of the training data including sets of 2%, 5%, 10%, 25%, and 100%.<sup>5</sup>

<sup>5</sup>All experiments for the same portion of the data are trained with the same sample. We sample the data by iterating over the data and take every 50th, 20th, 10th, 4th, and all question examples correspondingly. We evaluate on the full validation set of SQuAD.

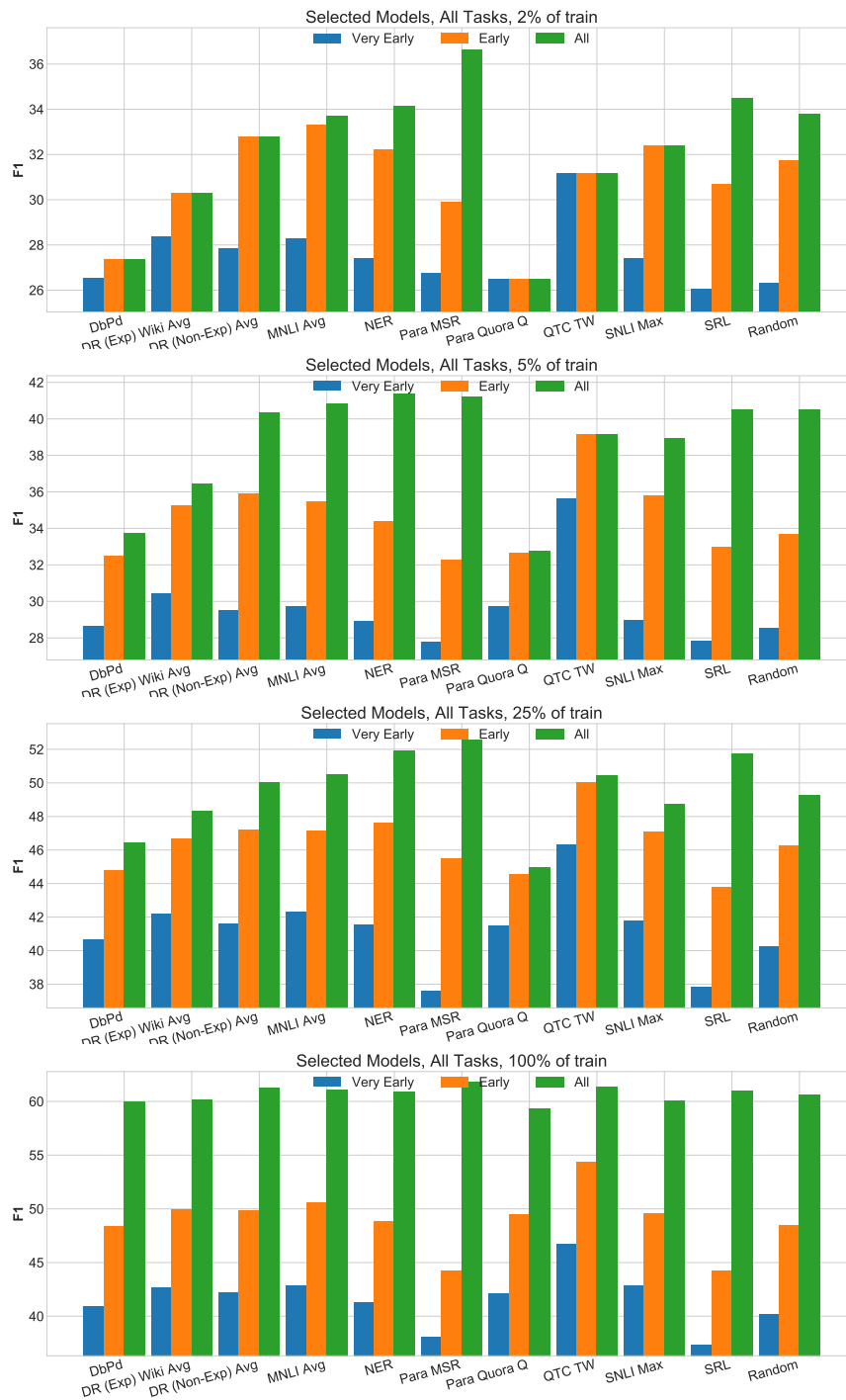


Fig. 4.7 Comparison of different skill representations on SQuAD. Top results by training stage. Training with different sizes of the training data (2%, 5%, 25%, 100%), evaluated on the validation set. *Very early*: 1250 steps for 2%, 5%, 5000 steps for the rest of the sizes, *Early*: < 2500 steps for 2% and 5% and 10000 steps for the rest, *All*: (50000) steps.

We observe (Figure 4.7) changing performance of the transfer depending on the training data size and the training stage. Some skill representations (QTC, NER, NLI) have strong

performance *Very early* in training (1250 steps for 2%, 5% and 5000 steps for the rest of the sizes), still dominate in the whole *Early* stage (< 2500 steps for 2% and 5% and 10000 steps for the rest), whereas they are performing similarly to other tasks when the models are trained with *All* (50000) steps.

The task with the highest positive impact in the *Very early* and *Early* stages is Question Type Classification with Token-wise projections. It has consistent improvements across all data sizes. We argue that this is the case since finding the correct answer requires understanding the question and the specific type of the answer such as named entity type, reason, etc. At the same time, it is also surprising since the training data for the QTC task is relatively small (around 5000 examples).

The next best representations in the *Early* stage are Natural Language Inference models. Representations pre-trained on this task have been shown to perform well across different tasks (Conneau et al., 2017b). The SNLI and MNLI have a high number of training examples (500 thousand) which might be critical for learning good representations.

Another relation classification task that helps in the Early stage with low data sizes (2%, 5%) is the DR (Non-Exp). This model has identical architecture to the NLI pre-trained models. This also suggests that the relation classification architecture is suitable for learning transferable representations.

Explicit Discourse Relation Sense Classification fine-tuned on Wikipedia set is also improving over the baseline for the limited data setups with 2% and 5%. Below, we also perform experiments with different DR models to check if this is due to the task in general, the domain, or the architecture.

These are followed by Named Entity Recognition which was previously shown to be important for machine reading comprehension on SQuAD (Rajpurkar et al., 2016).

Another ‘skill’ that we examine is paraphrasing. The representations trained on Quora Question Paraphrasing detection yield the smallest improvements over *Random* trained with 25% and 100% of the data. In the early stages, Para MSR performs much worse than all models including the baseline but performs best when trained with the full number of steps. We hypothesize that the Para MSR representations learned helpful knowledge about the paraphrasing task, but it is harder to pick up from the reading comprehension model with limited training data.

We see SRL performing in a similar way to Para MSR - it performs poorly in the early stages but well when the model is trained long enough. As we observe in Section 4.3, the model that we use for SRL is probably not deep enough to learn to perform Semantic Role Labeling since this task usually needs a much deeper model (He et al., 2017b). However, we hypothesize that the representations picked relevant implicit semantic information which is

useful for the SQuAD task but is harder for the model to ‘notice’ early in training due to the weak signal.

Above, we compare the impact of different tasks trained with selected architectures, based on the source skill task results. We see that the variance between different models and settings is higher with limited RC data and we will be focusing on the settings with 2% to 25% of the data in the next evaluations.

To better understand the performance of the representations trained on supervised natural language tasks, we further evaluate different model configurations from Section 4.3. We analyze the results for our models grouped by model architecture and then specific tasks, in multiple RC data size settings.

#### 4.4.2.3 Skill Learning Architecture and Modifications

We want to check if the performance of the transferred representations is affected by the model architecture. We hypothesize that if some architecture is crucial, then most tasks trained with the same architecture should have performance or behavior, significantly better than the random initialization. Here we compare the learning architecture by task formulations as training objectives *Text Classification*, *Relation Classification*, *Sequence Labeling*.

**Text Classification** Figure 4.8 displays the results of the experiments with Text Classification architectures (Figure 4.3). We see that the performance for Question Type Classification (*QTC*) and DbPedia Text Classification differ greatly and the results hold for all data sizes. *QTC* models perform much better than the model with Random initialized representations early in training but the baseline performs better in the small data size (2% and 5%) when trained with all steps. The *DbPd* performs slightly better when trained with the aggregation across all tokens (the model in Figure 4.3A) and worse than the baseline when trained with the token-wise objective (the model in Figure 4.3B). In contrast, *QTC TW* (token-wise label-prediction) performs better than the standard *QTC* model. One reason is that enhancing the relevant tokens in the question with type information is more important for finding answer clues than the entity type information learned from the DbPedia Entity classification task. Concatenating the explicit token-wise label prediction logits with the text encoder output (*QTC TW with Lbl*) is beneficial for the small data setup but lacks behind the TW only setting when the data is more than 25%. The different natural language tasks perform very differently in the same setting. We argue that the linguistic tasks used for training and their relevance to the target SQuAD Reading Comprehension task are more important for learning good representations rather than the Text Classification architecture.

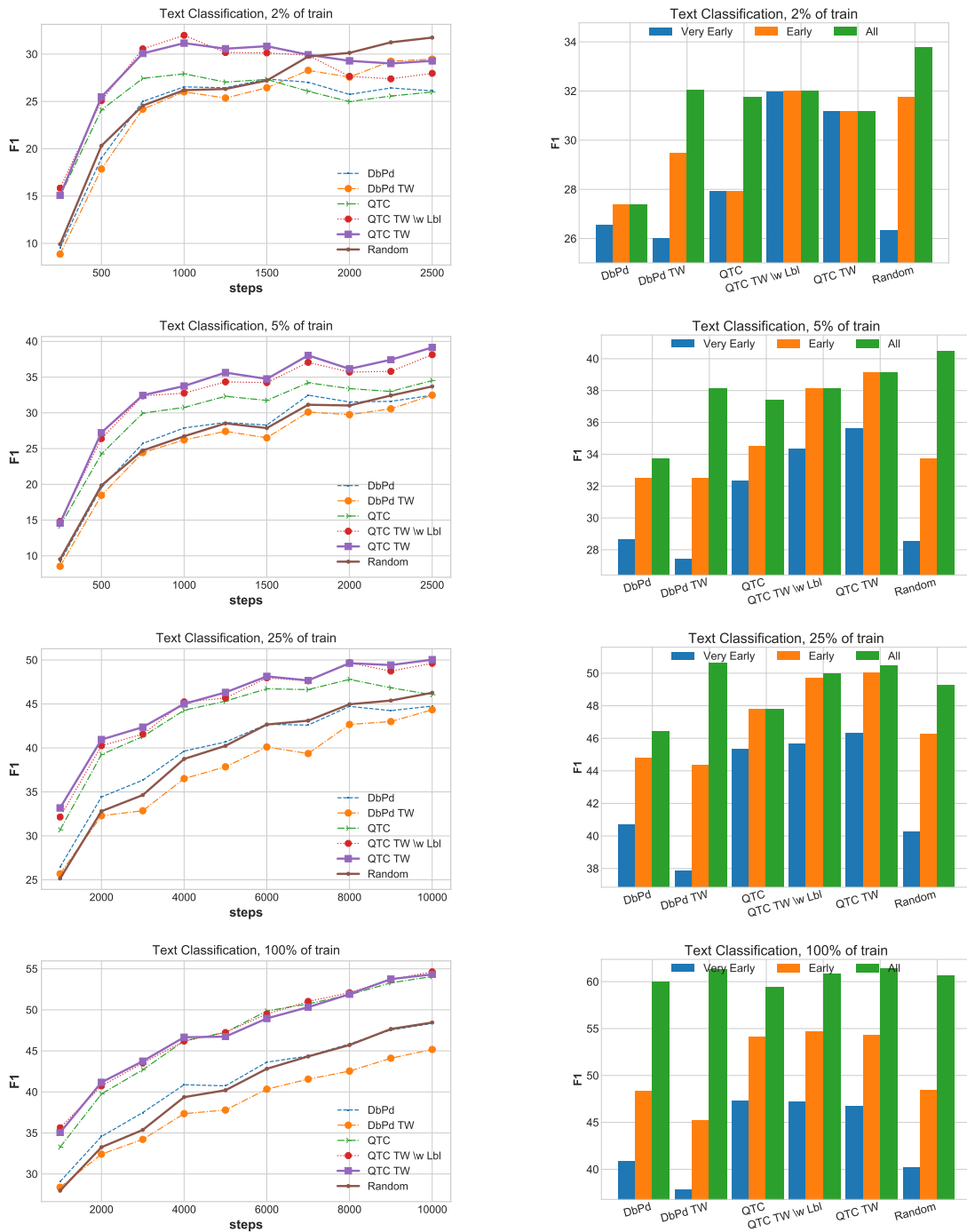


Fig. 4.8 Results on SQuAD for representations learned with the Text Classification architecture. Early (left column) and aggregated performance (right column) training with different sizes of the training data (2%, 5%, 25%, 100%).

**Sequence Labeling** We evaluate two skill tasks that are framed as Sequence Labeling - Named Entity Recognition and Semantic Role Labeling.

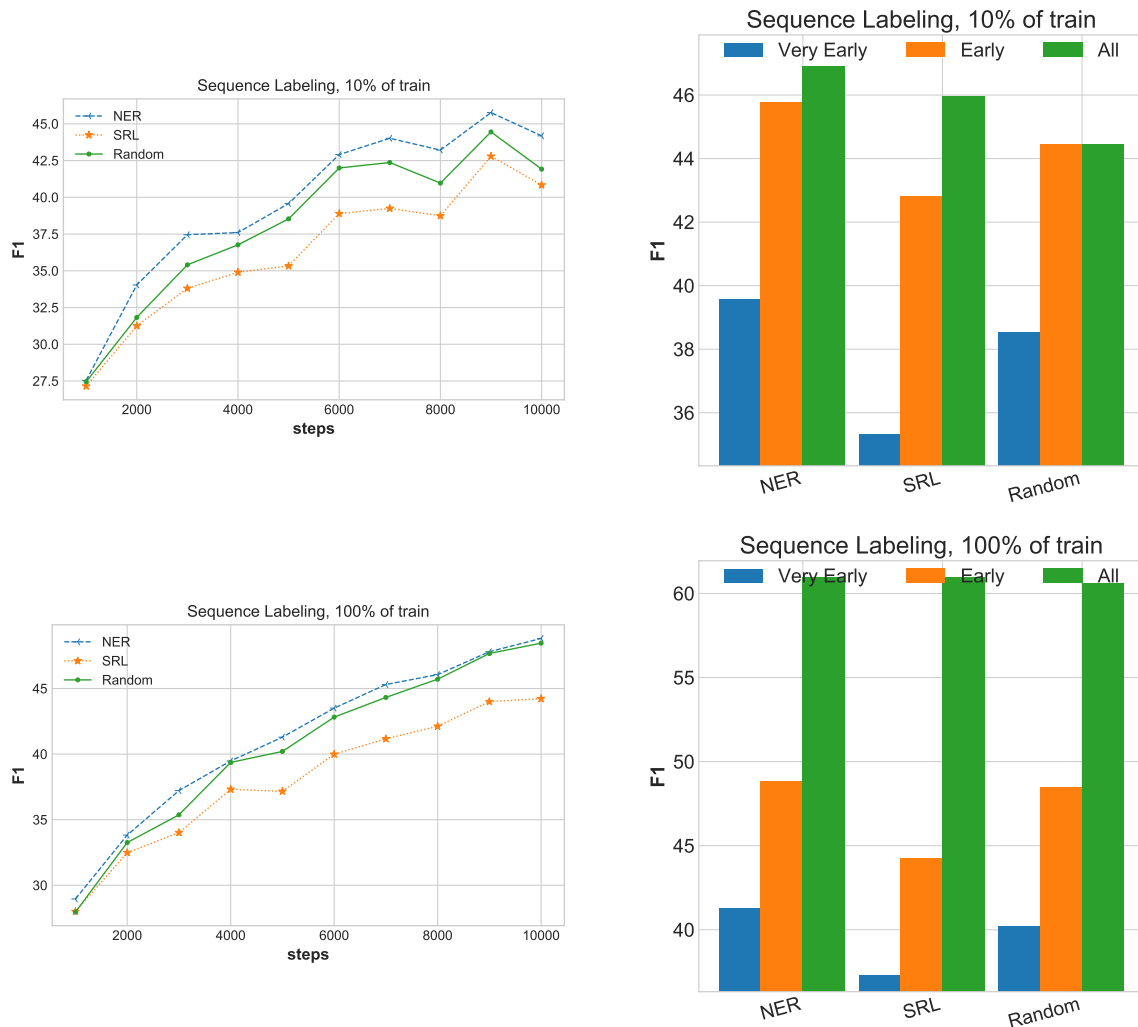


Fig. 4.9 Results on SQuAD for representations learned with the Sequence Labeling architecture. Early (left column) and top (right column) when training with 10% and 100% of the training data are evaluated on the validation set.

In Figure 4.9 we have the representations learned from these tasks to the reading comprehension task on SQuAD. While *NER* expectedly yields good performance, the *SRL* representations are performing worse than the baseline early in training. Both tasks perform similarly in later training steps. We hypothesize that this could be due to the sequence labeling architecture, which is similar to the Reading Comprehension objective that uses token-wise objective and would benefit from contextual token-wise information that the Bi-LSTM provides. Moreover, the performance of the learned representations trained on the sequence labeling tasks compares to the best-performing task of Question Type Classification across all data settings, late in training (Figure 4.6).

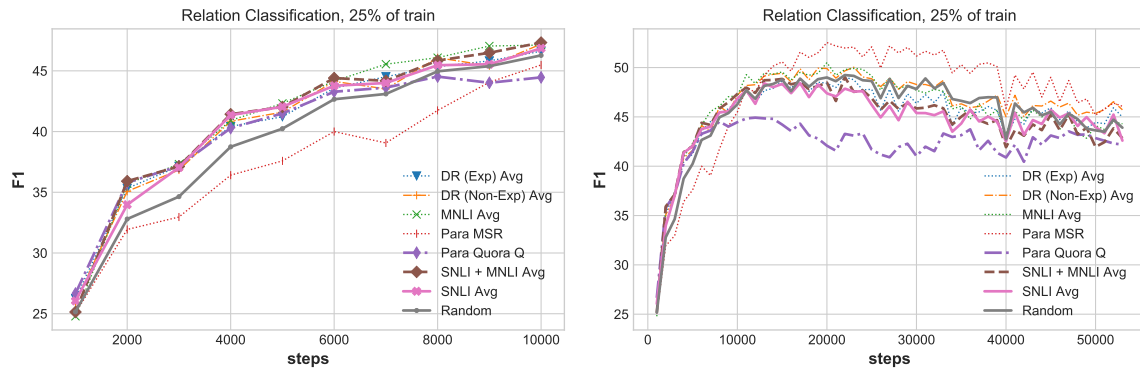


Fig. 4.10 Results on SQuAD for representations learned with the Relation Classification architecture. Early (left column) and all steps (right column) training 25% of data are evaluated on the validation set.

**Relation Classification** We further compare the tasks that are trained as Relation Classification problems. Most of our tasks are trained with this architecture. These include Natural Language Inference trained on SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), and combination of both, paraphrasing detection on sentences (*Para MSR*) and similar question (*Para Quora Q*), as well as Explicit and Non-Explicit Discourse Relation Sense Classification (*DR (Exp)*, *DR (Non-Exp)*). We discussed the performance of these in Section 4.4.2.2. For *DR* and *NLI* tasks, we have different datasets for the same task (SNLI, MNLI), or different representations of the task (Explicit vs Non-Explicit DR), so we look at these aspects in greater detail. We would like to get better insights into the importance of these for obtaining representations that can be transferred to the higher-level task of reading comprehension.

In Figures 4.11 and 4.12 we present RC results for the pre-trained DR and NLI representations, evaluated on SQuAD.

For DR we can compare the representations by the source training model depending on i) Explicit vs Non-Explicit, ii) the in-domain validation fine-tuning (Wiki vs Non-Wiki validation), and iii) representation pooling (Max vs Avg). The source-task results for DR are in the Discourse Relations of Table 4.3. For NLI we can compare the representations by dataset (SNLI vs MNLI vs SNLI+MNLI) and representation pooling (Max vs Avg).

We observe similar performance differences across DR models when the RC data is greater than 2% (Figure 4.11). When we train the model with 100% of the data for the full steps, the performance between different models is very close. The greatest differences are when we compare the 25% data size - they are displayed in Figure 4.11. We observe that Non-Explicit Relations perform better than Explicit.

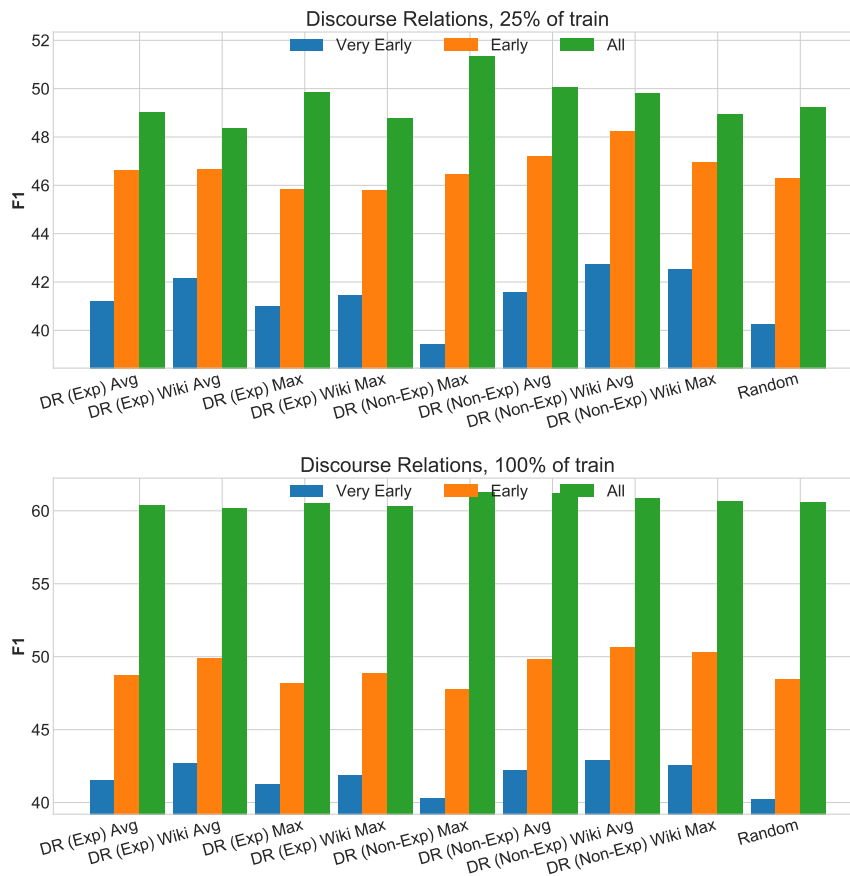


Fig. 4.11 Results on SQuAD for representations learned from Discourse Relation Sense Classification. Models trained with 25% and 100% of the RC data.

In the *Very Early* stage of training, the performance of representations obtained from DR models validated on the Wiki subset of the data performs much better than the models that are validated on the WSJ text. This shows that in the *Very Early* stages the in-domain calibration of the pre-trained representations helps. This also correlates with source task performance on the Test set (Table 4.3) - higher source task performance yields higher performance in *Very Early* transfer. In terms of pooling strategy (Average vs Max), the results for representations trained on DR models with Average pooling yield better results on the test set of the source task, and these results are translated to the performance of the RC task, *Very Early* in training.

For NLI, the representations obtained with Average pooling also yield better results than Max pooling for MNLI and SNLI+MNLI which also correlates with the source task performance (Figure 4.12).

**Combining Multiple Representations** The results of the combination and ablation of multiple skill tasks are shown in Figure 4.13. The combination of features is not very efficient



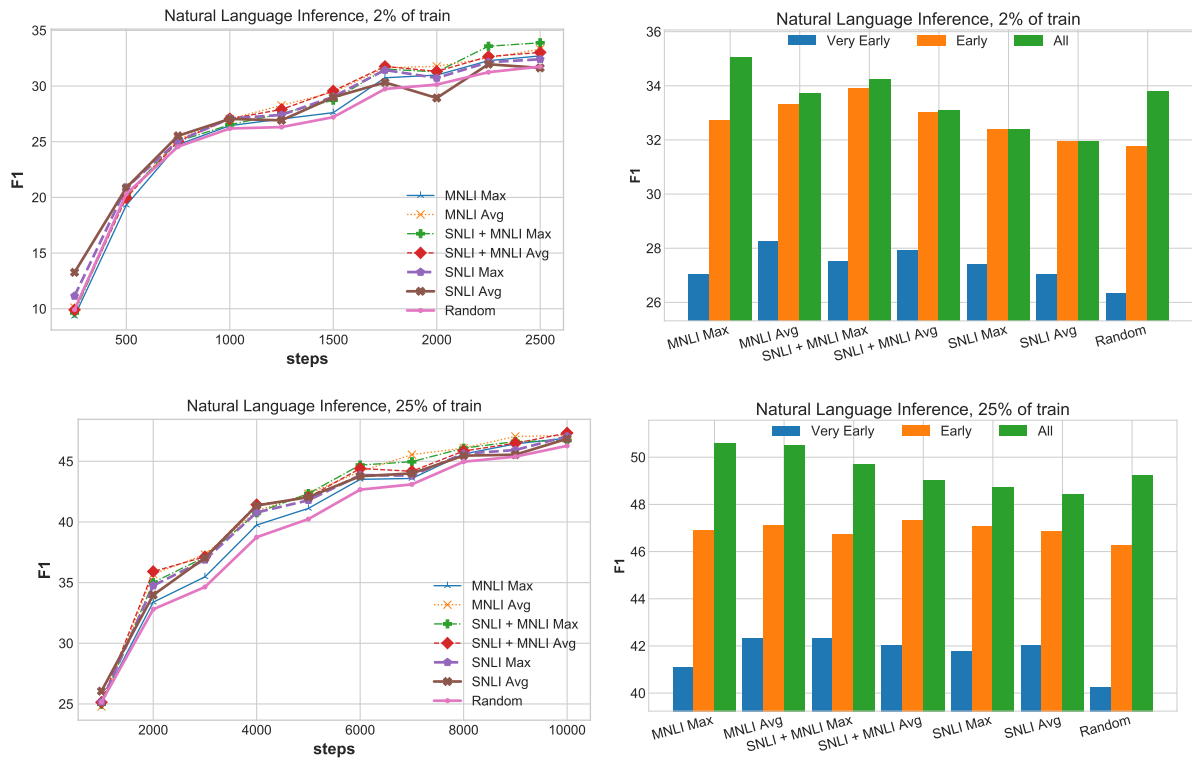


Fig. 4.12 Results on SQuAD for representations learned from Natural Language Inference. Early (left column) and all steps (right column) training with different sizes of the training data (2%, 25% ) are evaluated on the validation set.

since for each token from the text we have to compute the various contextual representations. Therefore, we only run experiments with 25 percent of the data. The bottom part of the figure clearly shows that the combinations of skill representations work very well in the *Very early* and *Early* stages in training. Also, the ablations result in expected gains when the worst performing single representations (Para Quora Q) and drops when the best performing (QTC) is excluded.

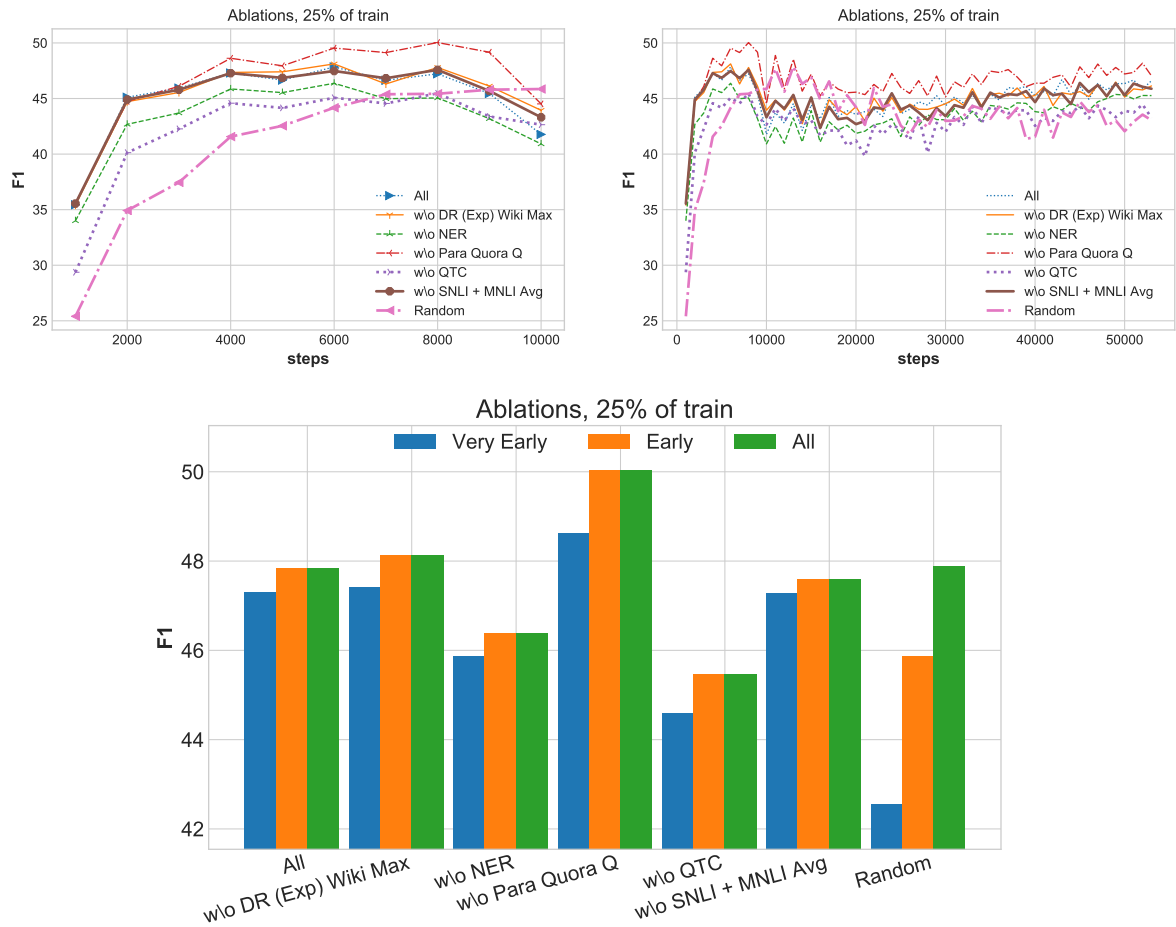


Fig. 4.13 Results for tasks Ablations. Early (left column) and all steps (right column) training with 25% of the training data are evaluated on the validation set.

## 4.5 Discussion

In this work, we experiment with the transfer of linguistic knowledge from supervised human annotated tasks to machine reading comprehension.

**Training Stages** Our results show that there are several tasks (Question Type Classification, Natural Language Inference, Discourse Relations) that help greatly in the *Very early* and *Early stage* of training, or after training for a long time (Paraphrasing, SRL), and others do not help much when transferred to reading comprehension (DbPedia Entity Classification). However, if we train the base model long enough, it often reaches the performance of the models with transferred representations.

**Task Formulation** We observe that the tasks that are formulated as relation classification perform well in both early and full training. The models trained with sequence labeling (NER and SRL) have bad performance in the early training stages and limited data, but are among the best when the model is trained with more steps.

**Training Data Size** We also observed that most models that use supervised pre-trained representations have the highest improvement margin over the model without such knowledge when we use a smaller training size. These results show that we benefit most from linguistic knowledge transfer with limited training data. When the reading comprehension training data is sufficient, the model trained from scratch is sufficient to reach peak performance.

**Source Task Performance vs. Transfer Performance** In our experiments, we see that models that have higher performance on the source task, perform better when transferred to the target task. This correlation can be observed when we compare the improvements for *DR* and *NLI* where we compare different architecture variations (Avg vs. Max pooling) on the same train data. We hypothesize that if we have a better source model the representations would perform better when transferred to the target task. However, this would also make the comparison between tasks harder, since their models will be trained with different architectures.

**Domain** We observe that the domain of the source task is important for the target performance. The performance models trained with the Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2018) dataset performs better than the broad SNLI. The SQuAD dataset that we use is collected from Wikipedia. We observe that DR models obtained with validation on an in-domain Wiki set perform better when the representations are transferred to the RC task.

**Limitations** While our results are positive in some data settings and align with our motivations and assumptions (e.g. question understanding performed via question type classification helps answer questions), we observed that the performance might be affected by the properties of the chosen model architecture and data of the source task.

**Architecture** In our experiments we found that the transfer between the source tasks and the target task is most helpful early in training and when we have limited training data. However, it would be interesting to explore more architectures where we could have improvement of the transfer in the resource-rich setting. We expect that if we have an

architecture that uses more structured representations learned from the source task would work better for the MRC task.

**Evaluation with More Datasets** Due to the computationally intensive training for transfer learning to machine reading comprehension and the scarcity of hardware resources at the time when this work was performed, we transferred representations that are evaluated only on the SQuAD dataset. It would have been better to see how the pre-trained representations work on datasets that require different linguistic tasks.

**Hyper-parameter Search** Another limitation of our setup is that we do not perform extensive hyper-parameter searches for training our representations on our source tasks and the target task. It could be the case that some of the tasks would learn better representations in a specific hyper-parameter setting.

**Fine-tuning** In our setup we evaluate the transferred representations when transferred to the target task without further fine-tuning. While having the parameters fixed, makes it easier to examine the learned knowledge rather than the ability of continuous learning, further fine-tuning we would gain better results in the reading comprehension task.

**Join Training** We hypothesize that it would also be beneficial if we train the source representations, jointly over all tasks and find a setup where they have a positive impact on the MRC task when combined together. Moreover, currently, our setup would require the model to compute several contextual representations using several different task-based encoders, in order to benefit from the multiple source tasks.

**Annotated Task Data vs. Actual Linguistic Knowledge** In our setup, we use the assumptions common to the Computational Linguistics field, that a supervised task dataset, annotated with particular linguistic phenomena reflects this linguistics knowledge. Moreover, having a model that learns to perform well on the test sets of this dataset would learn the linguistic knowledge required to perform this task. After our work was done, the community started to realize that some datasets contain annotation biases (Gururangan et al., 2018), and often the models conveniently learn to pick on these biases (McCoy et al., 2019).

## 4.6 Summary and Conclusions

In this chapter, we proposed an approach to analyzing the impact of injecting linguistic knowledge from supervised language skill tasks into a Machine Reading Comprehension model. In particular, we train simple shallow models that rely on Bi-LSTM contextual representations and transfer the knowledge encoded in these representations to MRC. Our experiments include tasks created around several linguistic phenomena including Named Entity Recognition, Natural Language Inference, Paraphrasing, Semantic Role Labeling, and Discourse Relation, represented by different hand-annotated datasets. We evaluate the models with different data size settings and training phases. We find that representations learned from models trained on these tasks are most beneficial early in training or limited data settings.

In particular, we find that Question Type Classification is the most beneficial in *Very early* and *Early* training stages which suggest that this question understanding skill is very important for the task. The next most important task, early in training is the Natural Language Inference where learning this from a multi-domain dataset (MNLI) would be most beneficial. Representations trained on sentence paraphrasing, named entity recognition, and semantic role labeling are among the best when the RC model is trained with more training steps. We also observe that representations learned with models that perform better on the source task, perform better when transferred to the target task.

# Chapter 5

## Neural Machine Reading Comprehension with Structured Linguistic Knowledge

### 5.1 Introduction

Transformer-based self-attention models (Vaswani et al., 2017) have been shown to work well on many natural language tasks that require large-scale training data, such as Machine Translation (Vaswani et al., 2017; Dai et al., 2019), Language Modeling (Radford et al., 2018a; Devlin et al., 2019b; Dai et al., 2019; Radford et al., 2019) or Reading Comprehension (Yu et al., 2018), and can even be trained to perform surprisingly well in several multi-modal tasks (Kaiser et al., 2017c).

Recent work (Strubell et al., 2018) has shown that for downstream semantic tasks with much smaller datasets, such as Semantic Role Labeling (SRL) (Palmer et al., 2005), self-attention models greatly benefit from the use of linguistic information such as dependency parsing annotations. Motivated by this work, we examine to what extent we can use discourse and semantic information to extend self-attention-based neural models for a higher-level task such as Reading Comprehension.

Many datasets have been proposed for the Reading Comprehension task, starting with a small multi-choice dataset (Richardson et al., 2013), large-scale automatically created cloze-style datasets (Hermann et al., 2015; Hill et al., 2016) and big manually annotated datasets such as Onishi et al. (2016); Rajpurkar et al. (2016); Joshi et al. (2017); Kociský et al. (2017). Previous research has shown that some datasets are not challenging enough, as simple heuristics work well with them (Chen et al., 2016a; Weissenborn et al., 2017b; Chen et al., 2016b). In this work, we focus on the NarrativeQA (Kociský et al., 2017) dataset that

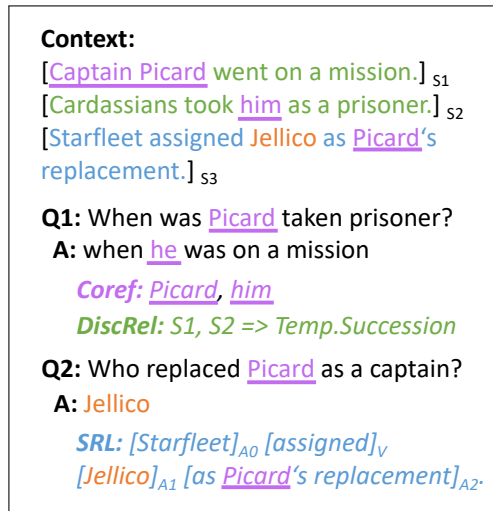


Fig. 5.1 Motivational example: context and questions with required discourse and semantic annotations.

was designed not to be easy to answer and that requires a model to read narrative stories and answer questions about them.

In terms of model architecture, previous work in reading comprehension and question answering has focused on integrating external knowledge (linguistic and/or knowledge-based) into recurrent neural network models using Graph Neural Networks (Song et al., 2018), Graph Convolutional Networks (Sun et al., 2018; De Cao et al., 2019), attention (Das et al., 2017b; Mihaylov and Frank, 2018; Bauer et al., 2018) or pointers to coreferent mentions (Dhingra et al., 2017d).

In contrast, in this chapter, we examine the impact of *discourse-semantic annotations* (Figure 5.1) in a *self-attention architecture*. We build on the QANet (Yu et al., 2018) model by modifying the encoder of its self-attention modeling layer. In particular, we *specialize self-attention heads to focus on specific discourse-semantic annotations*, such as, e.g., an ARG1 relation in SRL, a CAUSATION relation holding between clauses in shallow discourse parsing, or coreference relations holding between entity mentions.

Our contributions are the following:

- To our knowledge we are the first to explicitly introduce discourse information into a neural model for reading comprehension.
- We design a *Discourse-Aware Semantic Self-Attention* mechanism, an extension to the standard self-attention models – without significant increase of computation complexity.

<b>DR (NE)</b>	S-ARG1	S-ARG1	S-ARG1	S-ARG1	S-ARG1	S-ARG1	S-ARG1	.	S-ARG2	S-ARG2	S-ARG2	S-ARG2	S-ARG2	S-ARG2	.
<b>SRL</b>	B-ARG0	I-ARG0	B-V	B-ARG1	B-ARG2	I-ARG2	I-ARG2	.	B-ARG0	B-V	B-ARG1	B-ARG2	I-ARG2	I-ARG2	.
<b>Coref</b>	O	C5	O	C6	O	O	C6	.	C7	O	C8	O	C6	O	.
<b>Tokens</b>	The Cardassians took him as a prisoner .								Starleet assigned Jellico as Picard's replacement .						

Fig. 5.2 Example on different discourse-semantic annotations: DiscRel (Discourse Relations) (NE - Non-Explicit), SRL (Semantic Role Labeling), Coref (Co-reference resolution). The distinct horizontal lines show the interaction between the tokens: Coref - full context, SRL - single sentence, Non-Explicit DR - two neighbouring sentences.

- We analyze the impact of different discourse and semantic annotations for narrative reading comprehension and report improvements of up to 3.4 *Rouge-L* over the base model.
- We perform empirical fine-grained evaluation of the discourse-semantic annotations on specific question types and context size.

## 5.2 Discourse-aware Semantic Annotations

Understanding narrative stories requires the ability to identify events and their participants and to identify how these events are related in discourse (e.g., by *causation*, *contrast*, or *temporal sequence*) (Mani, 2012). We aim to extract structured knowledge about these phenomena from long texts and to integrate this information in a neural self-attention model, in order to examine to what extent such knowledge can enhance the efficiency of a strong reading comprehension model applied to NarrativeQA.

Specifically, we enhance self-attention with knowledge about entity coreference (Coref), their participation in events (SRL), and the relation between events in narrative discourse (Shallow Discourse Parsing (Xue et al., 2016b), DR).

All these linguistic information types are *relational* in nature. For integrating relational knowledge into the self-attention mechanism, we follow a two-step approach: i) we extract such relations from a multi-sentence paragraph and *project them down to the token level*, specifically to the tokens of the text fragments that they involve; ii) we design a neural self-attention model that *uses the interaction information between these tokens in a multi-head self-attention module*.

To be able to map the extracted linguistic knowledge to paragraph tokens, we need annotations that are easy to map to the token level (see Figure 5.2). This can be achieved



with tools for the annotation of span-based Semantic Role Labeling, Coreference Resolution, and Shallow Discourse Parsing.

**Events and Their Participants** Relations between characters in a story are expressed in the text through their participation in states or actions in which they fill a particular event argument with a specific semantic role (see Figure 5.2). For annotation of events and their participants, we use the state-of-the-art SRL system of He et al. (2017b) as implemented in AllenNLP (Gardner et al., 2017). The system splits paragraphs into sentences and tokens, performs POS (part of speech tagging) and for each verb token *V* it predicts semantic tags such as ARG0, ARG1 (Argument Role 0, 1 of verb *V*), etc. When several argument-taking predicates are realized in a sentence, we obtain more than a single semantic argument structure, and each token in the sentence can be involved in the argument structure of more than one verb. We refer to these annotations as different semantic views (Khashabi et al., 2018b), e.g., ‘semantic view for verb 1’. Different self-attention heads will be able to attend to individual semantic views.

**Coreference Resolution** Narrative texts abound of entity mentions that refer to the same entity in the discourse. We hypothesize that by directing the self-attention to this specific coreference information, we can encourage the model to focus on tokens that refer to the same entity mention. Although token-based self-attention models can attend over wide-ranged context spans, we hypothesize that it will be beneficial to allow the model to focus directly on the parts of the text that refer to the same entity. For coreference annotation, we use the *medium* size model from the neuralcoref spaCy extension available at <https://github.com/huggingface/neuralcoref>. For each token we give as information the label of the corresponding coreference cluster (see Figure 5.2) that it belongs to. Therefore, tokens from the same coreference cluster get the same label as input.

**Discourse Relations** In narrative texts, events are connected by discourse relations such as *causation*, *temporal succession*, etc. (Mani, 2012). In this work, we adopt the 15 fine-grained discourse relation sense types from the annotation scheme of the Penn Discourse Tree Bank (PDTB) (Prasad et al., 2008b). For producing discourse relation annotations we use the discourse relation sense disambiguation system from Mihaylov and Frank (2016b) which is trained on the data provided by the CoNLL Shared Task on Shallow Discourse Parsing (Xue et al., 2016b). In this annotation scheme discourse relations are divided into two main types: *Explicit* and *Non-Explicit*. *Explicit* relations are usually connected with an explicit *discourse*

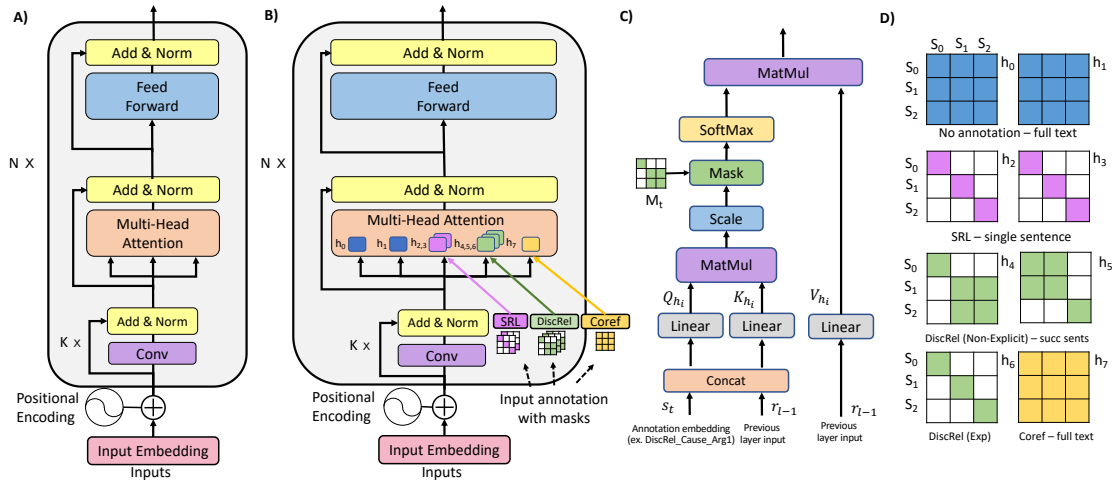


Fig. 5.3 A) Base Multi-Head Self-Attention Encoder Block, B) Discourse-Aware Semantic Self-Attention (DASSA) Encoder Block, C) Single Attention Head with discourse/semantic Information, D) Example of attention scope masks for different attention heads and different information.

*connective*, such as *because*, *but*, *if*. *Non-Explicit*<sup>1</sup> relations are not explicitly marked with a discourse connective and the arguments are usually contained in two consecutive sentences (see Figure 5.2). To extract explicit discourse relations we take into account only arguments that are in the same sentence. We consider as separate arguments (ARG1 and ARG2) text sequences that are on the left and right of an explicit discourse connective (CONN): ex. '[Jeff went home]<sub>ARG1\_CCR</sub> [because]<sub>CONN</sub> [he was hungry.]<sub>ARG2\_CCR</sub>', where CCR is Contingency.Cause.Reason'. To provide *Non-Explicit* discourse relation sense annotations, we annotate every consecutive pair of sentences with a predicted discourse relation sense type.

## 5.3 Discourse-Aware Semantic Self-Attention Model

### 5.3.1 Base Model

As a base reading comprehension model, we use QANet (Yu et al., 2018). QANet is a standard token-based self-attention model with the following components, which are common across many recent models: 1. *Input Embedding Layer* uses pre-trained word embeddings and convolutional character embeddings; 2. *Encoder Layer* consists of stacked *Encoder Blocks* (see Figure 5.3, A) based on *Multi-Head Self-Attention* (Vaswani et al., 2017) and *depth-wise*

<sup>1</sup>*Non-Explicit* relations include *Implicit*, *AltLex* and *EntRel* relation from PDTB. See Xue et al. (2016b) for details.

*separable convolution* (Chollet, 2016; Kaiser et al., 2017a); 3. *Context-to-Query Attention Layer* is a standard layer, that builds a token-wise attention-weighted question-aware context representation; 4. *Modeling Layer* has the same structure as 2. above but uses as input the output of layer 3.; 5. *Output layer* is used for prediction of *start* and *end* answer pointers. For detailed information about these layers, please refer to Yu et al. (2018). In this work we replace the standard *Multi-Head Self-Attention* with *Discourse-Aware Semantic Self-Attention*, using several different semantic and discourse annotation types. We describe this below and explain the differences to the standard *Encoder Block*.

### 5.3.2 Discourse-Aware Semantic Self-Attention

In Figure 5.3 we show the difference between the Base Multi-Head Self-Attention Encoder Block A) and the Discourse-Aware Semantic Self-Attention Encoder Block B). Both consist of positional-encoding + conv-layer  $\times K$  + multi-head-self-attention + feed-forward layer. The difference is that B is provided additional inputs that are used by multi-head self-attention. The multi-head self-attention is a concatenation of outputs from multiple single self-attention heads  $h_i$  followed by a linear layer. A single head of the extended multi-head self-attention is shown in Figure 5.3C and is formally defined as

$$a_{h_i} = \text{mask\_softmax} \left( \frac{Q_{h_i} K_{h_i}^T}{\sqrt{d_h}}, M_t \right) V_{h_i} \quad (5.1)$$

$$Q_{h_i} = W_{h_i}^Q [r_{l-1}; s_t] \in \mathbb{R}^{n \times d_h} \quad (5.2)$$

$$K_{h_i} = W_{h_i}^K [r_{l-1}; s_t] \in \mathbb{R}^{n \times d_h} \quad (5.3)$$

$$V_{h_i} = W_{h_i}^V r_{l-1} \in \mathbb{R}^{n \times d_h}, \quad (5.4)$$

where  $Q_{h_i}$ ,  $K_{h_i}$ ,  $V_{h_i}$  are components of the query-key-value attention and  $\sqrt{d_h}$  is used for weight scaling as originally proposed in Vaswani et al. (2017).  $W_{h_i}^Q$ ,  $W_{h_i}^K$ ,  $W_{h_i}^V$  are weights, specific for head  $h_i$ ,  $i \in 1..H$ <sup>2</sup>,  $r_{l-1}$  is the input from the previous encoder block,  $s_t$  is an embedding vector for the linguistic annotation type  $t$  (‘SRL\_Arg1’, ‘DiscRel\_Cause.Reason\_Arg2’, etc.),  $a_{h_i}$  is the output of head  $h_i$ .  $M_t$  is a sentence-wise attention mask as shown in Figure 5.3D.  $s_t$  and  $M_t$  are the main difference compared to the standard self-attention (Figure 5.3C).

In principle, representing edges of a graph (e.g., the V-ARG1 role from SRL) requires memory of  $n^2 d_h H$ , where  $n$  is the length of the context, which would be a bottleneck for computation on a GPU with limited memory (8-16GB). Instead, we adopt a strategy where the *relation* is represented as a *source and target node* and an *attention scope* (one

<sup>2</sup>Number of heads  $H=8$  as in original QANet specification if not specified otherwise.

sentence for SRL; two sentences for DR (Non-Exp); full context for Coref). The latter is controlled using the attention mask. The combination of flat token labels and mask reduces the maximum memory required for representing the information in the knowledge-enhanced head to  $2nd_hH$ . The attention masks, which we use for reducing the attention scope of the different semantic and discourse annotations, are shown in Figure 5.3D. These masks ensure that the corresponding attention heads will only attend to tokens from the corresponding scope (SRL: single sentence; DR (NonE): two sentences, etc.). The attention masks are symmetric to the matrix diagonal. Therefore, they can easily be computed ‘on-the-fly’ given only the sentence boundaries (corresponding to the horizontal lines in Figure 5.2).

To reduce the model memory further and still benefit from the full-context self-attention, we use the *Discourse-Aware Semantic Self-Attention* encoder (Figure 5.3B) only for blocks [1,3,5] of the *Modeling Layer* that consists of 7 stacked encoder blocks (indexed 0 to 6). Blocks [0,2,6] are set as the base encoders that look at the entire context (Figure 5.3A).

## 5.4 Data and Task Description

**NarrativeQA** We perform experiments with the NarrativeQA (Kociský et al., 2017) reading comprehension dataset. This dataset requires an understanding of narrative stories (English) in order to provide answers to a given question. It offers two sub-tasks: (i) answering questions about a long narrative summary (up to 1150 tokens) of a book or movie, or (ii) answering questions about entire books or movie scripts of lengths up to 110k tokens. We are focusing on the summary setting (i) and refer to the summary as *document* or *context*. The dataset contains 1572 documents in total, divided into Train (1102 docs, 32.7k questions), Dev (115 documents, 3.5k questions), and Test (355 documents, 10.5k questions) sets.

**Generative QA as Span Prediction** An interesting aspect of the NarrativeQA dataset is that in contrast to most other RC datasets, the two answers provided for each question are written by human annotators. Therefore, answers typically differ in form from the context passages that license them. To map the human-generated answers to answer candidate spans from the context, we use *Rouge-L* (Lin, 2004) to calculate a similarity score between token n-grams from the provided answer and token n-grams from candidate answers selected from the context (we select candidate spans of the same length as the given answer). If two answer candidates have the same *Rouge-L* score, we calculate the score between the candidates’ surrounding tokens (window size: 15 tokens to the left and right) and the question tokens and choose the candidate with the higher score. We retrieve the best candidate answer span for

each answer and use the candidate with the higher *Rouge-L* score as supervision for training. We refer to this method for answer retrieval as *Oracle (Ours)*.

## 5.5 Related Work

**Reading Comprehension with Knowledge** Recent work has proposed different approaches for integrating external knowledge into neural models for the high-level downstream tasks of reading comprehension (RC) and question answering (QA). One line of work leverages external knowledge from knowledge bases for RC (Xu et al., 2016; Weissenborn et al., 2017a; Ostermann et al., 2018; Mihaylov and Frank, 2018; Bauer et al., 2018; Wang et al., 2018b) and QA (Das et al., 2017b; Sun et al., 2018; Tandon et al., 2018). These approaches make use of implicit (Weissenborn et al., 2017a) or explicit (Mihaylov and Frank, 2018; Sun et al., 2018; Bauer et al., 2018) attention-based knowledge aggregation or leverage features from knowledge base relations (Wang et al., 2018b). Another line of work builds on linguistic knowledge from downstream tasks, such as coreference resolution (Dhingra et al., 2017d) or notions of co-occurring candidate mentions (De Cao et al., 2019) and OpenIE triples (Khot et al., 2017b) into RNN-based encoders. Recently, several pre-trained language models (Peters et al., 2018; Radford et al., 2018b; Devlin et al., 2019b) have been shown to incrementally boost the performance of well-performing models for several short paragraph reading comprehension tasks (Peters et al., 2018; Devlin et al., 2019b) and question answering (Sun et al., 2019), as well as many tasks from the GLUE benchmark (Wang et al., 2018a). Approaches based on BERT (Devlin et al., 2019b) usually perform best when the weights are fine-tuned for the specific training task. Earlier, many papers that do not use self-attention models or even neural methods have also tried to use semantic parse labels (Yih et al., 2016), or annotations from upstream tasks (Khashabi et al., 2018c).

**Self-Attention Models in NLP** Vanilla self-attention models (Vaswani et al., 2017) use positional encoding, sometimes combined with local convolutions (Yu et al., 2018) to model the token order in the text. Although they are scalable due to their recurrence-free nature, most self-attention models do not well work when trained with fixed-length context, because they often learn global token positions observed during training, rather than relative. To address this issue, Shaw et al. (2018) proposes relative position encoding to model the distance between tokens in the context. Dai et al. (2019) address the problem of moving beyond fixed-length context by adding recurrence to the self-attention model. Dai et al. (2019) argue that the fixed-length segments used for language modeling hurt the performance because they do not respect sentences or any other semantic boundaries. In this work, we

also support the claim that the lack of semantic, and discourse *boundaries* is an issue, and therefore we aim to introduce structured linguistic information into the self-attention model. We hypothesize that the lack of local discourse context is a problem for answering narrative questions, where the answer is contained inside the same sentence, or neighboring sentences and therefore, by offering discourse-level semantic structure to the attention heads, offer ways to restrict, or focus the model to wider or narrower structures, depending on what is needed for specific questions.

Self-attention architectures can be seen as graph architectures (imagine the token (node) interactions as an adjacency matrix) and are applied to graph problems (Veličković et al., 2018; Li et al., 2019). Therefore, in very recent work Koncel-Kedziorski et al. (2019) have used a self-attention encoder as a graph encoder for text generation, in a dual encoder model. A dual-encoder model similar to Koncel-Kedziorski et al. (2019) is suitable for a setting where the input is knowledge from a graph knowledge base. For a text-based setting like ours, where word order is important and the tokens are part of semantic arguments, an approach that tries to encode linguistic information in the same architecture (Strubell et al., 2018) is more appropriate. Therefore our method is most related to LISA (Strubell et al., 2018), which uses joint multi-task learning of POS and Dependency Parsing to inject syntactic information for Semantic Role Labeling. In contrast, we do not use multi-task learning, but directly encode semantic information extracted by pre-processing with existing tools.

**NarrativeQA** The summary setting of the NarrativeQA dataset (Kociský et al., 2017) has in the past been addressed with attention mechanisms by the following models: *BiAtt + MRU* (Tay et al., 2018a) is similar to *BiDAF* (Seo et al., 2017). It is bi-attentive (attends from context to query and vice versa) but enhanced with an MRU (Multi-Range Reasoning Unit). MRU is a compositional encoder that splits the context tokens into ranges (n-grams) of different sizes and combines them in summed n-gram representations and fully-connected layers. *DecaProp* (Tay et al., 2018b) is a neural architecture for reading comprehension, that densely connects all pairwise layers, modeling relationships between passage and query across all hierarchical levels. Bauer et al. (2018) observed that some of the questions require external commonsense knowledge and developed *MHPGM-NOIC* - a *seq2seq* generative model with a copy mechanism that also uses commonsense knowledge and ELMo (Peters et al., 2018) contextual representations. Hu et al. (2018b) used an implementation of Reinforced Mnemonic Reader (*RMR*) (Hu et al., 2018a). They also proposed *RMR + A2D*, a novel teacher-student attention distillation method to train a model to mirror the behavior of the ensemble model *RMR (Ens)*.

Model	B-1	R-L
<b>(Kociský et al., 2017)</b>		
Human	44.43	57.02
Oracle (original)	54.14	59.92
Seq2seq (no context) †	15.89	13.15
ASR †	23.30	22.26
BiDAF	33.72	36.30
<b>Previous work</b>		
BiAtt + MRU (Tay et al., 2018a)	36.55	41.44
DecaProp (Tay et al., 2018b)	44.35	44.69
MHPGM + NOIC (Bauer et al., 2018) †	43.63	44.16
RMR (Hu et al., 2018b)	48.40	51.50
RMR (Ens) (Hu et al., 2018b)	50.10	53.90
RMR + A2D (Hu et al., 2018b)	50.40	53.30
<b>This work</b>		
Oracle (ours)	70.71	70.82
BiDAF	47.19	49.63
QANet	46.37	48.66
+ DR (Exp)	50.12	52.14
+ DR (Exp) EMA	51.16	53.26

Table 5.1 Results on the NarrativeQA Test set. Models with † are generative, while the rest use span prediction.

## 5.6 Experiments and Results

In this section, we describe the experiments and results of our proposed model in different configurations. We compare the results of different models using overall results (Table 5.1) on the dataset, but also the performance for different question types (Figure 5.4) and context sizes (Figure 5.5).

### 5.6.1 Overall Results

Table 5.1 compares our baselines and proposed model to prior work. We report results for *Bleu-1*, and *Rouge-L* scores. The first section lists results on the NarrativeQA dataset as reported in **Kociský et al. (2017)**. *Oracle (original)* uses the gold answers as queries to match a token sequence (with the answer length) in the context that has the highest *Rouge-L*. In contrast, using *Oracle (Ours)*, described in Section 5.4, we report a +11 *Rouge-L* score improvement (Table 5.1: **This work**). The Oracle performance in this setting is important since the produced annotations are used for training of the span-prediction systems, and is

Config	DR-E	DR-NE	SRL	Coref	No
QANet (baseline)	-	-	-	-	8
DR (All)	2	2	-	-	4
DR (Exp)	2	-	-	-	6
DR (NonE)	-	2	-	-	6
Coref	-	-	-	3	5
SRL	-	-	3	-	5
SRL+ DR (Exp)	2	-	3	-	3
SRL + DR (NonE)	-	2	3	-	3
SRL + DR (All)	2	2	3	-	1
SRL + DR (Exp) + Coref	2	-	3	1	2
SRL + DR (All) + Coref	2	2	3	1	4

Table 5.2 The number of attention heads by discourse-semantic type. ‘No’ means that no linguistic annotation types are provided (attends to all tokens).

considered upper-bound.<sup>3</sup> *Seq2Seq (no context)* is an encoder-decoder RNN model trained only on the question. *ASR* is a version of the Attention Sum Reader (Kadlec et al., 2016) implemented as a pointer-generator that reads the question and points to words in the context that are contained in the answer. *BiDAF* is Bi-Directional Attention Flow (Seo et al., 2017) trained either with the *Oracle (original)* or *Oracle (ours)*. The models from **Previous Work** are described in Section 5.5. In the last section of Table 5.1 we present the results of our experiments (**This work**). Here, *BiDAF* and *QANet* are implementations available in the AllenNLP framework (Gardner et al., 2017). In the last two rows we give the results of QANet extended with the proposed Discourse-Aware Semantic Self-Attention, using intra-sentential, *Explicit* discourse relations (*DR (Exp)*, EMA is Exponential Moving Average).

### 5.6.2 Fine-grained Evaluation

We further analyze the performance of different configurations of our model by conducting fine-grained evaluation in view of question types (Figure 5.4) and context length (Figure 5.5).

We define a range of system configurations using attention heads enhanced with different combinations of linguistic annotation types, including *Explicit* (referred to as Exp or E) and *Non-Explicit* (NonE, NE), *Discourse Relations* (DiscRel, DR), *Semantic Role Labeling* (SRL), and *Coreference* (Coref), and configurations without any such additional information (*No*). We also experiment with a setting where instead of using specific discourse relation types

<sup>3</sup>The previous work that uses span-prediction models does not report their *Oracle* model used for training supervision.



	Absolute		Improvement over QANet baseline													
	Oracle	QANet	BiDAF	Coref	DR (All)	DR (Exp)	DR (NonE)	DR (Exp NoSense)	DR (NonE NoSense)	Sent span 3	SRL	SRL + DR (Exp)	SRL + DR (NonE)	SRL + DR (All)	SRL + DR (Exp) + Coref	SRL + DR (All) + Coref
all	70.82	48.66	0.97	1.21	1.34	3.43	0.98	0.56	1.34	1.61	1.53	1.84	1.40	1.31	1.13	1.21
how	53.48	33.00	2.26	0.71	1.00	1.32	1.86	1.11	-0.17	0.53	1.00	0.85	0.44	1.34	0.08	1.96
how far*	65.76	48.62	2.93	8.33	-26.40	10.45	10.45	2.86	10.45	16.70	2.93	10.45	10.45	-22.14	13.67	10.45
how long*	69.92	62.91	2.31	3.68	6.25	3.33	-1.34	-2.25	-0.37	-0.61	-2.32	2.50	4.04	-1.22	4.52	2.01
how many	54.57	46.75	5.36	12.36	2.45	2.54	5.50	2.50	8.63	6.99	3.26	5.42	3.00	5.58	10.15	7.05
how much*	67.59	61.87	5.55	4.48	9.44	12.23	1.91	-4.11	0.09	5.75	3.91	11.52	4.73	7.28	9.43	4.22
how old*	28.62	23.52	-4.13	4.22	-0.03	-9.91	1.05	-13.88	3.48	-0.65	5.22	1.87	1.14	-5.66	3.69	3.92
other*	50.11	18.59	-2.02	7.73	-5.60	-1.53	0.88	6.14	3.44	2.17	7.77	6.74	7.91	2.62	-0.48	3.43
what	69.73	47.88	0.69	0.77	0.40	3.11	0.28	0.71	0.62	0.68	1.24	1.36	1.61	0.97	0.54	0.55
when	64.11	39.66	5.35	4.25	2.05	5.71	4.04	4.88	5.03	5.81	4.80	6.98	2.57	3.25	4.21	6.55
where	80.66	62.75	-1.38	-0.22	-1.08	1.78	-0.01	0.10	0.61	-0.38	0.44	1.15	0.21	0.77	1.19	-0.40
which	85.64	55.99	-2.78	-1.59	0.94	3.44	-2.81	-0.24	-1.20	-0.95	-1.57	3.14	0.91	-1.06	-0.72	1.28
who	82.97	55.97	0.93	1.88	3.49	5.40	1.87	0.17	2.80	3.77	2.30	2.35	1.90	2.25	1.74	1.88
why	49.59	33.51	2.19	0.69	1.31	1.86	1.35	0.06	0.96	1.02	1.54	0.96	-0.15	0.17	0.60	0.31

Fig. 5.4 Rouge-L performance per Question Type on the NarrativeQA Test set. The first two columns represent *Absolute* values. The rest are improvements over the QANet baseline model (i) by BiDAF and (ii) configurations of QANet with linguistic information. Question types with \* have less than 100 instances in the Test set.

(such as DiscRel\_Exp\_Cause\_Arg1), we only identify that a token is a part of **any** (NoSense) discourse relation (e.g., DiscRel\_Exp\_Arg1) or simply a multi-sentence attention span *Sent span 3* with labels *Sent1*, *Sent2*, *Sent3* for each sentence. This is to examine whether the type of discourse relation is important or rather the attention scope (intra-sentential, cross-sentence - 2, 3 neighbouring sentences, full context).

**Question Type** Different question types might profit from different linguistic annotation types. We thus examine the performance of different question types, and analyze how it correlates with the presence of specific Semantic Self-Attention signals. We classify the questions into question types using a simple heuristic based on the question words as an indicator of their type (*How / Where / Why / Who / What ...*), and calculate the average *Rouge-L* for each such question type. The resulting scores are displayed in Figure 5.4. In the first two columns of the figure, we report the *Oracle* score and the baseline (QANet)

	Absolute		Improvement over QANet baseline														
200 - 400	71.54	50.89	-0.12	0.84	1.63	3.04	0.90	1.17	0.33	2.13	1.45	2.29	0.95	0.25	0.50	0.92	
400 - 600	69.69	48.73	1.79	1.28	1.35	3.39	0.26	0.96	1.40	1.11	2.28	1.68	0.98	0.78	1.95	1.33	
600 - 800	70.86	48.72	0.49	1.47	0.87	2.80	0.50	-0.56	1.73	1.23	0.67	1.41	0.81	0.98	0.39	0.62	
800 - 1000	70.54	46.31	1.35	1.42	1.90	4.23	1.64	1.02	1.64	1.73	1.40	1.71	2.07	2.50	1.44	1.79	
1000 - 1200*	73.05	51.66	1.96	0.19	0.33	3.99	2.45	0.48	1.07	2.70	3.60	3.34	3.51	2.16	2.27	1.65	
	Oracle	QANet	BiDAF	Coref	DR (All)	DR (Exp)	DR (NonE)	DR (Exp NoSense)	DR (NonE NoSense)	Sent span 3	SRL	SRL + DR (Exp)	SRL + DR (NonE)	SRL + DR (All)	SRL + DR (Exp) + Coref	SRL + DR (All) + Coref	

Fig. 5.5 Rouge-L performance by context length on the NarrativeQA Test set. The first two columns represent *Absolute* values. The rest are improvements over the QANet baseline model (i) by BiDAF and (ii) different configurations of QANet with linguistic information. Rows with \* have less than 100 instances in the Test set.

score. In the remaining columns we report (i) the improvement over the QANet baseline of BiDAF, and (ii) of our models with different combinations of discourse-aware semantic self-attention. In the first row, we report the score for each of the models on *all* questions. We observe that best-performing models on *all* questions are the ones that include *Explicit* DR, and/or SRL. In terms of hardness, *how* and *why* questions usually have the lowest score. This is not surprising since *Oracle* performance is also low. For these types of questions, the RNN-based encoder (BiDAF) and self-attention with DR (Exp) or DR (NonE) perform best. Almost all models with additional linguistic information improve over the baseline on *when* questions, lead by the SRL+DR (Exp) and SRL + DR (All) + Coref. *What* questions are improved most by DR (Exp) and SRL alone or when combined. *Who* questions gain the most from discourse relations and all models that contain SRL.

**Context Length** In Figure 5.5 we present the performance on documents of different lengths, in number of tokens. All presented models are trained on the examples from the Train set with context up to 800 tokens. Again, the models DR (Exp) and SRL+DR (Exp) show clear improvement across all context lengths. It is clear that all models show improvement over length *800-1000*. This supports our hypothesis that discourse information is required for generalizing to longer contexts. One reason is that some of the questions can be answered with a local context (one-two sentences) which is better represented given short discourse scope (one-three sentences) or long dependencies given coreference.

In the evaluation of multiple model configurations, we notice that in some cases a single discourse/semantic type (e.g. DR (Exp)) performs better than in combination with others (e.g. SRL+DR (Exp)). We hypothesize that the reason is that the linguistic annotations work well in combination with free *No* attention heads (see Table 5.2). Currently, we place multiple

---

**Context** Although he terrifies the fairies when he first arrives , Peter quickly gains favour with them . He amuses them with his human ways and agrees to play the panpipes at the fairy dances . Eventually , Queen Mab grants him the wish of his heart , and he decides to return home to his mother .

**Question** After scaring the fairies, how does Peter win them over ?

---

**Human 1:** he agrees to play the panpipes at all of the fairy dances.; **Human 2:** He amuses them with his human ways and plays the pipes at their dances.; **Oracle:** human ways and agrees to play the panpipes at the fairy dances ; **QANet:** gains favour ; **DR (Exp), DR (NE):** quickly gains favour with them; **Coref, SRL, SRL+DR(Exp):** He amuses them with his human ways and agrees to play the panpipes; **SRL+DR(NE):** He amuses them with his human ways and agrees to play the panpipes at the fairy dances

---

**Rationale:** To find the correct answer we need to know that (i) ‘gains favor’ is a synonym to ‘win’ in this context (commonsense); (ii) the following (2nd) sentence is the reason for the previous (1st) (DR - the model fails in this case) (iii) ‘them’ are ‘the fairies’, ‘he’ is Peter (Coref)

---

Fig. 5.6 Example of positive impact of SRL and Coref and negative impact from discourse relations (DR).

---

**Context** Jacob frequently visits Jeff and Kenny , who are serving time in a juvenile hall . Jacob initially threatens them , until eventually Jeff commits suicide . Jacob befriends Kenny , soon learning he has an early release and is illegally moving to New Mexico .

**Question** Why does Jeff committ suicide ?

---

**Human 1:** Jacob threatened them; **Human 2:** He is threatened by Jacob.; **Oracle:** site which he says is ; **QANet:** Jeff and Kenny , who are serving time in a juvenile hall; **DR (Exp), DR (NE), SRL, SRL+DR(Exp), SRL+DR(NE):** Jacob initially threatens them ; **Coref:** Jacob initially threatens them , until eventually Jeff commits suicide . Jacob befriends Kenny , soon learning he has an early release and is illegally moving to New Mexico

---

**Rationale:** To find the correct answer we need to understand that ‘until eventually’ suggests that the suicide of Jeff is caused by Jacob threatening ‘them’ (DR) and that Jeff is part of ‘them’ (Coref).

---

Fig. 5.7 Example of positive impact of SRL and Coref, and discourse relations (DR).

annotations on the same *Encoder Block* which reduces the number of free attention heads. For instance, for SRL+DR (Exp), each knowledge-enhanced encoder block has 3 SRL + 2 DR (Exp) + 3 No heads. In future work we plan to use different annotation heads per *Encoder Block (EB)*: e.g., *EB0* has 3 SRL + 5 No; *EB1* has 2 DR (Exp) + 6 No; etc.

---

<b>Context</b>	The four orphan children of the house , Edward , Humphrey , Alice and Edith , are believed to have died in the flames . However , they are saved by <b>Jacob Armitage</b> , a local verderer , who hides them in his isolated cottage and disguises them as his grandchildren . Under Armitage 's guidance , the children from an aristocratic lifestyle to that of simple foresters .
<b>Question</b>	Who rescues the children from fire at Arnwood ?
<b>Human 1, Human 2:</b>	<b>Jacob Armitage</b> ; <b>Oracle:</b> <b>Jacob Armitage</b> ; <b>DR (Exp), DR (NE), Coref:</b> <b>Jacob Armitage</b> ; <b>QANet, SRL, SRL+DR(Exp):</b> Pablo; <b>SRL+DR(NE):</b> Patience
<b>Rationale:</b>	To find the correct answer we need to understand at least that 'they' are 'the children' (Coref) and 'who did what to whom' in the context (SRL).

---

Fig. 5.8 Example of positive impact of Coref and DR and negative impact from SRL.

**Success and Failure Examples** In Figures 5.6, 5.7, 5.8 we show examples of context<sup>4</sup> and questions, together with the answers from human annotators and some of the examined models.<sup>5</sup> We provide a hypothetical rationale of what we would need to answer the question.<sup>6</sup>


## 5.7 Conclusion and Future Work

In this chapter, we use linguistic annotations as a basis for a *Discourse-Aware Semantic Self-Attention* encoder that we employ for reading comprehension on narrative texts.

The provided annotations of discourse relations, events, and their arguments as well as coreferring mentions, are using available annotation tools. Our empirical evaluation shows that discourse-semantic annotations combined with self-attention yield significant (+3.43 *Rouge-L*) improvement over QANet's token-based self-attention when applied to NarrativeQA reading comprehension. We analyzed the impact of different semantic annotation types on specific question types and context regions. We find, for instance, that SRL greatly improves *who* and *when* questions, and discourse relations improve also the performance on *why* and *where* questions. While all examined annotation types contribute, particularly strong and constant gains are seen with intra-sentential DR (all context ranges), followed by SRL (short to mid-sized contexts). Coreference shows positive, but weaker impact, mostly in mid-sized contexts.

<sup>4</sup>The part that contains the correct answer.

<sup>5</sup>For easier reading, we color the **gold**, **correct**, and **wrong** answers and underline the mentions of different characters.

<sup>6</sup> The examples are selected from NarrativeQA Test, in such a way, that they depict the strength and weaknesses of the different models, corresponding to the empirical evaluation on Figure 5.4 and they fit in the space limit.

# Chapter 6

## Summary and Conclusions

In this chapter, we summarize the contributions and findings from the previous three chapters. We also discuss some recent work in the field of machine reading comprehension and knowledge integration for natural language processing and discuss possible future directions.

### 6.1 Summary of the Contributions

In this thesis, we focused on approaches that integrate linguistic knowledge and external background and commonsense knowledge to the task of machine reading comprehension.

In Chapter 3 we proposed tackling the task of **neural machine reading comprehension with external commonsense knowledge**. We developed an approach that enhances an existing cloze-style reading comprehension dataset with knowledge retrieved from ConceptNet and encodes it in a neural model to improve the performance on the task. Our model uses an attention-based mechanism to explicitly select relevant knowledge, encoded in a neural memory and fuse it into a contextual representation of a given document and question. Our proposed neural model is interpretable in nature. First, the linear combination of different interactions between text-only and knowledge-enhanced representations allows us to track in which examples the knowledge is used to make a decision. Second, the explicit retrieval from the explicit memory shows what knowledge was selected to improve the representation. We demonstrate the interpretability using a case study and quantitative experiments on the different knowledge-to-text interactions. We evaluated the effectiveness of different sources of external knowledge for cloze-style reading comprehension where the answers are common nouns and named entities, and open book question answering for science questions. We showed that the integration of relevant knowledge is important and improving the performance further would benefit from better retrieval models and relevant knowledge sources.

In Chapter 4 we explore the benefits of **transferring linguistic knowledge from supervised natural language processing tasks to machine reading comprehension using neural representations**. We develop simple models around text classification, sequence labeling, and relation classification and use them for encoding linguistic knowledge from tasks that resemble skills such as identification of discourse relations, paraphrase identification and natural language inference, event detection, and named entity recognition. We compare the performance of these neural representations when adapted for machine reading comprehension in a simple encoder-based ‘skillful’ neural model. We show that using the representations, learned from specialized natural language processing tasks, boosts the performance of the neural reading comprehension model (i) early in training and (ii) when training on smaller portions (2, 5, 10, or 25 percent) of the original training data. We discuss the effectiveness of different skill tasks in different training stages and data sizes. We also perform ablation experiments of neural representations trained with various architectures and show that they improve the machine reading comprehension performance further when combined.

In Chapter 5 we aim to improve existing self-attention models for machine reading comprehension using structured linguistic knowledge. We develop a *Discourse-Aware Semantic Self-Attention* mechanism, an extension to the standard transformer-based self-attention mechanism (Vaswani et al., 2017) without a significant increase in computation complexity. We analyze the impact of different discourse and semantic annotations on narrative reading comprehension. We annotate the raw text of NarrativeQA dataset (Kociský et al., 2017) with available state-of-the-art tools for Semantic Role Labeling (Gardner et al., 2017) and Coreference Resolution. To further annotate our documents with discourse relations we developed a fast and simple method for discourse relation sense disambiguation (Appendix B). We propose evaluating the output of our models using a fine-grained evaluation of specific question types and context size regions. Our experiments with different discourse-semantic annotations show interesting dependence between discourse and semantic types and question types: ex. Semantic Role Labeling (events) improves *who* and *when* questions, intra-sentential Explicit discourse relations improve *why* and *where* questions. We also show that all relations improve the performance of answering questions on longer texts.

## 6.2 Current Trends and Future Directions

In this thesis, we explored approaches to machine reading comprehension and question answering that use external knowledge, retrieved from external sources or transferred from supervised language tasks and annotations. In this section, we will briefly discuss where

similar approaches to those proposed in this thesis were adopted in the current era of large pre-trained models, and what are plausible future directions in knowledge-enhanced neural networks.

**Transfer Learning from Pre-trained Language Models** In this thesis, we employed neural transfer learning of linguistic knowledge from supervised natural language processing tasks (Chapter 4). Using transfer learning became indeed a standard in the field. However, in contrast to our supervised approach of combining knowledge from multiple tasks, unlabeled pre-training from language models turned out to be much more successful.

A big breakthrough in the field of NLP was unleashed by Peters et al. (2018) who developed ELMo – a deep contextualized bi-directional recurrent neural network encoder, pre-trained with an unlabeled language model objective. Using the weights of this pre-trained model in existing task-specific neural models significantly improved the performance of multiple tasks, including machine reading comprehension (Peters et al., 2018). Radford et al. (2018a) trained a large transformer-based generative language model and proposed a method for adapting it to multiple tasks using fine-tuning. The fine-tuning technique was further developed by Devlin et al. (2019a) with the introduction of BERT and became a standard approach for tackling almost any existing NLP task and getting a supreme performance. Fine-tuning the BERT model surpassed the human annotator performance on the SQuAD (Rajpurkar et al., 2016) dataset and many other tasks. The surprising performance of large pre-trained models on natural language processing tasks is argued (Rogers et al., 2020) to be due to its ability to encode background and linguistic knowledge, implicitly learned in the pre-training phase.

While these models perform well with simple fine-tuning to a target task, they were shown to benefit from efficient adapter-based approaches that are similar to ours proposed in Chapter 4. Hounsby et al. (2019) examined using adapters as an efficient approach to consolidating knowledge in different layers of BERT. This was further developed for cross-lingual and multi-task transfer (Pfeiffer et al., 2020b) and Pfeiffer et al. (2020a) even created a framework to easily share adapters for different tasks and languages contributed by different authors. Using adapters and other knowledge extraction techniques and combining the knowledge from multiple tasks is a promising direction when the models become bigger and bigger and the pure fine-tuning approaches require a lot of resources.

**Pre-trained Language Models With External Knowledge** Approaches similar to our proposed in Chapter 3 have been used successfully for augmenting large pre-trained language models with external knowledge retrieved from a knowledge base and textual corpus. Zhang

et al. (2019) built ERNIE, a knowledge-augmented version of BERT (Devlin et al., 2019a) that employs external information into a ‘knowledgeable’ encoder and achieved better results on various tasks. Concurrently, Peters et al. (2019) proposed KnowBert which uses knowledge from WordNet and Wikipedia to augment the contextual representation of a language model and improve its performance for word sense disambiguation, entity typing, and relation extraction tasks. Guu et al. (2020b) implemented the retrieve-and-augment approach as an end-to-end system that jointly queries a densely-encoded version of Wikipedia, combines the result with a text document, and is trained to recover a missing token in the document. The system achieved SOTA performance on several Open QA tasks. (Lewis et al., 2020a) used a similar approach for retrieval-augmented generation. They combine a pre-trained retriever and a pre-trained sequence-to-sequence model and fine-tune them further together end-to-end. They show that the model performs better than other pre-training approaches without external memory on multiple knowledge-intensive NLP tasks and open-domain QA.

While the examples above show the great potential of knowledge-augmented pre-training, they often do not scale well. One limitation is the size of the explicit memory (document index) which does not allow to add knowledge from multiple domains easily at pre-training and adding so slows the retrieval. In contrast, several works have shown that adding more data crawled from the web, and scaling the model size of a generative model or masked language model, increases the performance on downstream tasks significantly: GPT-1 (Radford et al., 2018a) -> BERT (Devlin et al., 2019b) -> RoBERTA (Liu et al., 2019) -> GPT-2 (Radford et al., 2019) -> GPT-3 (Brown et al., 2020a).

**Language Models as Few-Shot Learners** Recently, Brown et al. (2020a) trained GPT-3 - a gigantic generative language model and demonstrated that is able to perform zero and few-shot in-context learning. They prompted the model with a simple task description and zero or few positive examples of the task the model performed surprisingly well on new examples without any parameter updates. To evaluate the generalization of such models Efrat and Levy (2020) proposed using instructions from crowdsourcing tasks and demonstrated the limitation of such models. Mishra et al. (2021) extended this to multiple tasks and demonstrated that these models perform well when the instructions are simpler.

Framing existing tasks as instruction understanding is an exciting direction where integrating external knowledge and knowledge from multiple tasks would be helpful if done at scale or with smart task mapping. One possible approach to include knowledge is to use metadata fields from crawled web pages or knowledge bases such as GDELT (<https://www.gdeltproject.org/>) that already parsed some relevant information from crawled pages. This knowledge can be grounded using schema formatting to achieve instruction-like



input for generative pre-training at scale. The approach of Mishra et al. (2021) is a way of transfer learning through grounding and extending this to multiple tasks or finding a suitable schema is also a promising direction in order to augment large pre-trained models. Depending on the number of datasets and size these can be used as later steps or continuous training to improve the performance.

# List of Figures

1.1	Reading Comprehension requires using external commonsense knowledge and combination of linguistic tasks and background knowledge. . . . .	2
2.1	Example context and multiple multi-choice questions from the MCTest dataset. * indicates the correct answer. . . . .	14
2.2	Example from the Common Noun subset of the cloze-style reading comprehension dataset CBTest. XXXXX is the question placeholder that has to be replaced with the correct choice. . . . .	16
2.3	Example of a context and question from SQuAD 1.0 ((Rajpurkar et al., 2016)) and unanswerable question from SQuAD 2.0 (Rajpurkar et al., 2018) . . . .	17
2.4	Flowchart of general architecture of a MRC/QA system. $D_{1..N}$ are one or more paragraph documents, $Q$ is a question and $C_{1..M}$ are number of candidates in terms of multi-choice answer candidates. The blue modules (solid line) are common for all neural network-based systems. Modules in green (punctuated line) are optional. Modules in orange (punctuated dotted line) are proposed in our contributions. . . . .	23
2.5	Flowchart of common neural reader model of a MRC system. The blue modules (solid line) are common for all neural network-based systems. Modules in orange (punctuated dotted line) are proposed in this work. . . .	24
3.1	Cloze-style reading comprehension with external commonsense knowledge. <b>Task setup</b> shows the official task formulation. <b>Commonsense knowledge</b> is external knowledge that is presented to the model. Connecting lines show some examples of desired attention between context and knowledge: solid - a higher attention score, punctuated - a lower attention score. . . . .	38
3.2	Full architecture of our approach. . . . .	41

3.3	The Knowledgeable Reader combines plain <i>context</i> & <i>enhanced (context + knowledge)</i> representations of <i>D</i> and <i>Q</i> and retrieved knowledge from the explicit memory with the <i>Key-Value</i> approach. . . . .	43
3.4	Encoding the knowledge triple using BiGRU. . . . .	46
3.5	An example for a question with a given set of choices and supporting facts. . . . .	49
3.6	# of items with reversed prediction ( $\pm$ correct) for each combination of ( <i>ctx+kn</i> , <i>ctx</i> ) for <i>Q</i> and <i>D</i> . We report the number of <i>wrong</i> $\rightarrow$ <i>correct</i> (blue) and <i>correct</i> $\rightarrow$ <i>wrong</i> (orange) changes when switching from score w/o knowledge to score w/ knowledge. The best model type is <i>Ensemble</i> . ( <i>Full model w/o D<sub>ctx</sub>, Q<sub>ctx</sub></i> ). . . . .	61
3.7	Interpreting the components of <i>KnReader</i> . Adding knowledge to <i>Q</i> and <i>D</i> increases the score for the correct answer. Results for top 5 candidates are shown. ( <i>Full model, CN data, CN5Sel, Subj/Obj, 50 facts</i> ) . . . . .	64
3.8	<b>Case 1:</b> Interpreting the components of <i>KnReader (Full model)</i> . Adding retrieved knowledge to <i>Q</i> and <i>D</i> helps the model to increase the score for the correct answer. Results for top 5 candidates are shown. (Subj/Obj as key-value memory, 50 facts, CN5Sel) (Item #357) . . . . .	65
3.9	<b>Case 2:</b> Interpreting the components of <i>KnReader (Full model)</i> . Adding retrieved knowledge to <i>Q</i> and <i>D</i> helps the model to increase the score for the correct answer. Results for top 5 candidates are shown. (Subj/Obj as key-value memory, 50 facts, CN5Sel) (Item #52) . . . . .	66
3.10	<b>Case 3:</b> Interpreting the components of <i>KnReader (Full model)</i> . Adding retrieved knowledge to <i>Q</i> and <i>D</i> helps the model to increase the score for the correct answer. Results for top 5 candidates are shown. (Subj/Obj as key-value memory, 50 facts, CN5Sel) (Item #240) . . . . .	67
3.11	<b>Case 4:</b> Interpreting the components of <i>KnReader (Full model)</i> . Adding retrieved knowledge to <i>Q</i> and <i>D</i> confuses the model and decreases the score for the correct answer. Results for top 5 candidates are shown. (Subj/Obj as key-value memory, 50 facts, CN5Sel) (Item #172) . . . . .	68
3.12	<b>Case 5:</b> Interpreting the components of <i>KnReader (Full model)</i> . Adding retrieved knowledge to <i>Q</i> and <i>D</i> confuses the model and decreases the score for the correct answer. Results for top 5 candidates are shown. (Subj/Obj as key-value memory, 50 facts, CN5Sel) (Item #187) . . . . .	69
4.1	Skillful Reader: Architecture for transferring knowledge from ‘skill’ language tasks to an RC model. . . . .	74
4.2	Vanilla Bi-LSTM for sequence labeling (NER) . . . . .	78

4.3	Text classification (Question Type Classification) with Bi-LSTM context encoder and A) sentence-wise label prediction and B) token-wise label supervision. . . . .	79
4.4	General model for learning representations from relation classification tasks, proposed by (Conneau et al., 2017b) . . . . .	80
4.5	Relation classification using pooled connective tokens for Explicit Discourse Relation Sense Classification. . . . .	80
4.6	Experiments with selected single representations on the full training data. . . . .	88
4.7	Comparison of different skill representations on SQuAD. Top results by training stage. Training with different sizes of the training data (2%, 5%, 25%, 100%), evaluated on the validation set. <i>Very early</i> : 1250 steps for 2%, 5%, 5000 steps for the rest of the sizes, <i>Early</i> : < 2500 steps for 2% and 5% and 10000 steps for the rest, <i>All</i> : (50000) steps. . . . .	89
4.8	Results on SQuAD for representations learned with the Text Classification architecture. Early (left column) and aggregated performance (right column) training with different sizes of the training data (2%, 5%, 25%, 100%). . . . .	92
4.9	Results on SQuAD for representations learned with the Sequence Labeling architecture. Early (left column) and top (right column) when training with 10% and 100% of the training data are evaluated on the validation set. . . . .	93
4.10	Results on SQuAD for representations learned with the Relation Classification architecture. Early (left column) and all steps (right column) training 25% of data are evaluated on the validation set. . . . .	94
4.11	Results on SQuAD for representations learned from Discourse Relation Sense Classification. Models trained with 25% and 100% of the RC data. . . . .	95
4.12	Results on SQuAD for representations learned from Natural Language Inference. Early (left column) and all steps (right column) training with different sizes of the training data (2%, 25% ) are evaluated on the validation set. . . . .	96
4.13	Results for tasks Ablations. Early (left column) and all steps (right column) training with 25% of the training data are evaluated on the validation set. . . . .	97
5.1	Motivational example: context and questions with required discourse and semantic annotations. . . . .	102
5.2	Example on different discourse-semantic annotations: DiscRel (Discourse Relations) (NE - Non-Explicit), SRL (Semantic Role Labeling), Coref (Coreference resolution). The distinct horizontal lines show the interaction between the tokens: Coref - full context, SRL - single sentence, Non-Explicit DR - two neighbouring sentences. . . . .	103

5.3	A) Base Multi-Head Self-Attention Encoder Block, B) Discourse-Aware Semantic Self-Attention (DASSA) Encoder Block , C) Single Attention Head with disource/semantic Information, D) Example of attention scope masks for different attention heads and different information. . . . .	105
5.4	Rouge-L performance per Question Type on the NarrativeQA Test set. The first two columns represent <i>Absolute</i> values. The rest are improvements over the QANet baseline model (i) by BiDAF and (ii) configurations of QANet with linguistic information. Question types with * have less than 100 instances in the Test set. . . . .	112
5.5	Rouge-L performance by context length on the NarrativeQA Test set. The first two columns represent <i>Absolute</i> values. The rest are improvements over the QANet baseline model (i) by BiDAF and (ii) different configurations of QANet with linguistic information. Rows with * have less than 100 instances in the Test set. . . . .	113
5.6	Example of positive impact of SRL and Coref and negative impact from discourse relations (DR). . . . .	114
5.7	Example of positive impact of SRL and Coref, and discourse relations (DR). . . . .	114
5.8	Example of positive impact of Coref and DR and negative impact from SRL. . . . .	115
B.1	System architecture: Training and evaluating models for Explicit and Non-Explicit discourse relation sense classification . . . . .	152
B.2	CNN architecture by Kim (2014) (figure is from Kim (2014)) . . . . .	155
B.3	Modified ARC-I CNN architecture for sentence matching. . . . .	156

# List of Tables

2.1	Examples of different natural language inference relations . . . . .	21
3.1	Characteristics of Children Book Test datasets. CN: <i>Common Nouns</i> , NE: <i>Named Entities</i> . Cells for <i>Train</i> , <i>Dev</i> , <i>Test</i> show overall numbers of examples and average story size in tokens. . . . .	39
3.2	Statistics for full OpenBookQA dataset. Parenthetical numbers next to each average are the <i>max</i> . . . . .	40
3.3	Results with different knowledge sources, for CBT-CN (Full model, 50 facts).	55
3.4	Results for CBT (CN) with different numbers of facts. (Full model, CN5Sel)	55
3.5	Results for different combinations of interactions between document (D) and question (Q) <i>context (ctx)</i> and <i>context + knowledge (ctx+kn)</i> representations. (CN5Sel, 50 facts) . . . . .	56
3.6	Results for key-value knowledge retrieval and integration. (CN5Sel, 50 facts). <i>Subj/Obj</i> means: we attend over the fact subject (Key) and take the weighted fact object as value (Value). . . . .	56
3.7	Comparison of KnReader to existing end-to-end neural models on the benchmark datasets. . . . .	57
3.8	Comparison of <i>KnReader</i> to existing ensemble models and models that use re-ranking. . . . .	57
3.9	Scores obtained by various solvers on OpenBookQA, reported as a percentage $\pm$ the standard deviation across 5 runs with different random seeds. Other baselines are described in the corresponding referenced section. For oracle evaluation, we use the gold science fact $f$ associated with each question, and optionally the additional fact $k$ provided by the question author. Bold denotes the best Test score in each category. . . . .	58
3.10	Sample <b>questions</b> predicted <b>correctly</b> (172/500) by all trained neural models without external knowledge. . . . .	70

3.11	Sample <b>questions</b> predicted <b>correctly</b> by the $f + k$ Oracle model (405/500) but were predicted <b>incorrectly</b> by all of the 4 neural models without knowledge (total of 69 out of 405). . . . .	70
3.12	Sample <b>questions</b> predicted <b>incorrectly</b> by all models w/o knowledge, as well as the $f + k$ Oracle model, even though the Oracle model has confidence higher than 0.90. . . . .	71
4.1	Skill tasks, datasets, and model types used in the experiments . . . . .	75
4.2	Skill datasets and the size of training, evaluation sets (number of examples), and vocabulary size (number of tokens). . . . .	75
4.3	Results with different configurations of the source tasks. . . . .	84
5.1	Results on the NarrativeQA Test set. Models with † are generative, while the rest use span prediction. . . . .	110
5.2	The number of attention heads by discourse-semantic type. ‘No’ means that no linguistic annotation types are provided (attends to all tokens). . . . .	111
B.1	Evaluation of our official submission system, trained on Train 2016 and evaluated on Dev, Test, and Blind sets. Comparison with our official system and our improved system with the official results of CoNLL 2015 Shared Task’s best system (Wang and Lan, 2015) and CoNLL 2016 Shared Task best systems in <i>Explicit</i> (Jain, 2016) and Non-Explicit (Rutherford and Xue, 2016). F-Score is presented. . . . .	157
B.2	Evaluation of different systems and feature configurations for Non-Explicit relation sense classification, trained on Train 2016 and evaluated on Dev. F-score is presented. . . . .	158

# References

2002. The effect of prior knowledge on understanding from text: Evidence from primed recognition. *European Journal of Cognitive Psychology*, 14(2):267–286.
- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from [tensorflow.org](http://tensorflow.org).
- Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. In *CoRR*, volume abs/1608.00318.
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07*, page 722–735, Berlin, Heidelberg. Springer-Verlag.
- Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*.
- Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium.
- Yoshua Bengio. 2011. Deep Learning of Representations for Unsupervised and Transfer Learning. *JMLR: Workshop and Conference Proceedings* 7, 7:1–20.
- Nikita Bhutani, H V Jagadish, and Dragomir Radev. 2016. Nested propositions in open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 55–64, Austin, Texas. Association for Computational Linguistics.
- Ellen Bialystok. 1988. Aspects of linguistic awareness in reading comprehension. *Applied Psycholinguistics*, 9(2):123–139.



- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc.
- John R. Bormuth. 1967. Comparable cloze and multiple-choice comprehension test scores. *Journal of Reading*, 10(5):291–299.
- John R. Bormuth. 1968a. The cloze readability procedure. *Elementary English*, 45(4):429–436.
- John R. Bormuth. 1968b. Cloze test readability: Criterion reference scores. *Journal of Educational Measurement*, 5(3):189–196.
- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *Proc. 2015 Conf. Empir. Methods Nat. Lang. Process. Port. 17-21 Sept. 2015*, (September):632–642.
- Chloé Braud and Pascal Denis. 2015. Comparing word representations for implicit discourse relation classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2201–2211, Lisbon, Portugal. Association for Computational Linguistics.
- Eric Breck, Marc Light, Gideon Mann, Ellen Riloff, Brianne Brown, and Pranav Anand. 2001. Looking under the hood: Tools for diagnosing your question answering engine. In *Proceedings of the ACL 2001 Workshop on Open-Domain Question Answering*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- 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. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165.
- Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583, Prague, Czech Republic. Association for Computational Linguistics.
- Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

- Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016a. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2016b. Reading Wikipedia to Answer Open-Domain Questions.
- Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *ACL*, pages 1657–1668.
- Wenhu Chen, Pat Verga, Michiel de Jong, John Wieting, and William W. Cohen. 2023. Augmenting pre-trained language models with QA-memory for open-domain question answering. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1597–1610, Dubrovnik, Croatia. Association for Computational Linguistics.
- Christian Chiarcos and Niko Schenk. 2015. A minimalist approach to shallow discourse parsing and implicit relation recognition. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 42–49, Beijing, China. Association for Computational Linguistics.
- Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.
- François Chollet. 2016. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357.
- Zewei Chu, Hai Wang, Kevin Gimpel, and David McAllester. 2016. Broad Context Language Modeling as Reading Comprehension. *Arxiv*.
- Junyoung Chung, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv e-prints*, abs/1412.3555. Presented at the Deep Learning workshop at NIPS2014.
- Yu-An Chung, Hung-Yi Lee, and James Glass. 2018. Supervised and unsupervised transfer learning for question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1585–1594, New Orleans, Louisiana. Association for Computational Linguistics.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

- Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter D Turney, and Daniel Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *AAAI*, pages 2580–2586.
- Allan Collins and Edward E. Smith. 1980. Teaching the process of reading comprehension. *Bolt Beranek and Newman Inc.*
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing. *Proceedings of the 25th international conference on Machine learning - ICML '08*, 20(1):160–167.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017a. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*, pages 670–680.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017b. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data.
- Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 593–602. Association for Computational Linguistics.
- Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy.
- Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017a. Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–365, Vancouver, Canada. Association for Computational Linguistics.
- Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017b. Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–365.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. 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 2306–2317, Minneapolis, Minnesota.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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.
- Bhuwan Dhingra, Hanxiao Liu, R. Salakhutdinov, and William W. Cohen. 2017a. A comparative study of word embeddings for reading comprehension. *ArXiv*, abs/1703.00993.
- Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017b. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846. Association for Computational Linguistics.
- Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017c. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846. Association for Computational Linguistics.
- Bhuwan Dhingra, Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2017d. Linguistic Knowledge as Memory for Recurrent Neural Networks. 1997.
- William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.
- Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation By Jointly Learning To Align and Translate. *Iclr 2015*, pages 1–15.
- Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? *ArXiv*, abs/2010.11982.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2021. Augmenting Transformers with KNN-Based Composite Memory for Dialog. *Transactions of the Association for Computational Linguistics*, 9:82–99.

- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874.
- Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.
- Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640.
- David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. 2017. Two-stage synthesis networks for transfer learning in machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 835–844, Copenhagen, Denmark. Association for Computational Linguistics.
- Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. *Acl*, page 11.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020a. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020b. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.
- Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks. pages 1–17.
- Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017a. Deep semantic role labeling: What works and what’s next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics.
- Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017b. Deep semantic role labeling: What works and what’s next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 473–483, Vancouver, Canada.
- Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. 2017c. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 199–208. Association for Computational Linguistics.

- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.
- Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children’s books with explicit memory representations. In *ICLR*.
- E. D. Hirsch. 2003. Reading comprehension requires knowledge of words and the world: Scientific insights into the fourth-grade slump and the nation’s stagnant comprehension scores. *American Educator*, pages 10–20.
- Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. 1999. Deep read: A reading comprehension system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 325–332, College Park, Maryland, USA. Association for Computational Linguistics.
- Jerry R Hobbs. 1978. Resolving pronoun references. *Lingua*, 44(4):311–338.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- N. Houlsby, A. Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and S. Gelly. 2019. Parameter-efficient transfer learning for nlp. In *ICML*.
- Jeremy Howard and Sebastian Ruder. 2018a. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.
- Jeremy Howard and Sebastian Ruder. 2018b. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.
- Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2015. Convolutional neural network architectures for matching natural language sentences. *CoRR*, abs/1503.03244.
- Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou. 2018a. Reinforced mnemonic reader for machine reading comprehension. In *Proceedings of International Joint Conferences on Artificial Intelligence Organization*, pages 4099–4106.
- Minghao Hu, Yuxing Peng, Furu Wei, Zhen Huang, Dongsheng Li, Nan Yang, and Ming Zhou. 2018b. Attention-guided answer distillation for machine reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2077–2086, Brussels, Belgium.
- Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics (ACL) 2017.
- Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. From particular to general: A preliminary case study of transfer learning in reading comprehension. In *Machine Intelligence Workshop at NIPS 2016*.
- Rudolf Kadlec, Martin Schmid, Ondřej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 908–918. Association for Computational Linguistics.
- Lukasz Kaiser, Aidan N. Gomez, and François Chollet. 2017a. Depthwise separable convolutions for neural machine translation. *CoRR*, abs/1706.03059.
- Lukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017b. One model to learn them all. *CoRR*, abs/1706.05137.
- Lukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017c. One model to learn them all. *CoRR*, abs/1706.05137.
- Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018a. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL*.
- Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Peter Clark, Oren Etzioni, and Dan Roth. 2016. Question answering via integer programming over semi-structured knowledge. In *IJCAI*.
- Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018b. Question Answering as Global Reasoning over Semantic Abstractions. In *AAAI*.
- Daniel Khashabi, Tushar Khot, Ashutosh Sabharwal, and Dan Roth. 2018c. Question answering as global reasoning over semantic abstractions. In *AAAI*.
- Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, P. Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings of EMNLP*.
- Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017a. Answering complex questions using open information extraction. In *ACL*.
- Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017b. Answering complex questions using open information extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 311–316.
- Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTail: A textual entailment dataset from science question answering. In *AAAI*.

- Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339, Austin, Texas. Association for Computational Linguistics.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations 2015*, pages 1–15.
- Sosuke Kobayashi, Ran Tian, Naoaki Okazaki, and Kentaro Inui. 2016. Dynamic Entity Representation with Max-pooling Improves Machine Reading. pages 850–855.
- Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2017. The NarrativeQA reading comprehension challenge. *CoRR*, abs/1712.07040.
- Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. pages 2284–2293.
- Fang Kong, Sheng Li, and Guodong Zhou. 2015. The sonlp-dp system in the conll-2015 shared task. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 32–36, Beijing, China. Association for Computational Linguistics.
- Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1378–1387, New York, New York, USA. PMLR.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.
- Sobha Lalitha Devi, Sindhuja Gopalan, Lakshmi S, Pattabhi RK Rao, Vijay Sundar Ram, and Malarkodi C.S. 2015. A hybrid discourse relation parser in conll 2015. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 50–55, Beijing, China. Association for Computational Linguistics.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. pages 260–270.



- Guillaume Lample, Alexandre Sablayrolles, Marc' Aurelio Ranzato, Ludovic Denoyer, and Herve Jegou. 2019. Large memory layers with product keys. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. 2015. Deeply-supervised nets. pages 562–570.
- Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020a. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Xin Li and Dan Roth. 2002a. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.
- Xin Li and Dan Roth. 2002b. Learning question classifiers.
- Yuan Li, Xiaodan Liang, Zhiting Hu, Yinbo Chen, and Eric P. Xing. 2019. Graph transformer.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain.
- Wang Ling, Chris Dyer, A. Black, I. Trancoso, Ramón Fernández Astudillo, Silvio Amir, Luís Marujo, and T. Luís. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *EMNLP*.
- Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzell. 2015. Learning to diagnose with lstm recurrent neural networks.
- L. Liu, J. Shang, F. Xu, X. Ren, H. Gui, J. Peng, and J. Han. 2018. Empower Sequence Labeling with Task-Aware Neural Language Model. In *AAAI*.
- Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.
- Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. 2016. Implicit discourse relation classification via multi-task neural networks. *CoRR*, abs/1603.02776.

- Teng Long, Emmanuel Bengio, Ryan Lowe, Jackie Chi Kit Cheung, and Doina Precup. 2017. World knowledge for reading comprehension: Rare entity prediction with hierarchical lstms using external descriptions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 825–834, Copenhagen, Denmark. Association for Computational Linguistics.
- Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *ArXiv*, abs/1604.00788.
- Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnn-crf. pages 1064–1074.
- Inderjeet Mani. 2012. *Computational Modeling of Narrative*, volume 5.
- Christopher D. Manning, M. Surdeanu, John Bauer, J. Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL*.
- Ana Marasović and Anette Frank. 2016. Multilingual Modal Sense Classification using a Convolutional Neural Network. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, Berlin, Germany.
- Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 368–375, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.
- Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *CoRR*, abs/1708.00107.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.
- Todor Mihaylov, Daniel Balchev, Yassen Kiprov, Ivan Koychev, and Preslav Nakov. 2017. Large-scale goodness polarity lexicons for community question answering. In *In Proceedings of the 40th International Conference on Research and Development in Information Retrieval*.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2381–2391, Brussels, Belgium.

- Todor Mihaylov and Anette Frank. 2016a. Discourse relation sense classification using cross-argument semantic similarity based on word embeddings. In *CoNLL-16 shared task*, pages 100–107.
- Todor Mihaylov and Anette Frank. 2016b. Discourse relation sense classification using cross-argument semantic similarity based on word embeddings. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task*.
- Todor Mihaylov and Anette Frank. 2017a. Story cloze ending selection baselines and data examination. In *Proceedings of the Second Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics – Shared Task*, Valencia, Spain. Association for Computational Linguistics.
- Todor Mihaylov and Anette Frank. 2017b. Story Cloze Ending Selection Baselines and Data Examination. In *LSDSem – Shared Task*.
- Todor Mihaylov and Anette Frank. 2018. Knowledgeable Reader: Enhancing Cloze-Style Reading Comprehension with External Commonsense Knowledge. In *ACL*, pages 821–832.
- Todor Mihaylov and Anette Frank. 2019. Discourse-Aware Semantic Self-Attention For Narrative Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*.
- Todor Mihaylov and Preslav Nakov. 2016. SemanticZ at SemEval-2016 Task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings. In *SemEval '16*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT '13*, pages 746–751, Atlanta, Georgia, USA.
- Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to wordnet: An on-line lexical database\*. In *International Journal of Lexicography*, volume 3, pages 235–244.

- Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. 2017. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517, Vancouver, Canada. Association for Computational Linguistics.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. (August):1003.
- Bhavana Dalvi Mishra, Lifu Huang, Niket Tandon, Wen tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: A challenge dataset and models for process paragraph comprehension. In *NAACL*.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Natural Instructions: Benchmarking Generalization to New Tasks from Natural Language Instructions.
- T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.
- Tsendsuren Munkhdalai and Hong Yu. 2016. Reasoning with Memory Augmented Neural Networks for Language Comprehension. In *International Conference on Learning Representations (ICLR) 2017*.
- Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alha Freihat, Jim Glass, and Bilal Randeree. 2016a. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval '16*, San Diego, California, USA.
- Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016b. Semeval-2016 task 3: Community question answering. In *SemEval '16*, pages 525–545.
- Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms Marco: a Human Generated Machine Reading Comprehension Dataset. (Nips):1–10.
- Franz Josef Och. 1999. An efficient method for determining bilingual word classes. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, Bergen, Norway. Association for Computational Linguistics.
- Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, Austin, Texas. Association for Computational Linguistics.

- Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. Mscript: A novel dataset for assessing machine comprehension using script knowledge.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- 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. In *NIPS-W*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.
- P David Pearson, Jane Hansen, and Christine Gordon. 1979. The effect of background knowledge on young children’s comprehension of explicit and implicit information. *Journal of Reading Behavior*, 11(3):201–209.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.
- Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765. Association for Computational Linguistics.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.
- Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.
- Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *EMNLP*.
- Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 683–691, Suntec, Singapore. Association for Computational Linguistics.
- Emily Pitler, Mridhula Raghupathy, Hena Mehta, Ani Nenkova, Alan Lee, and Aravind Joshi. 2008. Easily identifiable discourse relations. In *Coling 2008: Companion volume: Posters*, pages 87–90, Manchester, UK. Coling 2008 Organizing Committee.
- Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. 2014. Improving the Reproducibility of PAN’s Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)*, pages 268–299, Berlin Heidelberg New York. Springer.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. pages 1–40. Association for Computational Linguistics.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008a. The penn discourse treebank 2.0. In *In Proceedings of LREC*.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008b. The Penn Discourse treebank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-08)*, Marrakech, Morocco.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018a. GPT: Improving Language Understanding by Generative Pre-Training. *arXiv*, pages 1–12.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018b. Improving Language Understanding by Generative Pre-Training.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- 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. *Journal of Machine Learning Research*, 21(140):1–67.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6323 LNAI(PART 3):148–163.
- Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia. Association for Computational Linguistics.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.
- Sebastian Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. (May).
- Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive Neural Networks.
- Attapol Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 645–654, Gothenburg, Sweden. Association for Computational Linguistics.
- Attapol Rutherford and Nianwen Xue. 2015. Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 799–808, Denver, Colorado. Association for Computational Linguistics.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. 2017. Bi-Directional Attention Flow for Machine Comprehension. In *Proceedings of International Conference of Learning Representations 2017*, pages 1–12.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana.
- Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2016. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*.
- Parmjit Singh, T Lin, E.T. Mueller, G Lim, T Perkins, and W.L. Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *Lecture Notes in Computer Science*, volume 2519, pages 1223–1237.
- Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015.
- Anders Søgaard and Yoav Goldberg. 2016. Deep multitask learning with low level tasks supervised at lower layers. page 231235.
- Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. 2018. Exploring Graph-structured Passage Representation for Multi-hop Reading Comprehension with Graph Neural Networks.
- Alessandro Sordani, Phillip Bachman, and Yoshua Bengio. 2016. Iterative alternating neural attention for machine reading. abs/1606.02245.
- R. Speer and Joshua Chin. 2016. An ensemble method to produce high-quality word embeddings. *ArXiv*, abs/1604.01692.
- Robert Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *AAAI*.
- Caroline Sporleder and Alex Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: An assessment. *Nat. Lang. Eng.*, 14(3):369–416.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. volume 15, pages 1929–1958.
- Evgeny Stepanov, Giuseppe Riccardi, and Ali Orkan Bayer. 2015. The unitn discourse parser in conll 2015 shared task: Token-level sequence labeling with argument-specific models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 25–31, Beijing, China. Association for Computational Linguistics.



- Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038.
- Saku Sugawara, Hikaru Yokono, and Akiko Aizawa. 2017. Prerequisite skills for reading comprehension: Multi-perspective analysis of mctest datasets and systems. In *AAAI*, pages 3089–3096.
- Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-To-End Memory Networks. In *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Curran Associates, Inc.
- Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242.
- Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2019. Improving machine reading comprehension with general reading strategies. 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 2633–2643, Minneapolis, Minnesota.
- Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.
- Niket Tandon, Bhavana Dalvi, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 57–66.
- Thomas Pellissier Tanon, Denny Vrandeć, San Francisco, Sebastian Schaffert, and Thomas Steiner. 2016. From Freebase to Wikidata : The Great Migration. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1419–1428.
- Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018a. Multi-granular sequence encoding via dilated compositional units for reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2141–2151, Brussels, Belgium. Association for Computational Linguistics.
- Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. 2018b. Densely connected attention propagation for reading comprehension. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems, NIPS'18*, pages 4911–4922.
- Wilson L. Taylor. 1953. cloze procedure: A new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433.
- Robert J. Tierney and James W. Cunningham. 1980. Research on teaching reading comprehension. *Handbook of Research on Reading*, pages 609–655.

- Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.
- Quan Hung Tran, Vu Tran, Tu Vu, Minh Nguyen, and Son Bao Pham. 2015. JAIST: Combining multiple features for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval '15*, pages 215–219, Denver, Colorado, USA.
- Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.
- Adam Trischler, Zheng Ye, Xingdi Yuan, Philip Bachman, Alessandro Sordoni, and Kaheer Suleman. 2016. Natural language comprehension with the epireader. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 128–137. Association for Computational Linguistics.
- 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 *Nips*.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*.
- Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. 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.
- Hai Wang, Takeshi Onishi, Kevin Gimpel, and David McAllester. 2016a. Emergent Logical Structure in Vector Representations of Neural Readers. pages 1–16.
- Jianxiang Wang and Man Lan. 2015. A refined end-to-end discourse parser. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 17–24, Beijing, China. Association for Computational Linguistics.
- Liang Wang, Meng Sun, Wei Zhao, Kewei Shen, and Jingming Liu. 2018b. Yuanfudao at semeval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 758–762.
- Shuohang Wang and Jing Jiang. 2016. Machine Comprehension Using Match-LSTM and Answer Pointer. *Arxiv*, pages 1–12.

- Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198. Association for Computational Linguistics.
- Zhiguo Wang, Haitao Mi, Wael Hamza, and Radu Florian. 2016b. Multi-perspective context matching for machine comprehension. *CoRR*, abs/1612.04211.
- Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. *OntoNotes: A Large Training Corpus for Enhanced Processing*.
- Dirk Weissenborn, Tomas Kocisky, and Chris Dyer. 2017a. Dynamic integration of background knowledge in neural NLU systems. *CoRR*, abs/1706.02596.
- Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017b. Making neural qa as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280. Association for Computational Linguistics.
- Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *TACL*.
- Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015a. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.
- Jason Weston, Antoine Bordes, Sumit Chopra, Tomas Mikolov, and Alexander M. Rush. 2015b. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. *arXiv Prepr*.
- Jason Weston, Sumit Chopra, and Antoine Bordes. 2015c. Memory networks. In *International Conference on Learning Representations (ICLR), 2015*.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.
- Daniel T Willingham. 2006. How knowledge helps. *American Educator*, 45(1):42.
- Yuxiang Wu, Yu Zhao, Baotian Hu, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2022. An efficient memory-augmented transformer for knowledge-intensive NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5184–5196, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. In *International Conference on Learning Representations (ICLR), 2017*, volume abs/1611.01604.

- Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on Freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2326–2336, Berlin, Germany.
- Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad, Christopher Bryant, and Attapol Rutherford. 2015. The conll-2015 shared task on shallow discourse parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 1–16, Beijing, China. Association for Computational Linguistics.
- Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Bonnie Webber, Attapol Rutherford, Chuan Wang, and Hongmin Wang. 2016a. The conll-2016 shared task on multilingual shallow discourse parsing. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task*, Berlin, Germany. Association for Computational Linguistics.
- Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Bonnie Webber, Attapol Rutherford, Chuan Wang, and Hongmin Wang. 2016b. The conll-2016 shared task on multilingual shallow discourse parsing. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task*, Berlin, Germany. Association for Computational Linguistics.
- Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446. Association for Computational Linguistics.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *International Conference on Learning Representations (ICLR), 2015*.
- Yi Yang, Wen-Tau Yih, and Christopher Meek. 2015. WIKIQA: A Challenge Dataset for Open-Domain Question Answering. *Proc. EMNLP 2015*, pages 2013–2018.
- Z. Yang, Bhuwan Dhingra, Y. Yuan, Junjie Hu, William W. Cohen, and R. Salakhutdinov. 2017a. Words or characters? fine-grained gating for reading comprehension. *ArXiv*, abs/1611.01724.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. 2017b. Reference-aware language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1851–1860. Association for Computational Linguistics.
- Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023, Cambridge, MA. Association for Computational Linguistics.

- Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany.
- Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *ICLR 2018*.
- Biao Zhang, Jinsong Su, Deyi Xiong, Yaojie Lu, Hong Duan, and Junfeng Yao. 2015a. Shallow convolutional neural network for implicit discourse relation recognition. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, Lisbon, Portugal. Association for Computational Linguistics.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. *arXiv preprint arXiv:1509.01626*.
- Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015c. Character-level convolutional networks for text classification. In *NIPS*.
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

# Appendix A

## Source Code

The source code used for obtaining the results for Chapters 3,4, and 5 of this Thesis is available at <https://heidata.uni-heidelberg.de/dataset.xhtml?persistentId=doi:10.11588/data/HU3ARF>. For details, see the README.md in each chapter folder.

# Appendix B

## Discourse Relation Sense Classification

### B.1 A System for Discourse Relation Sense Classification

In this section, we describe our approaches from (Mihaylov and Frank, 2016b), developed for the CoNLL 2016 Shared Task’s supplementary task on Discourse Relation Sense Classification. Our system employs a Logistic Regression classifier with several cross-argument similarity features based on word embeddings and performs with overall F-scores of 64.13 for the *Dev* set, 63.31 for the *Test* set and 54.69 for the *Blind* set, ranking first in the *Overall* ranking for the task. We compare the feature-based Logistic Regression classifier to different Convolutional Neural Network architectures. We further enriched our model for Non-Explicit relations by including similarities of explicit connectives with the relation arguments, and part-of-speech similarities based on modal verbs. This improved our *Non-Explicit* result by 1.46 points on the *Dev* set and by 0.36 points on the *Blind* set. We use our best system as a base for our discourse relation annotation, used for Machine Reading Comprehension of narrative texts as described in Chapter 5.

#### B.1.1 Discourse Relation Sense Classification Data

The CoNLL 2016 Shared Task on Shallow Discourse Parsing (Xue et al., 2016a) focuses on identifying individual discourse relations presented in text. The shared task has a main track that requires end-to-end discourse relation parsing and a supplementary task that is restricted to discourse relation sense classification. For the main task, systems are required to build a system that given a raw text as input can identify arguments *Arg1* and *Arg2* that are related in the discourse, and also classify the type of the relation, which can be *Explicit*, *Implicit*, *AltLex* or *EntRel*. A further attribute to be detected is the relation *Sense*, which can be one of 15 classes organized hierarchically in 4 parent classes. With this work, we participated in the

Supplementary Task on Discourse Relation Sense Classification in English. The task is to predict the discourse relation sense when the arguments *Arg1*, *Arg2* are given, as well as the *Discourse Connective* in case of explicit marking.

In our contribution, we compare different approaches including a Logistic Regression classifier using similarity features based on word embeddings, and two Convolutional Neural Network architectures. We show that an approach using only word embeddings retrieved from *word2vec* (Mikolov et al., 2013b) and cross-argument similarity features is simple and fast, and yields results that rank first in the *Overall*, second in the *Explicit* and forth in the *Non-Explicit* sense classification task.

Our system’s code is publicly accessible<sup>1</sup>.

## B.1.2 Related Work

This year’s CoNLL 2016 Shared Task on Shallow Discourse Parsing (Xue et al., 2016a) is the second edition of the shared task after the CoNLL 2015 Shared task on Shallow Discourse Parsing (Xue et al., 2015). The difference to 2015’s task is that there is a new Supplementary Task on Discourse Relation Sense classification, where participants are not required to build an end-to-end discourse relation parser but can participate with a sense classification system only.

Discourse relations in the task are divided into two major types: Explicit and Non-Explicit (*Implicit*, *EntRel* and *AltLex*). Detecting the sense of Explicit relations is an easy task: given the discourse connective, the relation sense can be determined with very high accuracy (Pitler et al., 2008). A challenging task is to detect the sense of Non-Explicit discourse relations, as they usually don’t have a connective that can help to determine their sense.

In the previous version of the task *Non-Explicit* relations have been tackled with features based on Brown clusters (Chiaros and Schenk, 2015; Wang and Lan, 2015; Stepanov et al., 2015), VerbNet classes (Kong et al., 2015; Lalitha Devi et al., 2015) and MPQA polarity lexicon (Wang and Lan, 2015; Lalitha Devi et al., 2015).

Earlier work (Rutherford and Xue, 2014) employed Brown cluster and coreference patterns to identify senses of implicit discourse relations in naturally occurring text. More recently Rutherford and Xue (2015) improved inference of implicit discourse relations via classifying explicit discourse connectives, extending prior research (Marcu and Echihiabi, 2002; Sporleder and Lascarides, 2008). Several neural network approaches have been proposed, e.g., Multi-task Neural Networks (Liu et al., 2016) and Shallow-Convolutional Neural Networks (Zhang et al., 2015a). Braud and Denis (2015) compare word representations for

---

<sup>1</sup><https://github.com/tbmihailov/conll16st-hd-sdp> - Source code for our Discourse Relation Sense Classification system



implicit discourse relation classification and find that denser representations systematically outperform sparser ones.

### B.1.3 Method

We divide the task into two subtasks, and develop separate classifiers for Explicit and Non-Explicit discourse relation sense classification, as shown in Figure . We do that because the official evaluation is divided into Explicit and Non-Explicit (Implicit, AltLex, EntRel) relations and we want to be able to tune our system accordingly. During training, the relation type is provided in the data, and samples are processed by the respective classifier models in *Process 1 (Non-Explicit)* and *Process 2 (Explicit)*. During testing the gold *Type* attribute is not provided, so we use a simple heuristic: we assume that *Explicit* relations have connectives and that *Non-Explicit*<sup>2</sup> relations do not.

As the task requires that the actual evaluation is executed on the provided server, we save the models so we can load them later during evaluation.

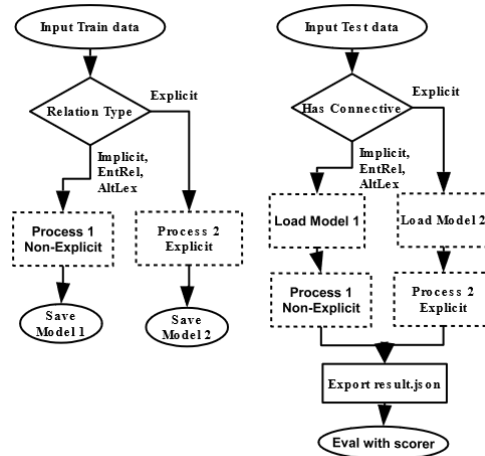


Fig. B.1 System architecture: Training and evaluating models for Explicit and Non-Explicit discourse relation sense classification

For classifying *Explicit* connectives we follow a feature-based approach, developing features based on word embeddings and semantic similarity measured between parts of the arguments *Arg1* and *Arg2* of the discourse relations. Classification is into one of the fifteen classes of relation senses. For detecting *Non-Explicit* discourse relations we also make

<sup>2</sup>In fact, some *AltLex* discourse relations do have connectives, but they are considered *Non-Explicit*. More detailed analysis will be required to improve on this simple heuristic. Given that their distribution across the data sets is very small, they do not have much influence on the overall performance.

use of a feature-based approach, but in addition, we experiment with two models based on Convolutional Neural Networks.

### B.1.3.1 Feature-based approach

For each relation, we extract features from *Arg1*, *Arg2* and the *Connective*, in case the type of the relation is considered *Explicit*.

**Semantic Features using Word Embeddings.** In our models we only develop features based on word embedding vectors. We use *word2vec* (Mikolov et al., 2013b) word embeddings with vector size 300 pre-trained on Google News texts.<sup>3</sup> For computing similarity between embedding representations, we employ cosine similarity:

$$1 - \frac{u \cdot v}{\|u\| \cdot \|v\|} \quad (\text{B.1})$$

**Embedding representations for Arguments and Connectives.** For each argument *Arg1*, *Arg2* and *Connective* (for Explicit relations) we construct a centroid vector ( $\vec{c}$ ) from the embedding vectors  $\vec{w}_i$  of all words  $w_i$  in their respective surface yield.

$$\text{centroid}(\vec{w}_1 \dots \vec{w}_n) = \frac{\sum_{i=1}^n \vec{w}_i}{n} \quad (\text{B.2})$$

**Cross-argument Semantic Vector Similarities.** We calculate various similarity features based on the centroid word vectors for the arguments and the connective, as well as on parts of the arguments:

**Arg1 to Arg2 similarity.** We assume that for given arguments *Arg1* and *Arg2* that stand in a specific discourse relation sense, their centroid vectors should stand in a specific similarity relation to each other. We thus use their cosine similarity as a feature.

**Maximized similarity.** Here we rank each word in *Arg2*'s text according to its similarity with the centroid vector of *Arg1*, and we compute the average similarity for the top-ranked  $N$  words. We chose the similarity scores of the top 1,2,3 and 5 words as features. The assumption is that the average similarity between the first argument (*Arg1*) and the top  $N$  most similar words in the second argument (*Arg2*) might imply a specific sense.

---

<sup>3</sup><https://code.google.com/archive/p/word2vec/> - Pre-trained vectors trained on part of the Google News dataset (about 100 billion words).

**Aligned similarity.** For each word in *Arg1*, we choose the most similar word from the yield of *Arg2* and we take the average of all best word pair similarities, as suggested in Tran et al. (2015).

**Part of speech (POS) based word vector similarities.** We used part of speech tags from the parsed input data provided by the organizers and computed similarities between centroid vectors of words with a specific tag from *Arg1* and the centroid vector of *Arg2*. Extracted features for POS similarities are symmetric: for example, we calculate the similarity between *Nouns* from *Arg1* with *Pronouns* from *Arg2* and the opposite. The assumption is that some parts of speech between *Arg1* and *Arg2* might be closer than other parts of speech depending on the relation sense.

**Explicit discourse connectives similarity.** We collected 103 explicit discourse connectives from the Penn Discourse Treebank (Prasad et al., 2008a) annotation manual<sup>4</sup> and for all of them construct vector representations according to (), where for multi-token connectives we calculate a centroid vector from all tokens in the connective. For every discourse connective vector representation, we calculate the similarity with the centroid vector representations from all *Arg1* and *Arg2* tokens. This results in adding 103 similarity features for every relation. We use these features for implicit discourse relations sense classification only.

We assume that knowledge about the relation sense can be inferred by calculating the similarity between the semantic information of the relation arguments and specific discourse connectives. Our feature-based approach yields very good results on Explicit relations sense classification with an F-score of 0.912 on the *Dev* set. Combining features based on word embeddings and similarity between arguments in Mihaylov and Nakov (2016) yielded state-of-the-art performance in a similar task setup in Community Question Answering (Nakov et al., 2016a), where two text arguments (question and answer) are to be ranked.

### B.1.3.2 CNNs for sentence classification

We also experiment with Convolutional Neural Network architectures to detect Implicit relation senses. We have implemented the CNN model proposed in Kim (2014) as it proved successful in tasks like sentence classification and modal sense classification (Marasović and Frank, 2016). This model (Figure ) defines one convolutional layer that uses pre-trained *Word2Vec* vectors trained on the Google News dataset. As shown in Kim (2014), this architecture yields very good results for various single-sentence classification tasks. For our relation classification task, we input the concatenated tokens of *Arg1* and *Arg2*.

---

<sup>4</sup><https://www.seas.upenn.edu/~pdtb/PDTBAPI/pdtb-annotation-manual.pdf> - The Penn Discourse Treebank 2.0 Annotation Manual

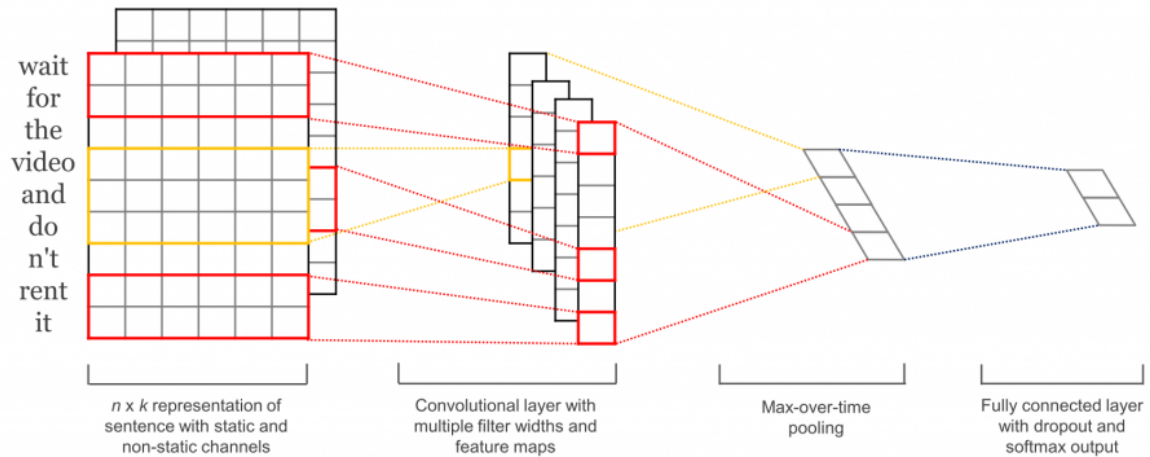


Fig. B.2 CNN architecture by Kim (2014) (figure is from Kim (2014))

### B.1.3.3 Modified ARC-1 CNN for sentence matching

An alternative model we try for Implicit discourse relation sense classification is a modification of the *ARC-1* architecture proposed for sentence matching by Hu et al. (2015). We will refer to this model as *ARC-IM*.

The modified architecture is depicted in Figure . The input of the model are two sentences  $S_x$  and  $S_y$  represented as sequence of tokens' vector representations of *Arg1* and *Arg2*. Here, separate convolution and max-pooling layers are constructed for the two input sentences, and the results of the max-pooling layers are concatenated and fed to a single final *SoftMax* layer. The original *ARC-1* architecture uses a *Multilayer Perceptron* layer instead of *SoftMax*. For our implementation, we use TensorFlow (Abadi et al., 2015).

## B.1.4 Experiments and Results

### B.1.4.1 Data

In our experiments we use the official data (English) provided from the task organizers: *Train* (15500 Explicit + 18115 Non-Explicit), *Dev* (740 Explicit + 782 Non-Explicit), *Test* (990 Explicit + 1026 Non-Explicit), *Blind* (608 Explicit + 661 Non-Explicit). All models are trained on *Train* set.

### B.1.4.2 Classifier settings

For our feature-based approach, we concatenate the extracted features in a feature vector, scale their values to the 0 to 1 range, and feed the vectors to a classifier. We train and evaluate

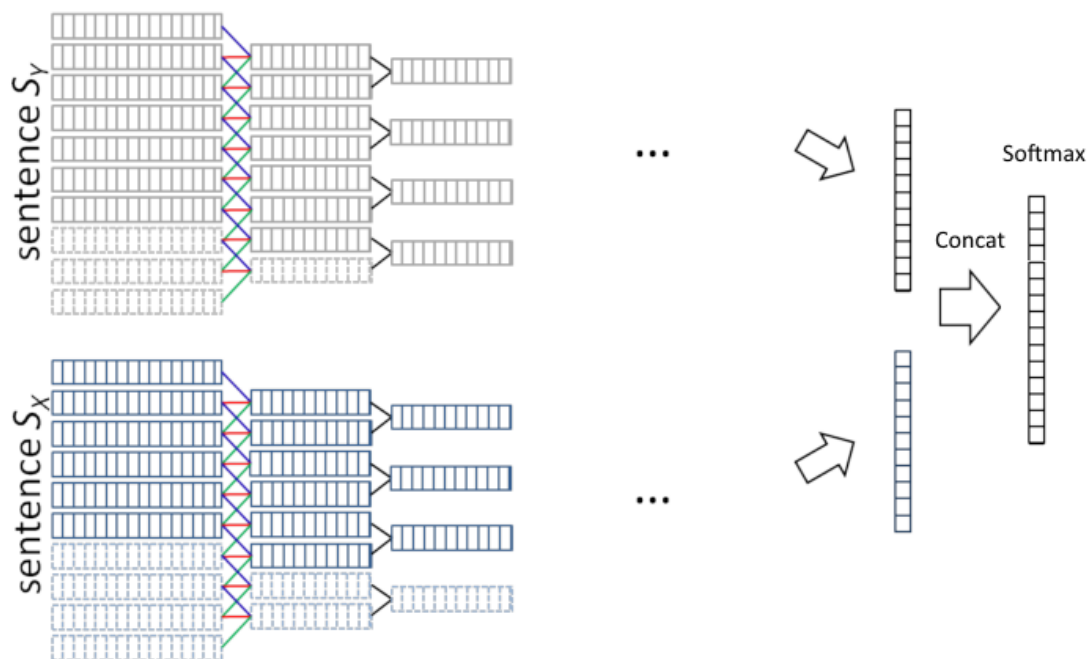


Fig. B.3 Modified ARC-I CNN architecture for sentence matching.

an L2-regularized Logistic Regression classifier with the LIBLINEAR (Fan et al., 2008) solver as implemented in *scikit-learn* (Pedregosa et al., 2011). For most of our experiments, we tuned the classifier with different values of the C (cost) parameter and chose  $C=0.1$  as it yielded the best accuracy on 5-fold cross-validation on the training set. We use these settings for all experiments that use the logistic regression classifier.

#### B.1.4.3 Official submission (LR with E+Sim)

Our official submission uses the feature-based approach described in Section for both *Explicit* and *Non-Explicit* relations with all features described above, except for the *Explicit connective similarities (Conn)* and *Modal verbs similarities (POS MD)* which have been added after the submission deadline. Table presents the results divided by senses from our official submission performed on the TIRA evaluation platform (Potthast et al., 2014) server. We also compare our official and improved system results to the best-performing system in the CoNLL 2015 Shared Task (Wang and Lan, 2015) and the best-performing systems in the CoNLL 2016 Discourse Relation Sense Classification task. With our official system, we rank first in the *Overall*<sup>5</sup> ranking. We rank second in the *Explicit* ranking with a small difference of 0.07 behind the best system and fourth in the *Non-Explicit* ranking with a more

<sup>5</sup>Overall score is the F-score on All (both *Explicit* and *Non-Explicit*) relations.

Sense	WSJ Dev Set			WSJ Test Set			Blind Set (Official task ranking)		
	Overall	Exp	Non-E	Overall	Exp	Non-E	Overall	Exp	Non-E
Comparison.Concession	33.33	40.00	0.00	36.36	44.44	0.00	91.67	100.00	0.00
Comparison.Contrast	74.31	94.44	16.07	65.99	92.19	9.60	21.24	25.81	0.00
Contingency.Cause.Reason	51.48	78.95	38.51	64.36	94.03	47.93	35.71	82.61	18.03
Contingency.Cause.Result	38.94	91.43	15.38	40.74	100.00	17.53	53.33	91.67	27.78
Contingency.Condition	95.56	95.56	-	87.50	87.50	-	89.66	89.66	-
EntRel	58.73	-	58.73	70.97	-	70.97	47.06	-	47.06
Expansion.Alt	92.31	92.31	-	100.00	100.00	-	100.00	100.00	-
Expansion.Alt.Chosen alt	71.43	90.91	0.00	22.22	100.00	6.67	0.00	-	100.00
Expansion.Conjunction	70.45	97.00	40.00	75.88	98.36	40.26	63.48	94.52	27.51
Expansion.Instantiation	47.73	100.00	34.29	57.14	100.00	44.29	55.56	100.00	50.00
Expansion.Restatement	31.13	66.67	29.56	31.31	14.29	31.94	32.39	66.67	30.88
Temporal.Async.Precedence	78.46	98.00	13.33	82.22	100.00	11.11	84.44	97.44	0.00
Temporal.Async.Succession	82.83	87.23	0.00	58.82	63.49	0.00	96.08	96.08	-
Temporal.Synchrony	77.30	80.77	0.00	80.25	83.33	0.00	59.70	59.70	100.00
<b>System</b>	<b>All senses - comparison</b>								
Our system (Official)	64.13	91.20	40.32	<b>63.31</b>	89.80	<b>39.19</b>	54.69	78.34	34.56
Our improved system	64.77	91.05	41.66	62.69	90.02	37.81	<b>54.88</b>	78.38	34.92
Wang and Lan, 2015	<b>65.11</b>	90.00	<b>42.72</b>	61.27	<b>90.79</b>	34.45	54.76	76.44	36.29
Rutherford and Xue, 2016	-	-	40.32	-	-	36.13	-	-	<b>37.67</b>
Jain, 2016	62.43	<b>91.50</b>	36.85	50.90	89.70	15.60	41.47	<b>78.56</b>	9.95

Table B.1 Evaluation of our official submission system, trained on Train 2016 and evaluated on Dev, Test, and Blind sets. Comparison with our official system and our improved system with the official results of CoNLL 2015 Shared Task’s best system (Wang and Lan, 2015) and CoNLL 2016 Shared Task best systems in *Explicit* (Jain, 2016) and *Non-Explicit* (Rutherford and Xue, 2016). F-Score is presented.

significant difference of 2.75 behind the best system. We can see that similar to (Wang and Lan, 2015) our system performs well in classifying both types, while this year’s winning systems perform well in their winning relation type and much worse in the others. <sup>6</sup>

#### B.1.4.4 Further experiments on Non-Explicit relations

In Table we compare different models for *Non-Explicit* relation sense classification trained on the *Train* and evaluated on the *Dev* set.

**Embeddings only experiments.** The first three columns show the results obtained with three approaches that use only features based on word embeddings. We use *word2vec* word embeddings. We also experimented with pre-trained *dependency-based* word embeddings (Levy and Goldberg, 2014), but this yielded slightly worse results on the *Dev* set.

<sup>6</sup>The winner team in *Non-Explicit* (Rutherford and Xue, 2016) does not participate in *Explicit*.

Sense	Embeddings only			Logistic Regression with Embeddings + Features			
	LR	CNN	CNN ARC-1M	E+Sim	E+Sim+Conn	E+Sim+Conn+POS	MD
Comparison.Concession	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Comparison.Contrast	2.33	13.68	8.51	16.07	<b>18.80</b>		17.86
Contingency.Cause.Reason	25.00	29.30	35.90	38.51	40.24		<b>42.17</b>
Contingency.Cause.Result	3.57	9.20	<b>19.28</b>	15.38	15.38		13.70
EntRel	53.13	59.53	56.87	58.73	60.80		<b>61.26</b>
Expansion.Alt.Chosen alt	0.00	0.00	0.00	0.00	0.00		0.00
Expansion.Conjunction	35.90	38.29	14.67	40.00	40.91		<b>41.27</b>
Expansion.Instantiation	0.00	21.98	4.08	<b>34.29</b>	31.43		33.80
Expansion.Restatement	12.74	0.00	21.56	<b>29.56</b>	26.87		27.45
Temporal.Async.Precedence	0.00	0.00	0.00	13.33	<b>17.65</b>		12.90
Temporal.Async.Succession	0.00	0.00	0.00	0.00	0.00		0.00
Temporal.Synchrony	0.00	0.00	0.00	0.00	0.00		0.00
All	35.54	34.34	36.21	40.32	40.99		<b>41.66</b>

Table B.2 Evaluation of different systems and feature configurations for Non-Explicit relation sense classification, trained on Train 2016 and evaluated on Dev. F-score is presented.

**Logistic Regression (LR).** The *LR* column shows the results from a Logistic Regression classifier that uses only the concatenated features from the centroid representations built from the words of *Arg1* and *Arg2*.

**CNN experiments.** The *CNN* column shows results obtained from the Convolutional Neural Network for sentence classification (Section ) fed with the concatenated *Arg1* and *Arg2* word tokens’ vector representations from *Word2Vec* word embeddings. For our experiments we used default system parameters as proposed in Kim (2014): filter windows with size 3,4,5 with 100 feature maps each, dropout probability 0.5, and mini-batch of size 50. We train the model with 50 epochs.

**CNN ARC-1M experiments** The *CNN ARC-1M* column shows results from our modification of ARC-1 CNN for sentence matching (see Section ) fed with *Arg1* and *Arg2* word tokens’ vector representations from the *Word2Vec* word embeddings. We use filter windows with size 3,4,5 with 100 feature maps each, shared between the two argument convolutions, dropout probability 0.5 and mini-batch of size 50 as proposed in Kim (2014). We train the model with 50 epochs.

Comparing *LR*, *CNN* and *CNN ARC-1M* according to their ability to classify different classes we observe that *CNN ARC-1M* performs best in detecting *Contingency.Cause.Reason* and *Contingency.Cause.Result* with a substantial margin over the other two models. The *CNN* model outperforms the *LR* and *CNN-ARC1M* for *Comparison.Contrast*, *EntRel*, *Expansion.Conjunction* and *Expansion.Instantiation* but cannot capture any *Expansion.Restatement* which leads to worse overall results compared to the others. These insights show that the

Neural Network models are able to capture some dependencies between the relation arguments. For *Contingency.Cause.Results*, *CNN ARC-1M* even clearly outperforms the LR models enhanced with similarity features (discussed below). We also implemented a modified version of the *CNN ARC-2* architecture of Hu et al. (2015), which uses a cross-argument convolution layer, but it yielded much worse results.<sup>7</sup>

**LR with Embeddings + Features** The last three columns in Table show the results of our feature-based Logistic Regression approach with different feature groups on top of the embedding representations of the arguments. Column *E+Sim* shows the results from our official submission and the other two columns show results for additional features that we added after the submission deadline.

Adding the cross-argument similarity features (without the POS modal verbs similarities) improves the overall result of the embeddings-only Logistic Regression (*LR*) baseline significantly from F-score 35.54 to 40.32. It also improves the result on almost all senses individually. Adding *Explicit connective similarities* features improves the *All* result by 0.67 points (*E+Sim+Conn*). It also improves the performance on *Temporal.Async.Precedence*, *Expansion.Conjunction*, *EntRel*, *Contingency.Cause.Reason* and *Comparison.Contrast* individually. We further added *POS similarity features* between *MD (modal verbs)* and other part of speech tags between *Arg1* and *Arg2*. The obtained improvement of 0.67 points shows that the occurrence of modal verbs within arguments can be exploited for implicit discourse relation sense classification. Adding the modal verbs similarities also improved the individual results for the *Contingency.Cause.Reason*, *EntRel* and *Expansion.Conjunction* senses.

Some relations are hard to predict, probably due to the low distribution in the train and evaluation data sets: *Comparison.Concession*<sup>8</sup>, *Expansion.Alt.Chosen alt*<sup>9</sup>, *Temporal.Async.Succession*<sup>10</sup>, *Temporal.Synchrony*<sup>11</sup>.

## B.1.5 Summary

In this section, we describe our system for participation in the CoNLL Shared Task on Discourse Relation Sense Classification. We compare different approaches including Logistic Regression classifiers using features based on word embeddings and cross-argument similarity and two Convolutional Neural Network architectures. Our official submission uses a logistic regression classifier with several similarity features and performs with overall F-scores of

---

<sup>7</sup>We are currently checking our implementation.

<sup>8</sup>*Comparison.Concession, Non-Explicit*: Train:1.10 %, Dev:0.66 %: Test:0.59 %.

<sup>9</sup>*Expansion.Alt.Chosen-alt, Non-Explicit*: Train:0.79 %, Dev:0.26 %: Test:1.49 %.

<sup>10</sup>*Temporal.Async.Succ, Non-Explicit*: Train:0.80 %, Dev:0.39 %: Test:0.49 %.

<sup>11</sup>*Temporal.Synchrony, Non-Explicit*: Train:0.94 %, Dev:1.19 %: Test:0.49 %.



64.13 for the *Dev* set, 63.31 for the *Test* set, and 54.69 for the *Blind* set. After the official submission, we improved our system by adding more features for detecting senses for Non-Explicit relations and we improved our *Non-Explicit* result by 1.46 points to 41.66 on the *Dev* set and by 0.36 points to 34.92 on the *Blind* set.

We could show that dense representations of arguments and connectives jointly with cross-argument similarity features calculated over word embeddings yield competitive results, both for Explicit and Non-Explicit relations.