Document Meta-Information as Weak Supervision for Machine Translation

Inauguraldissertation zur Erlangung des akademischen Grades
Doktor der Philosophie der Neuphilologischen Fakultät
der Ruprecht-Karls-Universität Heidelberg
vorgelegt von

Laura Elisabeth Jehl

12. Dezember 2018

Erstgutachter:	Prof. Dr. Stefan Riezler Institut für Computerlinguistik, Universität Heidelberg
Zweitgutachter:	Prof. Dr. Michael Gertz Institut für Informatik, Universität Heidelberg
Datum der Disputation:	25. Februar 2019

Zusammenfassung

Datengetriebene maschinelle Übersetzungssysteme konnten seit den ersten bahnbrechenden Arbeiten in den 1990-er Jahren enorm verbessert werden. Neuerdings werden Systeme vorgestellt, die isolierte Sätze ebenso gut übersetzen wie professionelle menschliche Übersetzer. Die Voraussetzung dafür sind ausreichend übersetzte Trainingssätze in der Zieldomäne, von denen ein System lernen kann. In Domänen, für die es keine übersetzten Sätze gibt, nimmt die Qualität dieser Systeme jedoch drastisch ab. Zudem beziehen maschinelle Übersetzungssysteme üblicherweise keinen Kontext über die Satzebene hinaus mit ein. Solches Kontextwissen kann jedoch notwendig sein, um einen Satz korrekt zu übersetzen. Die vorliegende Arbeit hat zum Ziel, einen Beitrag zur Lösung der oben genannten Probleme zu leisten, indem sie untersucht, wie Meta-Informationen auf Dokumentebene in Übersetzungssysteme integriert werden können. Beispiele für Meta-Informationen auf Dokumentebene sind Informationen zur inhaltlichen Kategorisierung und Urheberschaft des Dokuments oder explizite Verbindungen zwischen Dokumenten, zum Beispiel durch Hyperlinks oder Zitate. In vier kumulativ durchgeführten Fallstudien werden Methoden entwickelt und ausgewertet, mit denen solche Meta-Informationen als schwaches Trainingssignal für die maschinelle Übersetzung nutzbar gemacht werden können. Dabei werden Meta-Informationen sowohl eingesetzt, um ein System auf eine Zieldomäne anzupassen, für die passende Trainingsdaten fehlen, als auch, um Trainingsdaten auf Satzebene mit Kontextinformation auf Dokumentebene anzureichern. In allen Fallstudien kann die Übersetzungsqualität verbessert werden.

Abstract

Data-driven machine translation has advanced considerably since the first pioneering work in the 1990s with recent systems claiming human parity on sentence translation for high-resource tasks. However, performance degrades for low-resource domains with no available sentence-parallel training data. Machine translation systems also rarely incorporate the document context beyond the sentence level, ignoring knowledge which is essential for some situations. In this thesis, we aim to address the two issues mentioned above by examining ways to incorporate document-level meta-information into data-driven machine translation. Examples of document meta-information include document authorship and categorization information, as well as cross-lingual correspondences between documents, such as hyperlinks or citations between documents. As this meta-information is much more coarse-grained than reference translations, it constitutes a source of weak supervision for machine translation. We present four cumulatively conducted case studies where we devise and evaluate methods to exploit these sources of weak supervision both in low-resource scenarios where no task-appropriate supervision from parallel data exists, and in a full supervision scenario where weak supervision from document meta-information is used to supplement supervision from sentence-level reference translations. All case studies show improved translation quality when incorporating document meta-information.

Contents

CHAPTER 1

Introduction

Interest in machine translation has been apparent since the advent of electronic computers in the 1940s (Hutchins, 2007). Early approaches, focusing on extensive lexical and grammatical rules based on formal linguistic theory, were successful in limited domains and computer-aided translation. In the 1990s, data-driven machine translation, which uses statistical methods to infer knowledge from large amounts of translated data, emerged as a strong competitor to these approaches. Starting with the IBM models by Brown, Della Pietra, Della Pietra, and Mercer (1993), phrase-based and hierarchical phrase-based systems (Chiang, 2005; Koehn, Och, & Marcu, 2003) received interest from Natural Language Processing (NLP) researchers and, increasingly, from companies. Since 2015, the phrase-based statistical machine translation (SMT) paradigm has been surpassed by neural machine translation (NMT) models based on deep neural networks (Bahdanau, Cho, & Bengio, 2015; Sutskever, Vinyals, & Le, 2014). These models have advanced the state-of-the-art considerably, leading to the first claims of human parity in 2018 (Linn, 2018).

Despite this impressive progress, machine translation as a whole cannot be considered a solved problem. One issue for current systems is their reliance on large amounts of translated data. In fact, advances in machine translation have to a large extent been

fueled by regular, public evaluation campaigns.[1] An important part of these evaluation efforts has been compiling and distributing suitable data for training and evaluation. While such data is invaluable for progress in machine translation research, low-resource domains or languages still suffer a significant decay in translation quality. Another issue is the current focus on the sentence as the unit of translation. Even in a resource-rich scenario, some ambiguities in a sentence cannot be satisfactorily resolved without knowledge of a wider context. If no context is provided, even human translators will be unable to produce the correct translation.

Solving the two problems described above has been the target of recent research. To address the first issue, the reliance on translated data, previous work has either focused on cheaply expanding the amount and range of available data, for example by mining translations from multilingual sources (Munteanu & Marcu, 2005; Smith et al., 2013; Wołk & Marasek, 2015, *inter alia*) or by using crowdsourcing to produce translations for low-resource languages (Post, Callison-Burch, & Osborne, 2012; Zaidan & Callison-Burch, 2011b, *inter alia*). Recently, training methods for neural machine translation systems have been introduced, which are capable of being trained only from monolingual data in the source and target language (Artetxe, Labaka, Agirre, & Cho, 2018; Lample, Denoyer, & Ranzato, 2018). An extensive amount of work has also been conducted on the problem of domain adaptation - i.e., fitting an existing model to a new domain – (Axelrod, He, & Gao, 2011; B. Chen, Kuhn, & Foster, 2013; Eidelman, Boyd-Graber, & Resnik, 2012; Foster & Kuhn, 2007; Hasler, Haddow, & Koehn, 2014; Hewavitharana, Mehay, Ananthakrishnan, & Natarajan, 2013; Matsoukas, Rosti, & Zhang, 2009, *inter alia*) for SMT and (B. Chen, Cherry, Foster, & Larkin, 2017; B. Chen & Huang, 2016; Chu, Dabre, & Kurohashi, 2017; Freitag & Al-Onaizan, 2016; R. Wang, Utiyama, Liu, Chen, & Sumita, 2017; Zhang, Li, Way, & Liu, 2016, *inter alia*) for NMT. To address the second issue, the restriction to sentence-level translation, various approaches have been devised to include the document text beyond the sentence, e.g. by keeping a cache of previous translations (Grave, Joulin, & Usunier, 2017; Hardmeier, Nivre, & Tiedemann, 2012; Tiedemann, 2010, *inter alia*). Other approaches have targeted specific phenomena related to missing context information, such as anaphora resolution or dropped pronoun restoration (Hardmeier & Federico, 2010; L. Wang, Tu, Way, & Liu, 2018, *inter alia*).

This thesis complements the above-mentioned approaches by exploring the potential of a rarely tapped resource in the context of machine translation research: document meta-information, including metadata, document-level links, or other document characteristics. Where most of the approaches mentioned above only rely on the document text, this

[1] The most well-known campaigns are the Conference on Machine Translation, the International Workshop an Spoken Language Translation and the NIST Open Machine Translation Evaluations.

thesis focuses on the meta-information beyond the text. As most of the available data for machine translation is restricted to sentence-level plain text, this information can be easily overlooked. At the same time, it can be obtained for many data collections, which lack translated training data. This thesis sets out to devise and test methods for making use of this information in machine translation. In response to the first problem, we explore document meta-information as guidance for automatically extracting additional training data, and as a weak signal for supervised training. In response to the second problem, we include document information in the translation model as a source of knowledge about the document context.

1.1 Research Question

Before the contributions of this thesis are laid out in more detail, the core concepts and main research question are defined.

Machine Translation All the work presented in this thesis is concerned with improving machine translation. By machine translation we mean the automatic translation of text in language A to language B using a machine translation system. More specifically, we work on *data-driven* machine translation which uses statistical methods and large data sets, rather than expert knowledge, to model how to translate a sentence from language A to language B. Section 2 provides a detailed overview of relevant theory and technology.

Document meta-information While there is a substantial body of literature on the question of what constitutes a document (Buckland, 2015), for this thesis it suffices to follow the "conventional, material view" of a document as a "graphic record, usually text" that is located on some medium, in our case a digital file (Buckland, 2015, p.7). We use *document meta-information* as a broad blanket term to refer to any explicitly available information about a text document as a whole, beyond the surface text. Meta-information is normally stored in document metadata, which contains curated, structured information about a document. We also include document-level links within a larger document collection. Thirdly, we make use of other structural characteristics of a document derived e.g. from its medium, such as a character limit for Twitter messages, for example. In summary, this thesis focuses on *explicit*, *document level* information, that exists *outside of the text*.

Weak supervision In fully supervised learning, the available supervision is strong enough to constitute a ground truth for the task. We distinguish these scenarios from weak supervision scenarios, where the available supervision provides some guidance, but is not sufficient to establish a ground truth. The supervision could, for example, be too coarse-grained, unreliable or incomplete (categorization from Zhou (2017)). Hence, lower performance is expected compared to a full supervision scenario.

To combine the above concepts, this thesis aims to leverage weak supervision in the form of document metadata, document-level links or document characteristics to improve machine translation. As document meta-information, unlike translated data, is insufficient to establish a ground truth for a translation task, this information can only constitute weak supervision for this task. Nevertheless, this information could provide helpful guidance in low-resource scenarios. Additionally, document meta-information might even contain contextual clues that are lacking from full sentence-level supervision due to the limited context. We examine these hypotheses in different case studies, identify available document meta-information and lay out, implement and evaluate novel methods for employing this information to improve machine translation.

1.2 Chapter Overview

While all chapters relate to the central research goal formulated above, they can each be viewed as self-contained units and read selectively according to special interest. Most contributions have been published in peer-reviewed outlets. In this section, we summarize the main directions of each chapter, its relation to the overall goals, as well as its applicability in a broader context - concerning different machine translation paradigms, as well as different translation tasks.

In **Chapter 3**, we use document meta-information in a topically constrained translation task, where in-domain supervision is not available. This chapter presents a solution that is specifically tailored to the considered task of topically constrained Arabic-English Twitter translation. Starting out from the observation that information and comments about events of global significance will dissipate on microblogging platforms like Twitter[2] simultaneously and multilingually, we present a method to construct a topically constrained Twitter translation system using automatically extracted pseudo-parallel in-domain data. Document meta-information in the form of user-generated annotations (hashtags), as well as information about the characteristics of Twitter messages is used to crawl and automatically pair Arabic and English Twitter messages concerning the aftermath of the Arab

[2] twitter.com/

Spring. The extracted messages are then used to extract an in-domain phrase-table. By combining this phrase-table with an out-of-domain phrase table in a phrase-based SMT system, substantial improvements are achieved over other domain adaptation methods. Most parts of this chapter are published in Jehl, Hieber, and Riezler (2012). The main contributions by the author of this thesis are:

- An end-to-end approach to building a topically constrained translation system for Twitter messages using pseudo-parallel data extracted by leveraging document meta-information (in cooperation with co-authors).
- Strategies for extracting an in-domain phrase table from candidate pairs of tweets.
- Construction of an in-domain evaluation set via crowdsourcing.
- Experimental evaluation of the proposed strategies.
- Extensive comparison to existing domain adaptation approaches which use monolingual data or a small development set.
- A detailed qualitative and quantitative analysis of the strength and weaknesses of the proposed approach.

In **Chapter 4** we use document meta-information to simulate a downstream task loss to optimize a query translation system. We start out from the observation that in patent prior art search query translation requires full text machine translation, but that search queries do not need to satisfy the same requirements as translations in a translation task. We again operate in a scenario of incomplete supervision, as we do not have references specifically for patent query translation. We approach the problem by leveraging patent citation information to construct a large-scale training set for cross-lingual patent retrieval, and by providing a tailored pairwise ranking approach to optimizing patent query translation in a phrase-based system using a downstream task loss for optimization. Our method shows promising trends according to several retrieval metrics. The baseline and data set of this task were also described in Sokolov, Jehl, Hieber, and Riezler (2013). The main contributions of this chapter are:

- A method to exploit the hierarchical structure of patent citations along with cross-lingual patent family relations for the generation of large-scale patent retrieval training and evaluation data.

- A state-of-the-art patent translation system, which was ranked second among constrained systems at the NTCIR10 shared task on patent translation (Simianer et al., 2013).

- A pairwise-ranking method for directly optimizing patent translation for patent retrieval quality.

- Evaluation of our method on Japanese-English patent retrieval.

In **Chapter 5** we again face a setup of incomplete supervision in a domain adaptation scenario, where no in-domain parallel data are available. We leverage cross-lingual document links directly as a weak supervision signal. As supervision from a relevant document provides a very weak signal compared to a translation reference, we add negative supervision from irrelevant documents. A natural way to integrate positive and negative supervision is via a structured ramp loss objective, which deliberately selects a *hope* and *fear* output, which are then promoted (hope) or demoted (fear). We call objectives with this property "bipolar". As the objective is agnostic of a specific translation paradigm, we explore it for both SMT and NMT paradigms. For SMT, we show that sparse lexical features weights can be learned successfully using weak supervision from document-level links with bipolar ramp loss. We also show improvements when using only weak links and when running our approach on top of other adaptation methods. For NMT, we focus on analyzing the performance of different ramp loss formulations and compare them to minimum risk training objectives that also optimize sequence-level metrics. Of the proposed objectives, only the bipolar variants, which incorporate positive and negative supervision, show significant improvements over an unadapted baseline. We also study the performance of ramp loss on a fully supervised task, as this objective has not yet been studied for NMT, and confirm the superiority of bipolar ramp loss compared to other objectives.Our work on SMT is published in Jehl and Riezler (2016). The work on NMT is part of a submitted paper draft (Jehl, Lawrence, & Riezler, 2018). To summarize, we provide the following contributions:

- New bipolar ramp loss objectives for weakly supervised learning scenarios with positive and negative supervision.

- Two new reward metrics, which can be incorporated into metric-augmented structured prediction objectives for using document-level links as weak supervision for machine translation.

- Setup of a Wikipedia translation task to evaluate the method, including a small development and test set and an analysis of parallelism on Wikipedia.

- Evaluation of our objectives for tuning sparse lexicalized features of an SMT system and comparison to other adaptation methods.

- Evaluation of bipolar and non-bipolar ramp loss objectives for NMT and comparison to minimum risk training in weakly and fully supervised scenarios.

In **Chapter 6** we use document meta-information on top of a full supervision setup. We present ways of incorporating document metadata as side constraints on the source side of training and test data. We evaluate our approach on patent translation, where rich metadata is available and presents additional modeling challenges, but the methods could be applied to other collections with metadata. Our work starts out from the observation that information about patent classification or a patent's author might provide clues for the correct word choice when translating individual sentences, as the frequencies of translation options vary across, e.g. patent classes. We provide different ways to pass metadata information to the translation model as side constraints at the phrase level (SMT) or at the sentence or word level (NMT) and evaluate our approaches on two patent translation tasks, showing significant improvements for both NMT task and for the Japanese-English SMT task. Most of this work has been published in Jehl and Riezler (2018). The main contributions are:

- General formulation of document meta-information as side constraints.

- Approaches to integrate document meta-information on the source side for SMT and NMT.

- Handling of multi-category and multi-value information.

- Extensive evaluation of the different strategies on two patent translation tasks.

Summary of contributions

This thesis investigates the use of document meta-information as weak supervision for statistical and neural machine translation. Four case studies are conducted where potentially useful document meta-information is identified in multilingual data collections. The scenarios include cases of incomplete supervision, as well as one case where document meta-information is used to complement full sentence-level supervision. Task-specific, as well as generally applicable, novel approaches to integrating weak supervision from document meta-information into these scenarios are laid out. Finally, the methods are applied

	Chapter 3	Chapter 4	Chapter 5	Chapter 6
DATA	Twitter	Patents	Wikipedia	Patents
TASK	domain adaptation	query translation	domain adaptation	end-to-end translation
SUPERVISION	incomplete	incomplete	incomplete	full
META-INFORMATION	hashtags, character limit	citations	interlingua, inter-article links	metadata
USE OF META-INFORMATION	pseudo-parallel data extraction	simulation of downstream loss	weak supervision instead of reference	side constraints
APPLICABILITY	tailored	tailored	general	general
TRANSLATION PARADIGMS	SMT	SMT	SMT/NMT	SMT/NMT

Table 1.1 Overview of cases studied in this thesis.

not only to established benchmark tasks, but also to new domains, such as Twitter messages or Wikipedia. Table 1.1 summarizes the different cases studied in this thesis, their use of meta-information, and their general applicability. The work for this thesis was conducted cumulatively, spanning the paradigm shift between statistical and neural machine translation. Mainly for this reason, Chapters 3 and 4 only include experiments using SMT systems. Both approaches are, however, easily transferable to the neural paradigm, as is discussed briefly at the end of the conclusion for each chapter. For each study, improvements over a system without meta-information are achieved. As each study presents a self-contained research project with a different task and different data, the results are not directly comparable. When looking at the magnitude of the gains achieved for each study, using document meta-information for pseudo-parallel data extraction is the most effective at the cost of generality, as this method needs to be tailored to the target data.

CHAPTER 2

Background

The following pages outline the fundamentals of data-driven machine translation technology, as relevant to the understanding of this thesis. Section 2.1 describes training data, data processing and evaluation. The subsequent sections describe two predominant approaches to data-driven translation: phrase-based *Statistical Machine Translation* (SMT) in Section 2.2 and *Neural Machine Translation* (NMT) in Section 2.3.

2.1 Input and Output

Data-driven machine translation systems are trained on millions of *parallel* training sentences. *Parallel text* or *bitext* has been translated by proficient human translators into one or more languages and is collected in a *parallel corpus*. As it is infeasible in a non-commercial setting to produce this data, efforts have been directed at collecting bitext from existing sources. These include multilingual parliamentary proceedings, such as the Europarl (Koehn, 2005) and MultiUN corpora (Eisele & Chen, 2010; Ziemski, Junczys-Dowmunt, & Pouliquen, 2016), the Web Inventory of transcribed TED talks (Cettolo, Girardi, & Federico, 2012) and multilingual news outlets[1] and the PatTR corpus contain-

[1] casmacat.eu/corpus/news-commentary.html

ing German, English and French patent text (Wäschle & Riezler, 2012a). Other parallel data sets, especially for non-European languages, have been provided as part of the NIST evaluation campaigns[2]. Assembling a parallel corpus is an involved process: After obtaining the text in digital form, the plain text is extracted and the data are aligned at the document level. As a machine translation system requires sentence-aligned data, the documents are broken down into sentences and aligned across languages. Both tasks are non-trivial, as end-of-sentence markers can be ambiguous and sentence boundaries can differ between languages. The need for heavy processing of parallel data could be viewed as one reason why document information has been disregarded in machine translation. The corpora rarely provide markup or annotation at the document level alongside the sentence-aligned text.

Due to the difficulty of obtaining bitext, only certain domains and genres are covered. Weaker sources of knowledge are provided by *comparable corpora* and *monolingual corpora*. A comparable corpus is a multilingual data set which does not contain direct translation, but whose contents are related across languages. Monolingual data do not contain knowledge about translational equivalence, but provide information about fluency and are easily available for many domains and languages. Especially in phrase-based SMT, monolingual target language data play an important role in language modeling.

Both SMT and NMT require data pre- and post-processing. *Tokenization* refers to splitting sentences into words (*tokens*). This is done by splitting sentences on whitespace for whitespace- delimited languages. For non whitespace-delimited languages like Japanese, tokenization requires more sophisticated models. Tokenization is required for machine translation, as units smaller than sentences are necessary for collecting reliable statistics. Other pre-processing steps are often applied to reduce the vocabulary. *Lowercasing* or *truecasing*, which removes sentence-initial uppercase, can be applied to languages with Latin script. Splitting text into *subword* units is another way to reduce vocabulary. This could be accomplished by splitting words at morpheme boundaries or by a purely count-based method such as byte-pair encoding (Sennrich, Haddow, & Birch, 2016c). *Post-processing* is applied after translation to reverse the pre-processing. This involves merging subword units into words, *detokenization* and *re-casing*. As all steps can introduce errors, this thesis applies minimal processing where possible. However, by applying special pre-processing steps, performance can improve significantly, as we show, for example, in Chapter 4.

We evaluate all end-to-end translation experiments using the BLEU score (Papineni, Roukos, Ward, & Zhu, 2002). The BLEU score computes the geometric mean of n-gram precisions p_n up to a maximum order N against one or more reference translations.[3] An

[2] `nist.gov/programs-projects/machine-translation`

[3] N is usually set to 4.

n-gram is a contiguous span of n tokens. The n-gram precision is modified to *clip* counts for each n-gram at the maximum count observed in one of the references. This clipping avoids, for example, overproduction of function words. As a precision-based metric would favor short translations, the BLEU score also contains a *brevity penalty* term, which is defined as

$$\mathrm{BP} = \exp\left(\min\left(0, 1 - \frac{|Y|}{|Y^*|}\right)\right)$$

for a set of translations Y and a set of reference translations Y^*.[4] The BLEU score is then computed as follows:

$$\mathrm{BLEU} = \mathrm{BP} \cdot \left(\prod_{n=1}^{N} p_n\right)^{\frac{1}{N}}$$

BLEU is corpus-based and aggregates *sufficient statistics* - n-gram counts and n-gram matches for each n, as well as output lengths for each sentence. As BLEU uses the geometric mean, it will automatically become 0, if $p_n = 0$ for one or more n. However, some training approaches require a metric, which can be calculated for single translations. The easiest approach to defining a *per-sentence approximation* to BLEU is to calculate a smoothed n-gram precision by adding 1 or a smaller value α to n-gram counts and matches. Problems with this simplistic approach and more sophisticated solutions on the sentence level have been put forward by Nakov, Guzman, and Vogel (2012).

2.2 Statistical Machine Translation

Statistical machine translation is used in all content chapters of this thesis (Chapters 3 - 6). This section summarizes the basics of SMT, following descriptions in Koehn (2010) and a survey by Lopez (2008).

2.2.1 Generative Model

SMT casts translating as solving the following problem:Given an input sentence x from a given *source language* (src), find the translation $\hat{y}$ in a given *target language* (trg) such that

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(y|x)\,. \tag{2.1}$$

In order to model this distribution in a tractable way, SMT models introduce *hidden variables*. These hidden variables, denoted by h, define a process by which a translation y

[4] $|Y^*|$ is the length of the reference *closest* to the candidate translation.

is created from an input sentence x. Hidden variables, or *derivations*, can be marginalized out to obtain the probability over translations:

$$\hat{y} = \operatorname*{argmax}_{y} \sum_{h} P(y, h|x)\,. \tag{2.2}$$

computing this sum is intractable in the translation formalisms applied in SMT. Therefore, an approximation is commonly made which replaces the sum over derivations with the highest scoring derivation:

$$\hat{y} = \operatorname*{argmax}_{y} \max_{h} P(y, h|x)\,. \tag{2.3}$$

In generative translation models, $P(y, h|x)$ is factorized using Bayes' rule as

$$P(y, h|x) = \frac{P(x, h|y) \cdot P(y)}{P(x)}\,. \tag{2.4}$$

Given this factorization, the translation problem can re-written as

$$\hat{y} = \operatorname*{argmax}_{y} \max_{h} P(x, h|y) \cdot P(y)\,, \tag{2.5}$$

where $P(x)$ can be ignored as it is constant. Factoring $P(y, h|x)$ into two separate models had been successfully applied to speech recognition before it was suggested for machine translation by Brown et al. (1993). The approach includes two independent knowledge sources: a *translation model* distribution $P(x, h|y)$, and a *language model* distribution $P(y)$. This factorization makes sense, as the translation model ensures *translational adequacy*, and the language model focuses on modeling target side *fluency*. Note, that the translation model direction is now reversed, conditioning on outputs rather than inputs. As the true distributions are unknown, they are approximated by models for which parameter estimation and inference are possible. This requires strong independence assumptions between the steps of the generative process. Below, two generative translation models, word alignment models and phrase-based models, are introduced, followed by an n-gram language model.

Word alignment The simplest translation models define the translation process via an *alignment function* $a(j)$ from positions j in x to positions in t in y.[5] According to this function, each token in y corresponds to exactly one token in x or to a special NULL token. The word alignment between a German source and an English target sentence is illustrated in Figure 2.1. A simple parametrization using alignment functions is IBM Model 1 (Brown et al., 1993). This model defines the probability of an output sentence

[5]This explanation now follows the reversed translation modeling direction as described above.

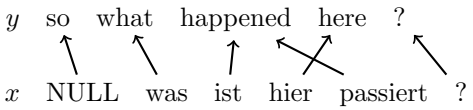


Figure 2.1 Word alignment

under alignment a as the product of the individual aligned tokens and a uniform alignment probability:

$$P(x, a|y) \approx \frac{\epsilon}{(|y|+1)^{|x|}} \prod_{j=1}^{|x|} p(x_j|y_{a(j)}), \tag{2.6}$$

where ϵ is a placeholder for a uniform distribution over the possible length of x (see Koehn (2010), page 86). The word alignment probabilities $p(x_j|y_{a(j)})$ are the only parameters in this model, as the model assumes independence between individual words. As there are usually no word-aligned corpora available, word alignments are inferred from sentence-aligned data in an unsupervised fashion using an expectation maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977). The EM algorithm learns the word translation probabilities that maximize the likelihood of the training corpus. IBM Model 1 and its refinements, IBM Models 2 through 5 (Brown et al., 1993), were quicky replaced by phrase-based translation formalisms for end-to-end translation systems. But they constitute an important step in the pipeline architecture of phrase-based models, as they are used to align training data at the word level, which is a pre-requisite for phrase extraction, as described below.

Phrase-based model The *IBM models* define only a uni-directional alignment function, allowing for one-to-many alignments, but not many-to-many. This makes the translation of non-decomposable phrases, such as idioms, difficult. Moreover, the word-based models do not easily capture the fact that larger chunks of a sentence often move their positions together when translated. These phenomena can easily be captured by *phrase-based translation formalisms*, where the units of translation are contiguous segments of variable length, or *phrases*. Deriving a translation can be viewed as the process of first selecting a segmentation of the source into phrases, then selecting a translation for each phrase from a *phrase translation table*, and finally *reordering* the phrase translations. Figure 2.2 illustrates the process.

A simple parametrization of the phrase-based translation process was given by Koehn et al. (2003). In this parametrization, the translation model probability is factored into the

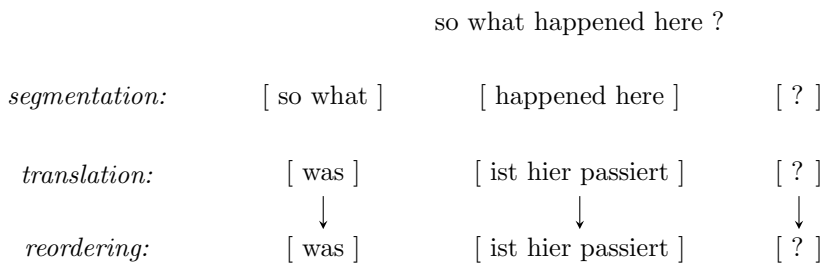


Figure 2.2 Phrase translation

phrase translation probabilities of all phrases participating in a derivation. Conditional independence is assumed between phrases. The full translation model is defined as

$$P(x, h|y) \approx \frac{1}{|\operatorname{Seg}(x)|} \prod_{i=1}^{I} p(\bar{x}_i|\bar{y}_i) \cdot d(start_i - end_{i-1} - 1)\,. \tag{2.7}$$

In this model, $\frac{1}{|\operatorname{Seg}(x)|}$ is a uniform probability distribution over all possible segmentations of x. $d(start_i - end_{i-1} - 1)$ is a distance-based reordering model, which captures how much phrases are moving from there position in the input sentence. Assuming that we prefer less movement, distortion can be modeled by an exponentially decaying cost function $d(z) = \alpha^z$ with $\alpha \in [0, 1]$. $\phi(\bar{x}_i|\bar{y}_i)$ is the probability of translating the i-th phrase $\bar{y}_i$ into its counterpart $\bar{x}_i$. The phrase translation probabilities are the parameters of this model, and are stored in a *phrase table*. A phrase table can be learned from word-aligned data using maximum likelihood estimation. Word alignment is run in both translation directions and the results are *symmetrized*. This symmetrization starts from the intersection of alignments and uses a heuristic method called `grow-diag-final-and` to add neighboring unidirectional alignment points (Koehn et al., 2003; Och & Ney, 2003). All phrases up to a *maximum phrase length*, which are *consistent* with the word alignment, are then extracted. A phrase pair is consistent with an alignment if and only if it contains at least one alignment link, and none of source or target words within the phrase are aligned to a word outside of the phrase. Phrase translation probabilities are then extracted using relative frequency counts:

$$p(\bar{x}|\bar{y}) = \frac{count(\bar{y}, \bar{x})}{count(\bar{x})}$$

Hierarchical phrase-based model Phrase-based translation cannot capture the regularities of long-range dependencies between words, as phrases need to be contiguous. In our previous example, *"happened"* translates to the German *"ist … passiert"*. A phrase-based model has no way to capture this dependency. These problems can be fixed by moving to a more powerful formalism, namely a synchronous context-free grammar. In this for-

malism, phrases are encoded as context-free grammar rules. A non-terminal symbol X is introduced to represent a gap which can be filled by another phrase of arbitrary length. A synchronous context-free grammar rule matching the example could be

$$X \rightarrow \textit{happened } X \mid \textit{ist } X \textit{ passiert}$$

In this formalism, a translation is derived by synchronously parsing the source sentence and generating the target sentence. The process is parameterized as

$$P(x, h|y) \approx \prod_{r \in h} p(r)\,, \tag{2.8}$$

where h is a synchronous parse tree by which x can be derived from y, and r is a hierarchical translation rule participating in this tree. The parameters of this model are the probabilities of the translation rules, r. These probabilities are again estimated using maximum likelihood estimation and stored in a *grammar*. Grammar extraction starts out from a *phrase table*. Hierarchical rules are generated by first adding all contiguous phrases as rules to the grammar. If a phrase pair contains another consistent phrase pair, the location of this smaller phrase pair can be replaced with a *gap* marked by the non-terminal symbol. If there is more than one gap, the non-terminal symbols are annotated with a position marker. Extracting hierarchical phrases is computationally challenging, requiring specialized algorithms and data structures (Lopez, 2007). A set of heuristics is used to keep the number of phrases manageable, such as limiting the number of allowed gaps to 2 or limiting the maximum phrase length (Chiang, 2005).

Language Model The second part of the generative decomposition of $P(y, h|x)$ is the *language model*, which approximates the true distribution over target strings, $P(y)$. As y is a string of tokens

$$y = y_1, y_2, y_3, \dots y_{|y|}\,,$$

the chain rule of probability can be used to decompose $P(y)$ as follows:

$$P(y) = P(y_1) \cdot P(y_2|y_1) \cdot P(y_3|y_1, y_2) \cdots P(y_{|y|}|y_1, y_2, \dots, y_{|y|-1})$$

In order to make the model tractable, conditional independence between a word and all but its $n-1$ preceding words assumed, leading to the following simplification and final parametrization of the model:

$$P(y) \approx \prod_{t=1}^{|y|} p(y_t|y_{t-n+1}, y_{t-n+2}, \dots, y_{t-1})\,, \tag{2.9}$$

In practice, the cases where $t < n$ are handled by adding $n-1$ beginning-of-sentence markers to each string. The parameters of this model are the probabilities of individual n-grams. These probabilities are also estimated from training data using maximum likelihood estimation. For example, the probability of the 4-gram "*was ist hier passiert*" would be estimated as

$$p(\text{passiert}|\text{was, ist, hier}) = \frac{count(\text{was ist hier passiert})}{count(\text{was ist hier})}\,.$$

In Equation 2.9, an unseen n-gram would be assigned a probability of zero, which would lead to the probability of the entire sequence being zero. In language models for SMT, unseen n-grams are assigned a non-zero probability via smoothing (see Chapter 7 in Koehn (2010)).

2.2.2 Discriminative Model

While the generative model would be theoretically optimal if the true distributions $P(x, h|y)$ and $P(y)$ were known, strong approximations have to be used in order to make it tractable, as has been described above. A much more flexible approach is given by using discriminative training, which directly optimizes the posterior distribution over outputs, $P(y, h|x)$ (Ney, 1995). In SMT, the discriminative approach commonly models the posterior in a log-linear or maximum entropy model, first proposed by Och and Ney (2002):

$$P(y, h|x) \approx \frac{\exp(\mathbf{w}^\top \phi(x, y, h))}{\sum_{y', h'} \exp(\mathbf{w}^\top \phi(x, y', h'))}\,, \tag{2.10}$$

where $\phi(x, y, h)$ is a vector of real-valued *feature functions* defined by the user and $\mathbf{w}$ is a weight vector, with one real-valued weight assigned to each feature. Inference in the log-linear model then becomes

$$\widehat{y} := \operatorname*{argmax}_y \max_h (\mathbf{w}^\top \phi(x, y, h))\,. \tag{2.11}$$

This model is more flexible than the generative model as it allows arbitrary components to be added in the form of additional feature functions.

In the log-linear model, the components of the generative model, the translation model log-probability, the language model probability, and the reordering cost, can be used as feature functions. Other feature functions include the reverse translation model log-probability, a *word penalty*(Och & Ney, 2002), which counts the number of produced output tokens, and *lexical weighting* features (Koehn et al., 2007), which consider the individual word translation probabilities within a phrase. A *lexicalized reordering model* (Tillmann, 2004)

can capture reordering decisions more precisely. As the above features have non-zero values for most inputs, we refer to them as "*dense features*". Other work has introduced so-called *sparse features*, which only take on a non-zero value in a few instances (Green, Cer, & Manning, 2014; Liang, Bouchard-Côté, Klein, & Taskar, 2006; Simianer, Riezler, & Dyer, 2012, inter alia). These features could, for example, be indicators that a certain n-gram has been observed on the source or target side or that a pair of source and target words have been observed together in a phrase.

The feature weights $\mathbf{w}$ are hyperparameters or meta-parameters of the translation model. Note that the generative components of the model, the phrase table and language model, are still trained generatively, and parameterized by phrase translation and n-gram probabilities. Meta-parameters are then trained discriminatively to assign weights to those models and other feature functions. By moving away from the generative model, these parameters can be optimized for an external error metric between a translation y and a reference y^*. The task of discriminatively optimizing meta-parameters for a specific translation quality (or error) metric is also called *tuning*. While reasonable weights for a small number of dense feature functions can be set by hand through careful experimentation, there is no guarantee that these weights are optimal. Setting weights for large, sparse feature sets is impossible to do manually and requires automatic optimization.

Minimum Error Rate Training (MERT) (Och, 2003) can be used to directly optimize error on the training set X, Y^*.

$$L_{error} = \text{err}\left(Y^*, \left\{\underset{y,h}{\text{argmax}}\, \mathbf{w}^\top \phi(x, y, h)\right\}_{x \in \boldsymbol{X}}\right), \tag{2.12}$$

where $\text{argmax}_y \max_h$ is abbreviated as $\text{argmax}_{y,h}$. For SMT, $1 - \text{BLEU}$ can be used to measure error. MERT uses a specialized line search over the k-best outputs generated by the current model to iteratively optimize each weight $w_i \in \mathbf{w}$ separately. As the objective contains an argmax operation and BLEU is not continuously differentiable, MERT has no guarantees for finding a globally optimal solution. The solution further depends on the initial values of $\mathbf{w}$. As MERT scales linearly with the number of features, it is computationally difficult to handle larger feature sets. Using larger feature sets with MERT is also prone to overfitting the training data, as the algorithm does not use any regularization. In this thesis, MERT is used for tuning dense feature sets in Chapters 3 and 4.

To counter the shortcomings of MERT, other tuning approaches have been suggested, which use different objectives. These objectives replace training set error with another loss function, which includes the error metric. The specific loss function is usually selected

because of its favourable mathematical properties such as smoothness or convexity, which allow for easier optimization and better extendability to large feature spaces. In this thesis we apply the pairwise ranking perceptron loss (Hopkins & May, 2011; Simianer et al., 2012) for tuning weights for large feature sets. Pairwise ranking losses conveniently reduce the optimization problem to classifying whether the ranking of any two translations y and y' according to the model is identical to their ranking according to the error metric, giving rise to the following criterion:

$$\text{err}(y^*, y) < \text{err}(y^*, y') \iff \mathbf{w}^\top \phi(x, y, h) > \mathbf{w}^\top \phi(x, y', h') \,.$$

Based on this condition, the loss is written as

$$L_{perc\_rank} = \sum_{\{y,h;y',h'\},\text{err}(y')>\text{err}(y)} \max\left(0, -\mathbf{w}^\top \Delta\phi(x, y, h, y', h')\right) \,, \tag{2.13}$$

where

$$\Delta\phi(x, y, h, y', h') := \phi(x, y, h) - \phi(x, y', h')$$

Following Collobert and Bengio (2004), a margin term can also be included in the perceptron loss for better generalization performance as

$$L_{perc\_rank} = \sum_{\{y,h;y',h'\},\text{err}(y')>\text{err}(y)} \max\left(0, 1 - \mathbf{w}^\top \Delta\phi(x, y, h, y', h')\right) \,. \tag{2.14}$$

The pairwise ranking perceptron is optimized using stochastic gradient descent. The preference pairs y, y' used to train the pairwise ranking perceptron are commonly generated from the list of k-best model outputs, $\mathcal{K}(x)$. As the number of possible pairs is quadratic in k, pairwise ranking optimizers often use heuristics to select pairs rather than generating all pairs. Unless otherwise specified, we use Simianer et al. (2012)'s scheme, which selects pairs from the top and bottom 20 percent of the n-best list. The pairwise ranking perceptron has been used for tuning log-linear models with large feature spaces in Chapters 4 and 6.

2.2.3 Decoding and Pipeline

In phrase-based models, solving Equation 2.11 is a highly complex search, or *decoding*, problem. In the *phrase-based model* a beam search algorithm is used which builds the translation left-to-right, keeping track of input coverage and n-gram histories in partial hypothesis states (Koehn, 2004). As an exhaustive search of all possible translations under the model is normally computationally infeasible, the search space is *pruned*, for example by only keeping the n hypotheses with the highest score for each number of input word

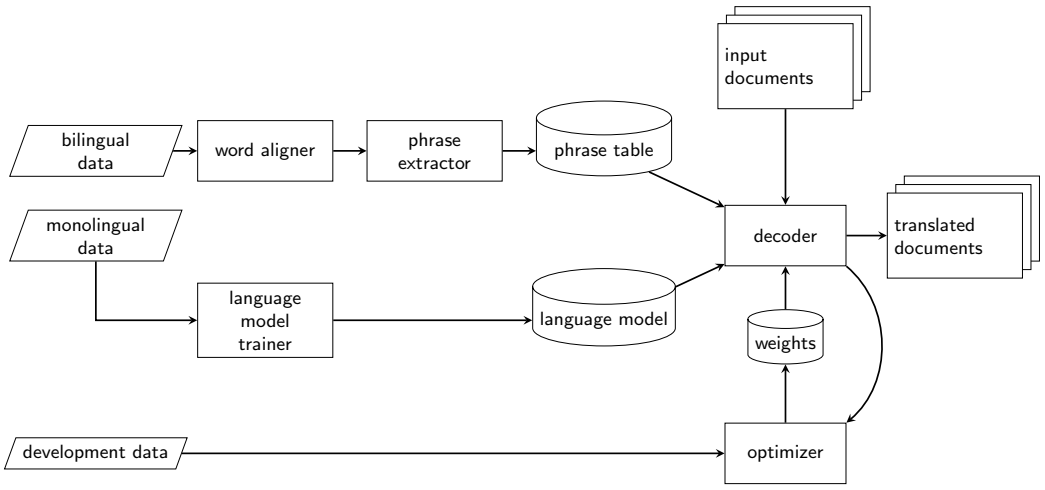


Figure 2.3 Phrase-based MT pipeline.

(Koehn, 2004). This approach is called *stack decoding*. For decoding in the hierarchical phrase-based formalism, dynamic programming algorithms from parsing, such as the CKY algorithm, can be transferred to machine translation. These algorithms build tree structures from the bottom up, covering larger input spans as they advance, in polynomial time. However, integrating n-gram language models becomes more challenging, as the output is no longer built left-to-right. *Cube pruning* (Chiang, 2007), which lazily combines only the most promising partial parses, makes language model integration computationally feasible.

Putting all steps together, one obtains a pipeline for phrase-based SMT systems. Figure 2.3 illustrates the data flows and steps necessary to build a complete phrase-based system. The translation model, a phrase table, or rule table in hierarchical phrase-based SMT, is trained by inferring word alignments from bilingual sentence-parallel data, and subsequently extracting phrase pairs. In open-vocabulary, open-domain translation, millions of aligned sentence pairs are necessary for good translations. The language model collects n-gram counts from monolingual target language data. For good performance, again, millions of sentences are required. When translation model and language model have been trained, feature weights of the log-linear model are learned to maximize performance on a small heldout data set of several thousand sentence pairs. This is accomplished using an iterative tuning process. Finally, input documents can be decoded using the trained models and weights. Phrase-based or hierarchical phrase-based models are employed in every part of the thesis. In Chapters 5 and 6 they are compared to neural machine translation models, which are discussed below.

2.3 Neural Machine Translation

Neural machine translation (NMT) has surpassed the performance of statistical machine translation (Jean, Firat, Cho, Memisevic, & Bengio, 2015; M.-T. Luong & Manning, 2015). This approach to translation is called *neural* as translation is modeled by a *non-linear artificial neural network*.[6] In this thesis we use the NEMATUS toolkit, which implements a recurrent encoder-decoder model with attention as proposed for machine translation by Bahdanau et al. (2015). Below, we describe the model architecture.[7] The exposition follows tutorials by Goldberg (2015) and Neubig (2017), as well as the text book by Goodfellow et al. (2016).

The recurrent NMT model is based on *recurrent neural language models* (RNN-LMs) (Mikolov, Karafiát, Burget, Černockỳ, & Khudanpur, 2010). Unlike the n-gram models discussed in the previous section, RNN-LMs are able to condition on the entire *history* of a word. Neural machine translation models extend the RNN-LM by also conditioning on the input sequence. By capturing the entire source context, as well as the complete target history, NMT models are better at capturing long-distance dependencies and producing long-range reordering than SMT models (Bentivogli, Bisazza, Cettolo, & Federico, 2016). In a recurrent NMT model, a translation $y = y_1 ... y_{|y|}, y_t \in V_{trg}$ where V_{trg} is a fixed size *output vocabulary*, is generated from left to right, one token at a time.The generation stops once an end-of-sequence symbol is produced or a maximum number of generation steps have been completed. Each token y_t is conditioned on the input sentence $x = x_1 ... x_{|x|}, x_j \in V_{src}$ where V_{src} is a fixed size *input vocabulary*, and all previously generated output words $y_1 ... y_{t-1}$. We abbreviate the output history by $y_{<t}$. The posterior probability of the entire translation y given input x is modeled as:

$$P(y|x) \approx \prod_{t=1}^{|y|} p_w(y_t|x, y_{<t}) \tag{2.15}$$

$p_w(y_t|x, y_{<t})$ is the output of a deep recurrent neural network. w represents the parameters of the network.

[6] An introduction to artificial neural networks and deep learning is beyond the scope of this thesis, but the reader is referred to the textbook by Goodfellow, Bengio, and Courville (2016).

[7] Recently, the convolutional sequence to sequence model by Gehring, Auli, Grangier, Yarats, and Dauphin (2017) and the transformer model by Vaswani et al. (2017) have emerged as successful alternatives to the architecture by Bahdanau et al. (2015), but we will focus on the models relevant for the work described in this thesis.

2.3.1 Translation Model

The core of the recurrent NMT model are two RNN-LMs, called an *encoder* and a *decoder*, which are trained jointly, and an *attention mechanism*, which connects the encoder and the decoder. The input is passed to the encoder via an additional input layer or *embedding layer*, and the distribution over the output vocabulary is generated by an output layer or *softmax layer*.

The *output layer* is a linear transformation of the hidden decoder state followed by a softmax operator which ensures that the outputs form a probability distribution over V_{trg}:

$$p_w(y_t|x, y_{<t}) = \text{softmax}(\mathbf{W}^{out}\mathbf{h}_t^{dec} + \mathbf{b}^{out})$$

where $\mathbf{W}^{out} \in \mathbb{R}^{|V_{trg}| \times |hidden|}$ and $\mathbf{b}^{out} \in \mathbb{R}^{|V_{trg}|}$ are parameters defining the output layer of the decoder, $|V_{trg}|$ is the target vocabulary size and $|hidden|$ is the *hidden layer* dimension, which is a user-defined hyperparameter of the model. The vector vector $\mathbf{h}_t^{dec} \in \mathbb{R}^{|hidden|}$ is computed by the decoder.

The *hidden decoder layer* is a recursively defined function, taking as input the previous hidden decoder state $\mathbf{h}_{t-1}^{dec}$ and a vector representation $\mathbf{y}_{t-1} \in \mathbb{R}^{|emb|}$ of the previous output y_{t-1}:

$$\mathbf{h}_t^{dec} = f^{dec}(\mathbf{y}_{t-1}, \mathbf{h}_{t-1}^{dec}) := \tanh(\mathbf{U}^{dec}\mathbf{y}_{t-1} + \mathbf{W}^{dec}\mathbf{h}_{t-1}^{dec} + \mathbf{b}^{dec})$$

where $\mathbf{W}^{dec}$, $\mathbf{U}^{dec}$ and $\mathbf{b}^{dec}$ are the parameters of the *hidden layer*. The decoder is initialized using a context vector $\mathbf{c}_0$ which is emitted by the input encoder.

The *hidden encoder layer* uses the same type of recursive function as the decoder, but with different inputs and parameters. A hidden input state $\overrightarrow{\mathbf{h}_j^{enc}}$ is computed as

$$\overrightarrow{\mathbf{h}_j^{enc}} = f^{enc}(\mathbf{x}_j, \overrightarrow{\mathbf{h}_{j-1}^{enc}}) := \tanh(\mathbf{U}^{enc}\mathbf{x}_j + \mathbf{W}^{enc}\overrightarrow{\mathbf{h}_{j-1}^{enc}} + \mathbf{b}^{enc}) \quad \text{with} \quad \overrightarrow{\mathbf{h}_0^{enc}} = \mathbf{0}\,.$$

Like the hidden decoder layer, for each input position $j \in 1...|x|$, the layer takes as input the previous *hidden encoder state* $\overrightarrow{\mathbf{h}_j^{enc}} \in \mathbb{R}^{|hidden|}$ and a vector representation of the j-th input word $\mathbf{x}_j \in \mathbb{R}^{|emb|}$. The parameters of the hidden encoder layer are $\mathbf{W}^{enc} \in \mathbb{R}^{|hidden| \times |emb|}$, $\mathbf{U}^{enc} \in \mathbb{R}^{|hidden| \times |hidden|}$ and $\mathbf{b}^{enc} \in \mathbb{R}^{|hidden|}$. In a simple NMT model without attention layer, the final encoder output $\overrightarrow{\mathbf{h}_J^{enc}}$ is used to initialize the encoder. A graphical representation of this simple model is shown in Figure 2.4.

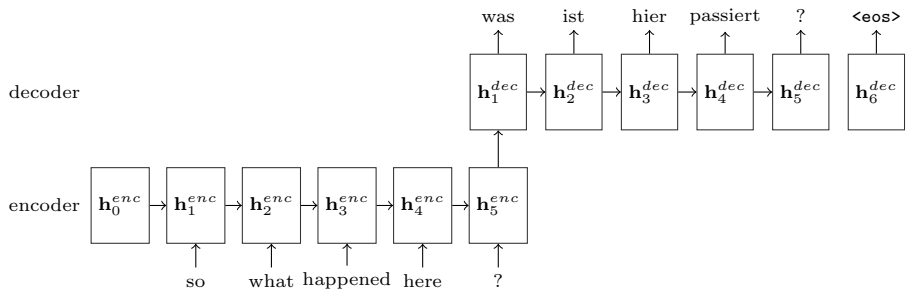


Figure 2.4 Recurrent sequence-to-sequence model.

f^{enc} and f^{dec} can be replaced by Long-Short-Term Memory (LSTM) units (Hochreiter & Schmidhuber, 1997) or Gated Recurrent Units (GRU) (Cho et al., 2014) in order to prevent the problem of vanishing gradients during training (Goodfellow et al., 2016, Section 10.10). These units are equipped with a *gating mechanism* that controls how much of the gradient is propagated to the next step. We use both types of units in our experiments and will specify the selected unit when describing the experimental setup.

The input and output vector representations, or *embeddings*, $\mathbf{x}_j$ and $\mathbf{y}_t$ are dense real-valued vectors with a dimensionality $|emb| \ll |V|$. Embeddings can be computed by applying a linear transform to a one-hot input encoding $\mathbf{r}$:

$$\mathbf{x}_j = \mathbf{E}\mathbf{r}_j \quad \text{where} \quad \mathbf{r}_j \in \{0,1\}^{|V|},\ \sum_{i=1}^{|V|} \mathbf{r}_j[i] = 1\,, \tag{2.16}$$

where the *embedding matrix* $\mathbf{E} \in \mathbb{R}^{|emb| \times |V|}$ is learned jointly with the other parameters of the model. As $\mathbf{r}_j$ is a one-hot encoding of the m-th item in V_{src}, Equation 2.16 corresponds to looking up the m-th column of $\mathbf{E}$.

In order to couple the beginning of the source and target sequence more closely, Sutskever et al. (2014) proposed to encode the input sentence in reverse order. The reverse hidden encoder states are computed as:

$$\overleftarrow{\mathbf{h}_j^{enc}} = f(\mathbf{x}_j, \overleftarrow{\mathbf{h}_{j+1}^{enc}}) = \tanh(\mathbf{U}^{enc}\mathbf{x}_j + \mathbf{W}^{enc}\overleftarrow{\mathbf{h}_{j+1}^{enc}} + \mathbf{b}^{enc})\,.$$

This method has been extended further to a *bidirectional encoder*, which performs both a left-to-right and a right-to-left pass over the input and concatenates both hidden states as

$$\mathbf{h}_j^{enc} = [\overrightarrow{\mathbf{h}_j^{enc}}); \overleftarrow{\mathbf{h}_j^{enc}})] \quad \text{with } \mathbf{h}_j^{enc} \in \mathbb{R}^{2|hidden|}$$

where [ ] represents vector concatenation.[8] A closer coupling between encoder and decoder can be achieved through an *attention mechanism*. This mechanism computes a *context vector* $\mathbf{c}_t$ at each output step t as a weighted sum over the hidden encoder states with *attention weights* $\alpha_{j,t}$:

$$\mathbf{c}_t = \sum_{j=1}^{J} \alpha_{j,t} \cdot \mathbf{h}_j^{enc} \quad \text{with } \mathbf{c}_t \in \mathbb{R}^{|hidden|}$$

The attention weight vector α_t is computed by taking a softmax over unnormalized attention scores $a_{j,t}$. These are computed by applying an additional non-linear layer to the hidden encoder and decoder states[9]

$$a_{j,t} = (\mathbf{w}^{att\top} \tanh(\mathbf{W}^{att}\mathbf{h}_j^{enc} + \mathbf{U}^{att}\mathbf{h}_t^{dec} + \mathbf{b}^{att}))\,,$$

where $\mathbf{W}^{att}, \mathbf{U}^{att} \in \mathbb{R}^{|hidden|\times|hidden|}$ and $\mathbf{b}^{att}, \mathbf{w}^{att} \in \mathbb{R}^{|hidden|}$. Through an attention layer, the coupling between decoder and encoder now happens at every each time step. $\mathbf{h}_t^{dec}$ is now computed depending on the previous context vector:

$$\mathbf{h}_t^{dec} = f^{dec}(\mathbf{y}_{t-1}, \mathbf{c}_{t-1}, \mathbf{h}_{t-1}^{dec})$$

The output layer then uses the current context vector:

$$p(y_t|y_{<t}, x) = \text{softmax}(\mathbf{W}^{out}\mathbf{h}_t^{dec} + \mathbf{U}^{out}\mathbf{c}_t + \mathbf{b}^{out})\,.$$

2.3.2 Training

NMT models are standardly trained to minimize the *cross entropy* of the model prediction $\hat{y}$ and a reference y^*, which corresponds to maximum likelihood estimation. If we assume that at each step t the reference token y_t^* has a probability of 1 and all other token in V_{trg} have a probability of 0, minimizing cross entropy between this distribution and the model distribution amounts to minimizing the *negative log-likelihood* of the reference under the model:

$$L_{MLE}(y^{(i)}) = -\frac{1}{|y^*|}\sum_{t=1}^{|y^*|} \log(p_w(y_t^* \mid x, y_{<t}^*))\,.$$

[8] Note that if a concatenation of hidden states is generated by the encoder, either the hidden layer dimensionality $|hidden|$ needs to be adjusted in the decoder to match the hidden encoder dimension, or an additional transformation needs to be applied to the encoder output.

[9] This description follows the original description of attention in Bahdanau et al. (2015), which is implemented in the NEMATUS toolkit used in this work (Sennrich et al., 2017).

The model parameters w are optimized using *stochastic gradient descent* (SGD). As an *online learning* algorithm, SGD performs an update after one single training example, as opposed to *batch* algorithms, which only update after seeing the entire finite training set. Online algorithms are used to train NMT models, because the size of the training set precludes the application of a batch algorithm. However, a *minibatch* version of SGD with minibatch size $M : 1 < M \ll |X|$ leads to more stable learning and allows to exploit hardware parallelism (see Goodfellow et al. (2016), Section 8.1.3). Algorithm 1 shows SGD in a *minibatch* version for the MLE objective. More advanced algorithms such as ADADELTA (Zeiler, 2012) or Adam (Kingma & Ba, 2015) compute a per-coordinate learning rate based on previous updates, allowing to control the rate at which each parameter is updated. This thesis uses both SGD and more complex optimizers.

Algorithm 1 SGD

Require: Learning rate η, minibatch size M
Require: Initial parameters w
while Stopping criterion is not reached **do**
  Sample minibatch of M examples $(x_1, y_1), ...(x_M, y^*_M)$
  $g \leftarrow \frac{1}{M} \nabla_w \sum_{m=1}^{M} L_{MLE}(y^*_m)$ ▷ compute gradient estimate
  $w \leftarrow w - \eta g$ ▷ apply update
end while
return w

The gradients for a neural network with respect to each parameter are computed automatically using the *back-propagation* algorithm (Rumelhart, Hinton, & Williams, 1986). This algorithm provides a dynamic programming method for automatic differentiation with respect to the different parameters. It performs calculations by representing the neural network as a *computation graph*. In this directed acyclic graph, each node represents a calculation, with the children of a node representing the arguments. The network is applied to an input by a *forward* pass over the computation graph. A *backward* pass computes the derivatives, given the forward computation. Building computation graphs and performing forward and backward passes can be easily done using libraries such as `theano`[10], `pytorch`[11] or `tensorflow`[12]. The training algorithms above can be applied to any neural network architecture. In order to compute gradients for recurrent neural networks using back-propagation, the recursive node is *unrolled* over all time steps. The gradients are then computed backwards for each time step, a procedure called *back-propagation through time* (BPTT). As propagating error through very deep networks can be subject to vanishing or exploding gradient problems, we use LSTM or GRU layers and clip the gradients if their norm is larger than a given threshold (Pascanu, Mikolov, & Bengio, 2013).

[10] `deeplearning.net/software/theano`. Note that `theano` is no longer being developed, but has been included as it has been used in parts of this dissertation.
[11] `pytorch.org`
[12] `tensorflow.org`

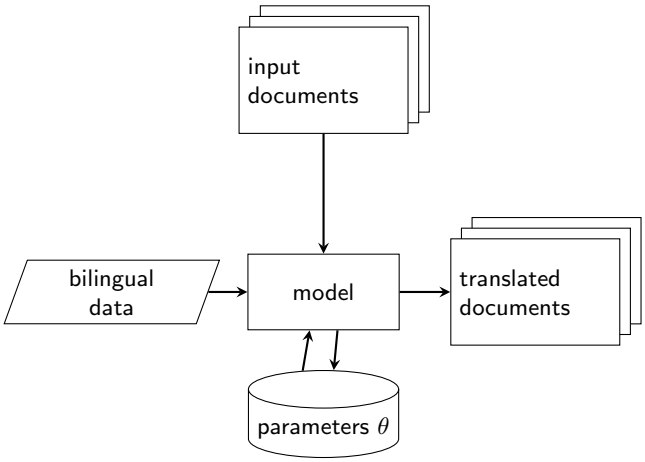


Figure 2.5 The NMT pipeline

2.3.3 Decoding and Pipeline

Decoding in an NMT model tries to find the sequence $\hat{y}$ such that

$$\hat{y} = \underset{y}{\operatorname{argmax}} \prod_{t=1}^{|y|} p_w(y_t|x, y_{<t}) .$$

The output sequence is generated step-by-step until the end-of-sentence marker is generated or a maximum step size is exhausted. The simplest, *greedy* approach to decoding, outputs the most likely word at each time step. Using *beam search* instead of greedy search allows the model to explore a larger search space or to generate the n most likely hypotheses, where n is the *beam size*. In this approach, at $t+1$, the model expands each of the current hypotheses with the n most probable words, but only keeps top n hypotheses with the highest total score. By avoiding the limiting independence assumptions of the SMT models, NMT models have surpassed SMT in benchmark evaluations and become the state of the art at the time this thesis was written. Using a recurrent encoder-decoder architecture with attention allows for a much simpler training protocol compared to the SMT pipeline, as shown in Figure 2.5. A drawback of the NMT architecture is its high computational cost requiring specialized hardware (Graphical Processing Units) and its need for large training data to achieve good performance. These factors make SMT still an attractive choice for low-resource settings.

CHAPTER 3

Exploiting Meta-Information for Task-Specific Parallel Data Mining

In-domain supervision is crucial when adapting a translation model to a new, known domain. But what if there is no full supervision available for the target domain? A known approach to this problem is to extract pseudo-parallel sentences or smaller fragments from weaker sources of supervision, such as comparable corpora. In this case, document metadata or cross-lingual document-level links can play a crucial role in identifying candidate sentence pairs. This chapter studies translation of microblog messages as a previously unexplored task. We present a tailored approach to the topically constrained extraction of pseudo-parallel microblog messages for an Arabic-English translation task using a cross-language information retrieval model. Document meta-information present in the messages is used to identify topically relevant messages and as an additional source of information in the retrieval model.

Our work starts out from the observation that information and comments about events of global significance will dissipate on public microblogging platforms like Twitter[1] simultaneously and multilingually. Being able to access content from different languages or to automatically make content available in different languages, could provide a valuable service to those involved in the event, as well as international organizations or aid workers. Twitter messages (tweets) are challenging to translate automatically, mainly because

[1] `https://twitter.com/`

the length restriction and informal register of tweets give rise to creative and varied uses of written language. Non-standard spelling and punctuation are used intentionally for pragmatic purposes, such as emphasis or expressing a user's emotion or the strength of their opinion. Abbreviations, colloquial expressions and syntactic variation present further challenges. For a detailed analysis of linguistic variation in social media and their challenges for Natural Language Processing (NLP) applications, see e.g. Eisenstein (2013). Language use on twitter can differ greatly from the formal register used in machine translation training sets from parliamentary proceedings or news articles, thus causing a large percentage of out-of-vocabulary terms (OOVs).

Twitter's Streaming API enables users to access large amounts of Twitter messages via a crawler. These messages are presented as structured data. They contain metadata such as time stamps, a user's location, gender, and other metadata, which users can choose to disclose. They also contain markup from within the text. Users can link to URLs, annotate their messages with topic or discourse markers, so-called *hashtags* (marked by a preceding '`#`'), or mention other users (marked by a preceding '`@`') . When referring to the same event, the same hashtags are often used across languages. Given the observation that users are spreading the same information about globally significant events in more than one language and that they use the same annotations, we hypothesize that some of the messages will be parallel or nearly parallel. This chapter describes first describes a method for identifying nearly parallel tweets in a topically constrained setting using a tailored cross-language information retrieval model. Second, various methods are described and evaluated for integrating the nearly parallel tweets with a phrase-based statistical machine translation (SMT) system trained on out-of-domain data. The effectiveness of the approach is demonstrated on the task of Arabic-English translation of tweets related to the aftermath of the Arab Spring.

This chapter presents work published in Jehl et al. (2012). This work was conducted jointly with the co-authors of the paper. As the main ideas were being developed collaboratively, they will be included here. Concerning implementation and modeling details, the chapter will focus on the work conducted by the author of this thesis and summarize briefly any relevant portions of the work which were mainly conducted by co-authors. While co-developing the overall approach and having input on all parts of the work, the author of this thesis was mainly responsible for: (1) Strategies for the extraction of phrases from noisy pseudo-parallel data. (2) A development set of manually translated tweets, collected via a crowdsourcing task, along with a description and analysis of the task. (3) An extensive series of experiments which evaluate various adaptation strategies on the Twitter translation task, including our custom strategies. (4) A quantitative and qualitative analysis of the different systems. The rest of this chapter is structured as follows: Section 3.1 gives an overview of related work. Section 3.2 describes our approach. Section 3.3

describes and analyzes the generation of an in-domain evaluation set via crowdsourcing. Section 3.4 specifies our experimental setup and presents the results. In Section 3.5 a quantitative and qualitative analysis of results is conducted.

3.1 Related Work

In recent years, Twitter has become a genre of interest to many parts of the NLP community. This has led to a large volume of publications and a dedicated annual workshop on Natural Language Processing for Social Media, which was first held in 2013.[2]

The potential of using social media content for disaster relief, one of the motivations of our work, has also been explored, for example, in the 2013 IJCNLP workshop on Language Processing and Crisis Information.[3] This workshop focused mostly on the Tōhoku earthquake of 2011, and contributors did not use multilingual data. The Haitian Creole translation task at WMT 2011[4] targeted the translation of short text messages, a domain similar to microblog messages, in the aftermath of the 2010 earthquake in Haiti. Like us, participants of this task faced a difficult domain with very little in-domain training data (17,192 sentences). The proposed solutions included the use of crowdsourcing (Hu et al., 2011) or parallel data extraction from comparable data (Hewavitharana, Bach, Gao, Ambati, & Vogel, 2011).

Automatically extracting parallel data from multilingual document collections has also been of general interest to the SMT community. Early research has focused on the extraction of parallel data from newswire (Fung and Cheung (2004); Munteanu and Marcu (2005); Tillmann and Xu (2009)). More recent work has targeted large-scale crawled web data (Smith et al. (2013)) or Wikipedia (Gupta, Pal, and Bandyopadhyay (2013); Ture and Lin (2012); Wołk and Marasek (2015)). While the approaches above focus on aligning parallel sentences in comparable documents, other work has tackled the task of extracting parallel fragments in non-parallel sentences (Bakhshaei, Safabakhsh, and Khadivi (2019); Cettolo, Federico, and Bertoldi (2010); Munteanu and Marcu (2006); Quirk, Udupa, and Menezes (2007); Vogel and Hewavitharana (2011)). Munteanu and Marcu (2006) is closest to our work, since they also use word-based cross-language information retrieval.

Our work, which was published in 2012, is the first to venture into the task of translating tweets. Since then, other researchers have taken up this task. While we use CLIR to iden-

[2] https://sites.google.com/site/socialnlp2016/
[3] https://sites.google.com/site/lpci2013workshop/
[4] http://www.statmt.org/wmt11/featured-translation-task.html

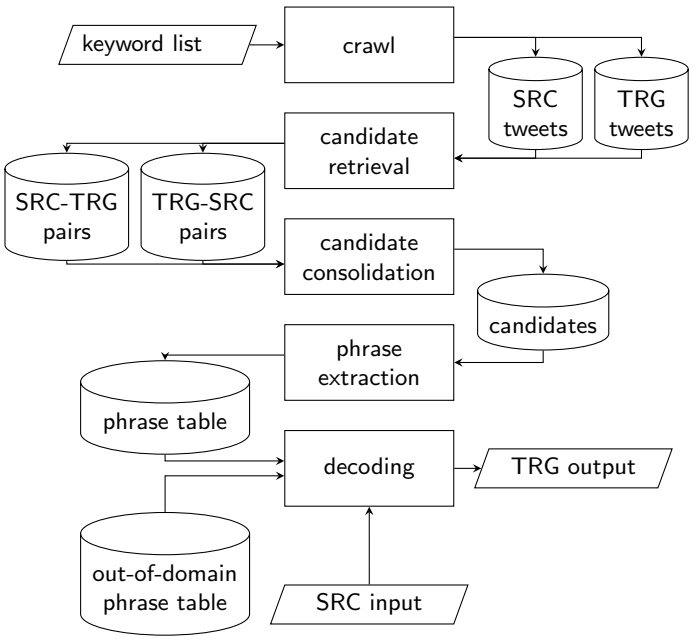


Figure 3.1 Pipeline for building an adapted Twitter translation system.

tify nearly parallel tweets across languages, Ling, Xiang, Dyer, Black, and Trancoso (2013) extract in-domain training data by identifying bilingual users on Twitter or Sina Weibo and extracting parallel segments from single tweets. Like us, they also use crowdsourcing to create an in-domain test set (Ling, Marujo, Dyer, Black, & Trancoso, 2014). In 2015, a shared task on translating tweets was organized (see Alegria et al. (2015)).[5] The test data for this task was obtained both by identifying bilingual users and by crowdsourcing translation of tweets.

3.2 Approach

This section describes our approach to extracting topically constrained pseudo-parallel tweets and to integrating these tweets with an existing out-of-domain translation system. As the approach follows a pipeline architecture, the first part of the section surveys the entire pipeline. The subsequent parts contain detailed descriptions for more complex steps.

3.2.1 Pipeline Overview

Our goal is to build a domain-specific Twitter translation system which can be used for translating tweets about events or developments of global significance. Twitter is different from other comparable corpora, because we do not have reliable document-level alignments. However, hashtags, user mentions and URLs provide us with a weak guiding signal. Furthermore, for most comparable corpora, document alignment and sentence alignment are two separate steps, and sentence alignment is often non-trivial. Since tweets cannot exceed 140 characters,[6] we do not differentiate between document and sentence level and thus do not require a separate sentence alignment step. In line with our motivation to build translation systems for events of global significance, we tailor our approach to the extraction of topically constrained tweets, where the topic is defined by a user-supplied keyword list. Our 5-step pipeline is shown in Figure 3.1. Below, we outline each step.

Step 1: Crawl. We start by creating a database of topically relevant Twitter messages. The database is constructed via a keyword-based crawl of the Twitter Streaming API. Keywords can be constructed by starting a test crawl with a seed set of relevant person or place names and tracking frequent terms within the crawl to find common spelling variations or additional important terms. In this step, we take advantage of within-text annotations (hashtags and usernames). These are often in English, even if the rest of the tweet uses another language.

Step 2: Candidate retrieval. After filtering and de-duplication of the crawled tweets, we search for nearly parallel pairs of tweets within the database. This is accomplished by performing cross-lingual information retrieval using tweets in one language as a query to retrieve similar tweets in the other language. Again, this step makes use of within-text annotations, as hashtags, usernames and URLs are used across languages.

Step 3: Candidate consolidation. The cross-lingual retrieval model provides a ranking of target language tweets for each source language query. After retrieval, a consolidation step uses this ranking information to create the final set of pairs. We explore different consolidation strategies.

[5] `http://komunitatea.elhuyar.eus/tweetmt/`

[6] This restriction has been changed to 280 characters in 2017 (Rosen, 2017).

Step 4: Phrase extraction. In the 4th step, we extract an in-domain phrase table for adapting an existing SMT model from the accepted pairs. We try out two different strategies for phrase table extraction.

Step 5: Decoding. For decoding, the extracted phrase table is integrated with the existing system. We look into different phrase table combination strategies and compare our approach to other adaptation methods.

In our experiments, step 1 ran over a period of several months. In a case of urgency, our setup could be used in an iterative fashion, starting out from a small database, which is then updated as more messages come in. Steps 2 to 5 could then be performed in regular intervals.

3.2.2 Database Generation and Cross-lingual Pairing of Tweets

The first two steps of our method, database generation and candidate retrieval, were implemented by Felix Hieber. We therefore focus on the description of the retrieval model and the most relevant implementation details. The reader is referred to Jehl et al. (2012) for more details about the implementation.

Retrieval model. We apply a word-based probabilistic cross-language information retrieval model to construct candidate pairs of English and Arabic tweets. This model combines the generative probabilistic cross-lingual retrieval model of Xu, Weischedel, and Nguyen (2001) with the concept of self-translation as formulated by Xue, Jeon, and Croft (2008). In this context, self-translation refers to a translation model mapping a source term onto itself in the target output. The retrieval score between a query $q = q_1, \ldots, q_{|q|}$ from a source language query collection Q and a document $d = d_1, \ldots, d_{|d|}$ from a target language document collection D is defined as the likelihood of d having generated q as follows:

$$P(q|d) = \prod_{q_i \in q} P(q_i|d) \tag{3.1}$$

$$= \prod_{q_i \in q} \lambda P_{translation}(q_i|d) + (1-\lambda) P_{backoff}(q_i|Q) \tag{3.2}$$

$$P_{translation}(q_i|d) = \beta \sum_{d_j \in d} P_{TM}(q_i|d_j) P(d_j|d) + (1-\beta) P_{self}(q_i|d) \tag{3.3}$$

Equation 3.1 defines the probability of generating q as the product of the probabilities of the individual query terms q_i given document d, showing that the model treats query terms independently. Equation 3.2 then defines the query term probability as a linear interpolation between a *translation* probability defined with respect to q_i and d, and a *backoff* probability. The backoff probability is computed between q_i and the collection of queries in the source language, Q. It acts as a smoothing distribution to avoid zero terms in the product. Finally, Equation 3.3 defines the *translation* probability between q_i and d. This distribution is also defined as a linear interpolation between two distributions. The first term defines the probability of q having been generated by d through a translation process. As it is unknown which document term generated a query term, this probability is computed by taking the expectation over all document terms. $P_{TM}(q_i|d_j)$ is the *T*ranslation *M*odel probability of query term q_i given document term d_j. This probability is weighted by the probability of term d_j in d. The second interpolation term, the *self-translation probability* $P_{self}(q_i|d)$, represents the probability of q_i being directly generated by d without a translation step. The self-translation probability is an important component of the model, as it captures tokens which are identical across languages, such as hashtags, user mentions, and URLs. As these tokens are very unlikely to appear in a word translation table from out-of-domain data, their translation probability will likely be zero. The self-translation probability will still assign some probability mass to occurrences of, e.g., the same hashtag across tweets. The distributions $P_{backoff}(q_i|Q)$, $P(d_j|d)$ and $P_{self}(q_i|d)$ are each estimated via relative frequency estimation on monolingual in-domain data. $P_{TM}(q_i|d_j)$ requires a word translation model, which can be estimated, for example, by running one or more of the IBM models (Brown et al., 1993). The interpolation weights λ and β are hyperparameters to be specified by the user.

Implementation notes. Twitter messages were crawled over a period of 4 months from September 2011 until January 2012 via the Streaming API[7] in keyword-tracking mode, using a list of keywords related to the events of the Arab spring. The database consisted of 25.5 million Twitter messages. After filtering out languages other than English and Arabic[8] and removing tweets repeated by users (called *re-tweets*), 6.6 million Arabic and 5.1 million English tweets remained. In order to find nearly parallel pairs of tweets, the system performs a cross-language information retrieval step using the retrieval model described above, where tweets in one language are treated as queries and tweets in the other language as documents. This step is performed in both directions, Arabic → English and English → Arabic. λ and β were selected by grid search on our in-domain development set (see Section 3.3, as well as Section 5.2 in Jehl et al. (2012)) and set to 0.9. Computing

[7] `https://dev.twitter.com/docs/streaming-api/`

[8] Language filtering was done using a Naive Bayes classifier implemented in the Java language detection library by Nakatani Shuyo `https://github.com/shuyo/language-detection`

consolidation	# extracted pairs
`top3`	14.8 million
`top1`	5.1 million
`top1 + intersect`	3.5 million

Table 3.1 Extracted sentence pairs for different consolidation strategies.

retrieval probabilities over the Cartesian product of the Arabic and English tweets was made possible by the use of an efficient Lucene[9] index for storing pre-computed word probabilites and running the model on a Hadoop[10] cluster. The retrieval step produces a ranking of target language tweets for each source language tweet.

3.2.3 Candidate Consolidation Strategies

In this step, we condense the rankings produced by the retrieval step into a set of accepted bilingual pairs. Due to the shortness of tweets, we do not need to perform sentence alignment. Instead, each accepted pair of tweets can be treated as a "parallel" sentence pair. The first strategy, `topN`, adds the N highest-ranking tweets paired with the query to the set of accepted pairs. The second strategy, `intersect`, requires retrieval in both language directions. The consolidation step then returns the intersection between the `topN` results in both directions. Table 3.1 shows example counts of extracted pairs for the different strategies. The `top1+intersect` strategy reduces the number of returned sentences by 31%.

3.2.4 Phrase Extraction Strategies

Since the retrieval-based pairing is noisy, we first pursue a conservative approach to extracting phrases from pseudo-parallel Twitter data (`conservative`). The conservative strategy is inspired by the work of Munteanu and Marcu (2006). They extract parallel fragments from comparable sentences by starting out with alignment points from a clean lexicon and then applying a smoothing filter to extract longer fragments. Our approach constructs word alignments for the pseudo-parallel candidates using word alignment tables from the out-of-domain translation model. We then apply the `grow-diag-final-and` heuristic (Koehn et al., 2003) to these initial alignments and use Och and Ney (2004)'s

[9] `lucene.apache.org`
[10] `hadoop.apache.org`

consolidation	extraction	# extracted phrases
`top3`	`conservative` (`bidirectional`, $m = 3$)	7 million
`top1`	`bold`	54 million
`top1+intersect`	`bold`	29 million

Table 3.2 Phrase table sizes for bold and conservative extraction strategies.

phrase extraction algorithm to construct an in-domain phrase table. This approach is similar to the smoothing approach mentioned before, as unaligned words from the vicinity of the initial alignments points will be added to the phrases by the extraction algorithm.

Constraints on the translation table and the minimum required alignment points can make this approach more or less restrictive: Experiments can either use alignments from unidirectional translation tables (`unidirectional`) or, in an even stricter setup, use a bidirectional word translation table (`bidirectional`), which only contains word pairs occurring both in the Arabic → English and English → Arabic word translation tables. Candidate pairs can further be filtered according to the number of initial alignment points, requiring each accepted pair to have at least m alignment points. The conservative strategy is tested with different consolidation settings up to `top10`.

The conservative phrase extraction strategy is restrictive, because it only learns new words, which occur in the proximity of known words from the out-of-domain model. Therefore, this strategy is contrasted with a bolder strategy (`bold`). The bold strategy treats the extracted candidate pairs like a parallel data set and runs EM-based word alignment from scratch. In order to combat noisy phrases and to keep data size manageable, the bold strategy is only applied to candidate sets generated with the `top1` and `top1+intersect` consolidation approaches. For all experiments, only phrases up to length 3 are extracted from the candidate pairs (the default is 7) in order to avoid learning too much noise.

Table 3.2 shows examples of extracted phrase table sizes using the bold and conservative strategies. The conservative strategy extracts much fewer phrases than the bold strategy, even though the bold strategy uses `top1` consolidated pairs and the conservative strategy uses `top3` consolidated pairs. Replacing `top1` consolidation with `top1+intersect` for the bold strategy reduced the phrase table size by 46%, while still producing 4 times more phrases than the conservative strategy.

3.3 Evaluation Data Construction

In order to measure performance of our approach, a small in-domain evaluation set is required. As we did not have access to professional translators, this evaluation set was created using Amazon Mechanical Turk[11]. In setting up the task we followed the exploratory work of Zaidan and Callison-Burch (2011b) on acquiring translations via crowdsourcing. This section describes the task setup as well as an analysis of the performance by the "turkers" (workers on Amazon Mechanical Turk).

We randomly set aside 2,000 Arabic tweets from the filtered crawl for manual translation. Hashtags, user mentions and URLs were removed from each message beforehand, because they do not require translations and would just artificially inflate BLEU scores at test time. Additional cleaning and filtering was done manually. We discarded messages which contained very little text or large portions of other languages, and removed any remaining Twitter markup. 1,029 tweets were retained as input to the Mechanical Turk task. We split the data into batches of ten sentences. Each batch then made up one HIT (*human intelligence task*). We require each HIT to be completed by three different workers. In order to have some control over translation quality, we inserted one control sentence per HIT, which was taken from the LDC-GALE Phase 1 Arabic Blog Parallel Text. We selected this data set because of its similarity to the Twitter domain. Workers were payed 10 cents per translation. Following the recommendation of Zaidan and Callison-Burch (2011b), all Arabic sentences were converted into images in order to prevent workers from pasting them into online translation engines. The instructions asked translators to insert an "*unknown*" token whenever they were unable to translate a word. Additionally, the HIT setup did not allow workers to skip a sentence, forcing them to complete an entire batch.

The HITs were completed by 38 workers in total. Most HITs were completed within three days. Workers who entered random symbols were easily identified manually, as they took only few seconds per HIT. These worker's submissions were rejected and the workers banned from completing more HITs. Figure 3.2 shows the number of completed HITs per worker. It can be seen that most workers completed only few HITs: There were 26 workers who completed five HITs or less, while twelve workers completed more than five HITs, and only four workers completed more than 20 HITs. Worker 22, who completed 70 HITs, is an outlier. Figure 3.3 on page 38 shows the average time each worker spent on each HIT. We can see that worker 22 spent 500 seconds per HIT, while others took six times as long. This low average time indicates that worker 22 invested less effort than others to produce translations. For future crowd-sourcing experiments, requiring workers

[11] `http://www.turk.com`

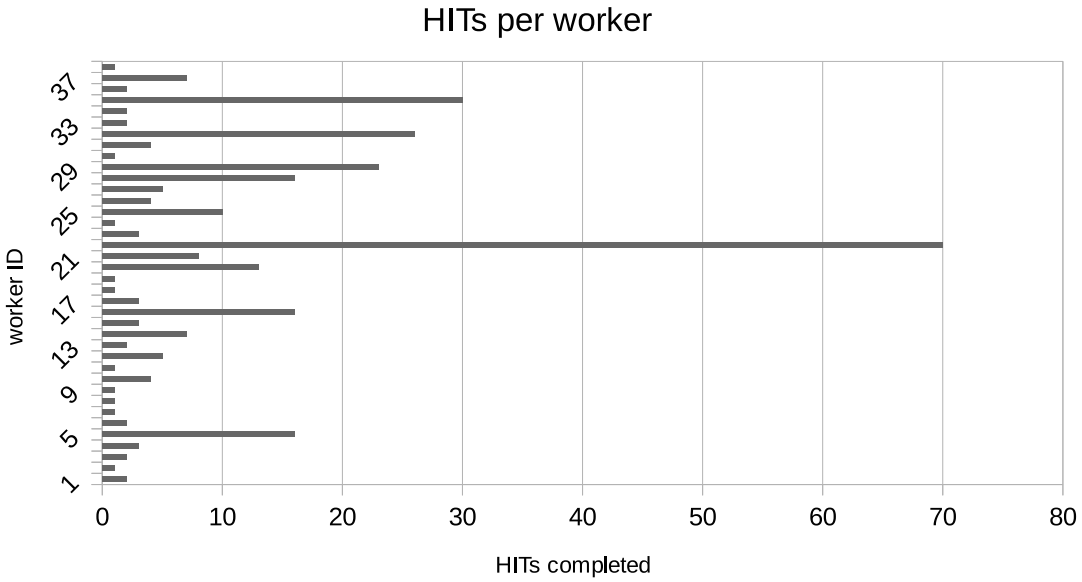


Figure 3.2 Number of HITs completed per worker.

to spend a minimum time on each HIT could help with potential low-effort translators. Alegria et al. (2015) automatically rejected workers who took less than 150 seconds to translate ten tweets. With an average of 500 seconds per HIT, worker 22 is still far above this threshold. On inspection, it was decided to accept the submissions by this worker, since the translations were of inferior quality, but were clearly better than random input. By collecting three translations per input, we allowed for some translation mistakes to be balanced out by the other translations. Table 3.3 on page 38 shows example translations for one of the quality control sentences, along with the reference translation by a professional translator. The example illustrates the tendency of translators to make grammar mistakes or odd word choices. Translators also tended to omit punctuation marks. However, some variations in the translations was caused by reasonable translation alternatives (such as "*gathered*" and "*collected*" in the example).

Our final evaluation data set consisted of 1,029 tweets with three translations each. In order to account for translation variants, all three translations obtained via Mechanical Turk are used as references for each input tweet. We randomly split our small parallel corpus, using half of the tweets for development and half for testing.

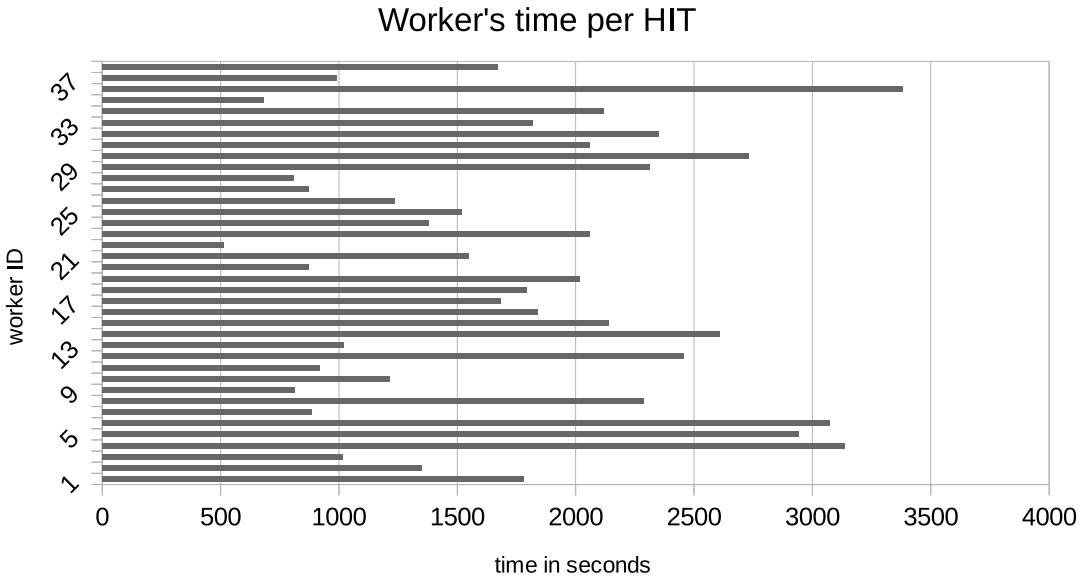


Figure 3.3 Average time per HIT per worker.

Reference	*breaking the silence, a campaign group made up of israeli soldiers, gathered anonymous accounts from 26 soldiers.*
Translation 1	*and breaking silence is a group of israeli soldiers that had unknown statistics from 26 soldiers israeli*
Translation 2	*breaking the silence by a group of israeli soldiers who gathered unidentified statistics from 26 israeli soldier.*
Translation 3	*breaking the silence is a group of israeli soldiers that collected unknown statistics of 26 israeli soldiers*

Table 3.3 Example of crowd-sourced translations.

3.4 Experiments

This section describes the out-of-domain baseline system, as well as various adapted baselines using only monolingual Twitter data and the crowd-sourced development set. We then compare these results to the results obtained by using various configurations of our approach described above, which uses the automatically extracted pseudo-parallel data.

3.4.1 Experimental Setup

Baseline SMT system. A baseline model was trained using 5.8 million parallel sentences in Modern Standard Arabic (MSA) and English from the 2009 NIST Open Machine Translation evaluation campaign[12]. This data contains parallel text from different domains, including UN reports, newsgroups, newswire, broadcast news and weblogs. The English side of the data was used to train a 5-gram language model. All experiments were conducted using the Moses machine translation toolkit[13] (Koehn et al., 2007) with standard settings. Language models were built using the SRILM toolkit[14] (Stolcke, 2002). The models used the standard dense features implemented in Moses (language model, 7 distortion model weights, 5 translation model weights and word penalty). Feature weights were tuned using MERT (Och, 2003). The baseline weights were optimized using the development data from the NIST evaluation (696 segments with 4 references).

Pre-processing. We applied several pre-processing steps to the Arabic and English data. To avoid sparsity, we normalized all digits by replacing them with one single digit, but preserving the structure of numerical expressions. All Arabic sentences were transliterated using the Buckwalter Arabic transliteration scheme[15] to avoid encoding difficulties. Transliteration was done using the MADA toolkit (Habash & Rambow, 2005). For Arabic tokenization, we chose not to use MADA/TOKAN, because we found that the implementation did not scale to our dataset (millions of microblog messages). Instead, we implemented the simpler approach by Lee, Papineni, Roukos, Emam, and Hassan (2003) for tokenization. This approach only requires a small set of annotated data to obtain a list of prefixes and suffixes and uses n-gram models to determine the most likely *prefix***-stem-suffix** split of a word. The n-gram model required for tokenization was trained on 5.8 million Modern Standard Arabic sentences from the NIST evaluation campaign. This data had previously been tokenized with the same method, trained to match the Penn Arabic Treebank, v3 (Maamouri, Bies, Buckwalter, & Mekki, 2004). All English sentences were tokenized and lowercased using scripts from the Moses toolkit. We removed all Twitter-specific text markup from Twitter data for translation, since it was not included in our evaluation set (see Section 3.3).

Adaptated baselines. We compare our approach with a series of known adaptation methods that only use monolingual in-domain data and/or a small tuning set. A similar set of

[12] nist.gov/itl/iad/mig/open-machine-translation-evaluation
[13] http://statmt.org/moses/
[14] speech.sri.com/projects/srilm/
[15] qamus.org/transliteration.htm

experiments is conducted in Bertoldi and Federico (2009) on Spanish-English adaptation in the domain of parliamentary proceedings. We conduct experiments using three adaptation methods: First, the crowd-sourced in-domain development set is used for optimizing the weights of the feature weights as proposed by Koehn and Schroeder (2007). Second, the English tweets in our crawl are used to build a 5-gram in-domain language model. This adaptation technique was first proposed by Zhao, Eck, and Vogel (2004). Third, the Arabic portion of our crawl is used to synthetically generate additional parallel training data. This is accomplished by machine-translating the Arabic tweets with the best system after performing the first two adaptation steps, as proposed by Ueffing, Haffari, and Sarkar (2007). Due to time constraints, only 1 million randomly selected Arabic tweets are used to generate synthetic parallel data. We also combine all three approaches described above. All experiments are compared to a baseline of using no in-domain data for translation model, language model and weight optimization.

Our Approach. Using the best adapted model, we then integrate the phrase tables created by our method. Moses offers different mechanisms for phrase-table combination. We experiment both with a strategy which loads translation options from *both* phrase tables, along with the *back-off* mechanism, where the in-domain phrase table is only used if no translation is found in the original phrase table.

3.4.2 Experimental Results

For all experiments, we report lowercased BLEU-4 scores (Papineni et al., 2002) as calculated by Moses' `multi-bleu` script. For assessing significance, we apply an approximate randomization test (Noreen, 1989; Riezler & Maxwell, 2005), computed with the MULTEVAL toolkit (Clark, Dyer, Lavie, & Smith, 2011).

Table 3.4 shows BLEU scores for the baseline, as well as language model, tuning set and self-training adaptation. All adaptation experiments were significantly better than the unadapted baseline. Using the in-domain development set while leaving everything else untouched led to an improvement of about 1 BLEU point over the baseline (Experiment 2). Experiments 3 and 4 use a Twitter language model. Experiment 3 improved over the baseline by 2 BLEU points, while Experiment 4, which used only in-domain language model and development data, improved by 1.8 points. There was no significant difference between Experiments 3 and 4. These results confirm Bertoldi and Federico (2009)'s findings of the language model being the most helpful component for domain adaptation. Experiment 5 used both in-domain and out-of-domain language models. The language models were treated as separate features in the log-linear model (called "log-linear in-

	translation model	language model	dev set	% BLEU
1	NIST	NIST	NIST	13.90
2	NIST	NIST	Twitter	**14.83**
3	NIST	Twitter	NIST	**15.98**
4	NIST	Twitter	Twitter	**15.68**
5	NIST	Twitter & NIST	Twitter	**16.04**
6	self-train	Twitter & NIST	Twitter	**15.79**
7	self-train & NIST	Twitter & NIST	Twitter	**15.94**

Table 3.4 Domain adaptation experiments. Boldface indicates a significant improvements over baseline (1).

terpolation" by Foster and Kuhn (2007)). There was no significant improvement over Experiments 3 and 4. In Experiments 7 and 8 a translation model was either trained completely on synthetically generated parallel in-domain data (7) or on the combination of self-translated and out-of-domain data (8). We observed no improvement by adding self-translated data to our translation model. As has been observed before by Bertoldi and Federico (2009), it did not matter whether the synthetic data were used on their own or in addition to the original training data. We use the configuration from Experiment 5 in all further experiments, since it yielded the highest absolute BLEU score.

Table 3.5 on page 42 shows the results for our approach, using different consolidation and extraction settings. For the experiments using `conservative` extraction (Experiments 8a-11), differences are small across different settings. Results 8a and 9a clearly perform worse other experiments with the same consolidation setup, showing that requiring five or more bidirectional alignment points for candidate pairs to even be considered is too restrictive. Due to time constraints and small differences in results not all phrase extraction settings were tried out with all candidate consolidation approaches. The `unidirectional`, $m = 3$ setting can be compared across all consolidation approaches (Experiments 8c, 9c, 10a and 11). For this setting, less restrictive consolidation (`top10`) seems to work better, although the differences between experiments are small. The best result for conservative phrase extraction (Experiment 9b) uses `top3` consolidation and filters all candidates with less than 3 `bidirectional` alignment points. This setting produces an improvement of 1 BLEU point over Experiment 5.

Experiments 12, 13 and 14 use the `bold` phrase table extraction strategy. Compared to the conservative strategy, the bold strategy produced a much larger improvement of almost 3 BLEU points over the adapted baseline and 5 BLEU points over the unadapted system.

	consolidation	extraction	phrase table combination	%BLEU
8a	`top10`	`bidirectional`, $m = 5$	both	16.38
8b	`top10`	`bidirectional`, $m = 3$	both	16.52
8c	`top10`	`unidirectional`, $m = 3$	both	16.90
9a	`top3`	`bidirectional`, $m = 5$	both	16.59
9b	`top3`	`bidirectional`, $m = 3$	both	**17.04**
9c	`top3`	`unidirectional`, $m = 3$	both	16.86
9d	`top3`	-	both	16.96
10a	`top1`	`unidirectional`, $m = 3$	both	16.79
10b	`top1`	-	both	16.79
11	`top1+intersect`	`unidirectional`, $m = 3$	both	16.53
12	`top1`	`bold`	both	**18.73***
13	`top1+intersect`	`bold`	both	**18.85***
14	`top1+intersect`	`bold`	back-off	**18.85***

Table 3.5 Variants of our approach using pseudo-parallel in-domain data. All runs use the same configuration as Experiment 5 in Table 3.4 for development set and language models. Boldface indicates a significant improvement over row 5 in Table 3.4 and * indicates a significant over row 9b, each at $p \leq 0.01$.

Results with the bold strategy are very similar to each other, but the best overall result was achieved in experiment 14 by using `top1+intersect` consolidation and adding the in-domain phrase table in back-off mode. This result suggests that the conservative strategy is unable to capture much of the information present in the automatically extracted nearly parallel tweets. It also suggests that the SMT model is able to handle a considerable amount of noise in the phrase table without any further restrictions.

3.5 Analysis

This section contains a more in-depth quantitative and qualitative analysis to outline the strengths and weaknesses of our proposed approach, as well as the remaining challenges.

Out-of-vocabulary words. As mentioned above, the Twitter translation task suffers from high numbers of out-of-vocabulary (OOV) items due to stylistic and orthographic variation. We therefore analyze to what extent the adapted models were able to solve this problem. We measure the effect of OOV words by counting the number of tokens in Arabic script that appear in the translated output. These tokens are copied and pasted into the output string when the decoder is unable to find a good translation. The Arabic token count is then divided by the total number of output tokens to compute the Arabic (AR) token rate. Table 3.6 reports the Arabic token rate for Experiments 1 (baseline with no adaptation), 5 (language model and development set adaptation), 9b (conservative translation model adaptation on top of 5), and 14 (bold translation model adaptation on top of 5). We can see that the adapted baseline (5) and the conservative in-domain phrase table (9b) have lower AR token rates than the unadapted baseline, with (9b) again having a lower AR token rate than (5). However, the rate remains high. The model using the bold in-domain phrase table (14), on the other hand, only has an AR token rate of 4.22%. This result shows that there is a large overlap in Twitter- or topic-specific vocabulary between the test data and the automatically extracted pseudo-parallel data, which the system is unable to learn from the out-of-domain training data.

Experiment	AR token rate
1	22.56
5	20.05
9b	17.47
14	4.22

Table 3.6 Arabic token rate for different adaptation methods.

Dialectal content. One difficulty posed specifically by Arabic-English translation of informal content is the large dialectal variation within Arabic. As most of our out-of-domain data set comes from official sources, it is mainly in Modern Standard Arabic. To gain an impression of the amount of dialectal content in our data, we used the Arabic Online Commentary Dataset created by Zaidan and Callison-Burch (2011a) to train language models for the purpose of classifying our test data. Table 3.7 shows the distribution of dialects in our test data according to language model probability of each tweet. This distribution should be viewed with a grain of salt, since the shortness of tweets might cause unreliable results when using a model based on word frequencies for classification. Still, the results points to a high proportion of dialectal content and spelling variation in our data. One frequent example of this is the preposition في , meaning "*in*", which is often written as فى . The out-of-domain translation model, as well as the conservatively extracted in-domain

phrase table failed to translate this frequently occurring word. Only when applying the bold phrase table generation strategy, did the system translate it correctly.

Dialect	# Sentences
Egyptian	141
Levantine	147
Gulf	78
Modern Standard Arabic	145

Table 3.7 Dialectal content in our test set as classified by the AOC dataset.

Qualitative Analysis of translation examples. Pages 45 and 46 show examples of translations generated using the different adaptation methods to illustrate strengths and weaknesses of our approach. The source is given along with the crowd-sourced reference translations. We then compare translations generated by Experiments 1, 5, 9b and 14. We also include a translation generated by Google's online translation service[16].

Example 1 contains a case where unknown words were learned through translation model adaptation. Models 9b and 14 correctly translated the word مسيلات as "*tear gas*", where the baselines (1 and 5) passed through an unknown term. Even the Google translator transliterated the word and produced the output "*Msellat*". The unadapted baseline erroneously translated the ambiguous place name "*sitra*" as "*jacket*". The same mistake was made by the Google translator and by two of the human translators. Only Experiment 9b produced the correct translation. A similar problem occurred when translating the ambiguous place name "*wadyan*", which could also be taken as meaning "*and religions*". This error was probably introduced by our pre-processing, which incorrectly split off the prefix "*w*" which often carries the meaning "*and*".

In Example 2, our best system (14) clearly outperformed all other systems, as it was the only one of our systems to translate the proper names "*obama*" and "*al awlaki*" at all. This shows the success of our method at adapting to specific terminology of the target domain.

[16] `translate.google.com`. Note that the translations were obtained in early 2013 and do not reflect the current state of the service.

Example 1

Source	سترة قوات الشغب تقتحم واديان مترجلة وتطلق مسيلات الدموع
Ref. 1	vest riot forces break into wadyan by foot and trough gas tear
Ref. 2	sotra the riot forces enter on foot and shoot tear bombs
Ref. 3	the cover for riot police enters wadian walking and shoot tear bombs

Baseline (1)	jacket riot forces storm and religions foot مسيلات وتطلق tears
(5)	sitra and religions of the foot of the riot forces storm مسيلات وتطلق tears
(9b)	in sitra riot police storming and religions of tear gas on foot
(14)	the riot police stormed and religions of the foot firing tear gas
Online	Riot troops stormed the jacket and religions foot and launches Msellat tears

Example 2

Source	أوباما سيتحدث اليوم عن مقتل العولقى
Ref. 1	obama will talk today about the killing of al - awlaki
Ref. 2	obama is talking today about el awlaqi death
Ref. 3	obama will speak today about the killing of al - awlaqi

Baseline (1)	أوباما سيتحدث today killed العولقى
(5)	أوباما سيتحدث friday for the killing of العولقى
(9b)	أوباما today on the killing of
(14)	obama today on the al awlaki killing
Online	Obama will speak today the death of al-Awlaki

Example 3

Source	الشبيحة في حماة يستغيثون :)
Ref. 1	the gangsters in hama are asking for help
Ref. 2	the gangs in hamah are peading :)
Ref. 3	the thugs in hama are calling for help :)
Baseline (1)	الشبيحة mired in calling for help :)
(5)	الشبيحة in hama calling for help :)
(9b)	inside the protectors of the calling for help :)
(14)	shabiha in hama calling for help :)
Online	Cbihh in Hama are crying :)

Example 4

Source	حريه :: عاملون بالمصرية للاتصالات يحتجزون رئيس الشركة فى غرفة بسنترال الأوبرا
Ref. 1	freedom :: workers in the egyptian for communication are holding the company president in a room in the opera central
Ref. 2	freedom , workers in egypt for calls detain the head of the company in a room in opera central
Ref. 3	hurriya :: workers in telecom egypt detaining the president of the company in a room in the opera central
Baseline (1)	: : free workers للاتصالات بالمصرية holding company chairman بسنترال فى الأوبرا chamber
(5)	: : workers free للاتصالات بالمصرية holding company chairman بسنترال الأوبرافى room
(9b)	free : : afcd بالمصرية hold ceo hostage ppl is the president of the chamber of الأوبرا بسنترال
(14)	egypt : : workers telecom workers are holding the head of the company in the chamber of really opera
Online	Freedom :: Telecom Egypt workers holding company's president in a room Psontral Opera

While the first two examples resemble news text, EXAMPLE 3 is a more informal message. It is particularly interesting to note that with our best system (14) the term "*shabiha*" is learned, which was used in Syria to mean "*thugs*" and specifically refers to armed civilians who assaulted protesters against Bashir Al-Assad's regime (see *Syria unrest: Who are the shabiha?*, 2012).

EXAMPLE 4 also shows substantial unknown word reduction by our best system. However, the term الأوبرا بسنترال ("*in Opera Central*", the location of Telecom Egypt) is incorrectly translated as "*really opera*".

Conclusions

In this chapter, we have formulated the hypothesis that document meta-information contained in Twitter messages could be leveraged to extract pseudo-parallel in-domain data to build a topically constrained translation system. Such a system could be helpful for, e.g., aid workers, the press or people involved in events of global significance. To test our hypothesis, we developed a tailored approach for mining pseudo-parallel data from a large crawl of Twitter messages. Document meta-information was used to crawl relevant messages, as well as to pair tweets cross-lingually. We also leveraged the shortness of tweets, which allowed us to directly pair retrieved candidate tweets with query tweets without having to perform sentence alignment. We then tested different ways of using those messages to improve an out-of-domain SMT system. Besides several ways of filtering candidate pairs returned by the retrieval step, we also compared conservative and bold strategies for extracting an in-domain phrase table. To evaluate our method we used an in-domain evaluation data set of relevant Arabic tweets, for which we solicited English translations using crowdsourcing. Analysis of the crowdsourced translations shows that with our setup it was possible to obtain an evaluation set in a reasonable amount of time. Workers who cheated the system could be easily filtered by the task completion time. Redundancy was found to be crucial, as the translations are not of professional quality, as expected. In the experiments, various configurations of our method were compared experimentally. We also tested several baselines using only monolingual Twitter data and the Twitter development set for adaptation.

The results showed that

- standard adaptation methods produced an improvement of up to 2 BLEU points over the out-of-domain baseline.

- the conservative phrase table extraction strategy improved the system by up to 1 BLEU point over the adapted baseline.

- the bold phrase table extraction strategy improved the system by almost 3 BLEU points over the adapted baseline.

A quantitative and qualitative analysis of the translations showed that translation of proper names and Arabic dialectal content still remains a challenge. However, it also showed that a substantial reduction of out-of-vocabulary terms can be achieved by our method.

Applicability to Neural Machine Translation (NMT) While our experiments have been conducted using an SMT system, the approach can be adapted to the neural paradigm. Our final system combines phrase tables from general-domain data with an in-domain phrase table. While this approach cannot be directly translated to NMT, there are several approaches to integrating in-domain data into an NMT model. First, continued training (Freitag & Al-Onaizan, 2016; M.-T. Luong & Manning, 2015) utilizes smaller amounts of in-domain data by first training a model on a large general domain data set and then continuing to train on the smaller in-domain data to bias the model towards the target domain. Second, if there is sufficient data, separate in-domain and out-of-domain models can be trained and combined by ensemble decoding as done in Sutskever et al. (2014). Finally, back-translation of monolingual in-domain data has shown much more promise for NMT than self-translation approaches have for SMT (Sennrich, Haddow, & Birch, 2016b). The CLIR model also utilizes a word translation table, created by the IBM models for word alignment algorithm associated with SMT. Recent work on neural information retrieval has suggested using word embeddings to model word translation probabilities for monolingual information retrieval (Zuccon, Koopman, Bruza, & Azzopardi, 2015). T. Luong, Pham, and Manning (2015) inter alia present ways to learn bilingual word embeddings ,which would allow to transfer this approach to the bilingual case.

CHAPTER 4

Simulating Task Loss Using Citation Information for Query Translation

We now turn to a scenario where full supervision for translation is available. This means that we have a large parallel data set from the desired domain. But what if the downstream application we are interested in is not translation, but another multilingual task? In this case, the translations may need to fulfill requirements that cannot be captured by optimizing a translation quality metric on references. The task studied in this chapter is query translation for cross-lingual patent prior art search as a downstream application. In this scenario, search queries have to be constructed from full-text patent applications. But even though full supervision is provided for end-to-end translation, no direct supervision is available for query translation. Unlike end-to-end translation systems, a query translation system is judged by its ability to generate queries which return relevant documents when passed to a retrieval system, not by their fluency or adequacy. We therefore propose an adapted training objective and algorithm to directly optimize a query translation system for retrieval performance. Given the absence of large-scale training data for this task, we use document information in the shape of cross-lingual document-level links to provide a training signal. More specifically, we simulate the task of patent prior art search using citation information in a large multilingual patent collection.

Before a patent application can be granted, the office to which the patent has been submitted, has to confirm the invention's novelty by checking that it is not part of the state

of the art.[1] Any existing patents invalidating an application's novelty are called *prior art*. Identifying such prior art is the main objective of patent prior art search. Patent prior art search is usually conducted by a trained human patent examiner. Given a new patent application, the examiner's task is to identify any previously published patents which invalidate the proposed patent's novelty. While automating prior art search may not eliminate the need for human review, it has the potential to speed up the search process. In particular, automatic query translation could replace a costly manual process of constructing a search query and translating query terms into the desired target language. In combination with a strong end-to-end translation system, the entire process of cross-lingual patent prior art search could be conducted in a single language. Automated cross-lingual prior art search also helps inventors, who could use such a system to check for existing prior art before filing a patent application. Prior art search differs from other retrieval tasks in two ways: First, the search query consists of a full-text document. These queries make it necessary to use a full SMT system rather than a simple dictionary lookup of terms, as might be sufficient for, e.g., web search queries. Second, patents, as well as their publication history, are publicly available, allowing us to build a large data set for training purposes.

In this work we focus on optimizing query translation for prior art search. We compare untuned translation systems to translation systems optimized for translation quality and for retrieval quality. Our main contributions are: (1) An objective and training algorithm for directly optimizing patent query translation for retrieval performance, as well as an analysis of the hyperparameters and a proof-of-concept evaluation of the approach. (2) The construction of large-scale training and evaluation data for Japanese-English cross-lingual patent retrieval. (3) A competitive Japanese-English patent translation system. While contribution (1) is entirely the author's own work, contributions (2) and (3) were made in collaboration with other authors and published in Sokolov et al. (2013) (contribution 2) and Simianer et al. (2013) (contribution 3). After reviewing related work in Section 4.1, we describe the construction of a large-scale data set for Japanese-English cross-lingual patent retrieval in Section 4.2. Section 4.3 gives details on the Japanese-English patent translation system. Section 4.4 presents a task-specific learning objective and a framework for optimizing the translation model for this objective. Lastly, Section 4.5 describes experiments and discusses results. This section also contains some analyses which motivate design and feature choices.

[1] See for example Article 54 of the European Patent Convention (`http://www.epo.org/law-practice/legal-texts/html/epc/2016/e/ar54.html`).

4.1 Related Work

There is a substantial amount of prior work on automated patent retrieval and patent translation, including several benchmark tasks. From 2009 to 2013, the CLEF Initiative[2] organized an Intellectual Property track (CLEF-IP). This track featured various patent related tasks, including patent prior art search on a multilingual document collection from the European Patent Office (EPO). The test collection created for this task is similar to the data set which is presented in this work. Differences with respect to our data set and task setup will be discussed in Section 4.2. Benchmarks for patent translation and patent retrieval were also held at the NTCIR workshops[3] from 2002 to 2013. The patent translation system presented here was submitted to the NTCIR-10 patent translation workshop in 2013, where it performed comparatively well. Section 4.3 contains our system description.

Most multilingual patent retrieval approaches use translation as a black box. Magdy and Jones (2011) suggest to learn a translation system on pre-filtered data. Their experiments mainly aim to speed up the translation process, while ours target the question how to optimize the translation system for patent retrieval. The work most similar to ours is Nikoulina, Kovachev, Lagos, and Monz (2012). They use data from the CLEF AdHoc-main, AdHocTEL and GeoCLEF tracks for training and evaluate on the CLEF AdHocTEL task. Like us, they use a ranking algorithm. However, their work re-ranks k-best translation outputs, leaving the translation system untouched. Our work goes further insofar as we specifically try to re-train the translation system to produce good search queries. Their approach offers the advantage that they can extract additional syntactic features from the completed hypotheses. However, their results fail to identify robust features across language pairs, and are often very similar for different feature sets, making it hard to determine whether syntax features actually help in query translation.

Since our work focuses on translation, we consider a scenario where we are given a monolingual retrieval system. We aim to produce query translations that will optimize this system's performance. Other recent work has sought a tighter integration of retrieval and translation in a probabilistic model (Ture, Lin, & Oard, 2012) or by using the SMT decoder directly for document ranking (Hieber & Riezler, 2015). While these approaches have shown promising improvements in retrieval quality, they also face issues of scalability. In contrast, our approach requires additional training time, but does not slow down test time performance compared to a *direct translation* baseline which produces target language queries by running a task-agnostic translation system.

[2] `http://www.clef-initiative.eu/web/clef-initiative/home`
[3] `http://research.nii.ac.jp/ntcir/index-en.html`

4.2 Creating a Crosslingual Patent Retrieval Corpus

The creation of this data set was carried out collaboratively with the author taking main responsibility, but other co-authors also contributing. We will give pointers to related publications where applicable.

As mentioned in the introduction, we focus on the task of simulated patent prior art search. Graf and Azzopardi (2008) describe a method to build an annotated test collection for patent prior art search. They suggest to exploit the patent citation graph to extract relevance judgments for patent retrieval from patent citations. The key idea of their proposal is to regard patent documents that are cited in a query patent, either by the patent applicant, or by the patent examiner in a patent office's search report, as relevant for the query patent. We adopt this approach. This means that our setup does not directly mimic prior art search, as our system is tasked with producing all patents cited by an application document, and rank them in the correct order - with examiner citations being assigned higher relevance than applicant citations. This method has also been applied to generate a test collection from European patent documents for CLEF-IP shared tasks for 2009 and 2010 (Roda, Tait, Piroi, and Zenz (2010), Piroi and Tait (2010)). However, these evaluations do not focus on the multilingual/translation aspect but more on the retrieval, as they allow versions of documents in all three official European patent languages. Since this study is focused on the translation system, only the source language portions of a query and only the target language portions of a document are extracted. This setup does not fully match the reality of patent prior art search, as many documents are partially translated, but provides a cleaner test bed for evaluating only query translation.

This study uses Japanese-English patent translation. The data set is based on Japanese patent applications from the NTCIR evaluation. Cross-lingual citation information is extracted from the citation graph provided with the MAREC[4] data set. This data set includes citation information for patents published in Japan and in the United States. Since the MAREC corpus only contains English abstracts for Japanese patents, but not the Japanese full texts, the patent documents in the NTCIR-10 test collection described above are merged with the Japanese (JP) section of MAREC, allowing them to conform to MAREC's XML format. Following previous work, we use the cross-lingual applicant and examiner citations of patent documents published by the United States Patent and Trademark Office (USPTO) for each Japanese patent application to define relevance levels.As applicant citations have been found to be less relevant than examiner citations (Criscuolo & Verspagen, 2008), they are assigned a lower relevance level (level 1) than patents cited by an examiner (level 2). Additionally, family patents are also added to the set of relevant

[4] `http://www.ifs.tuwien.ac.at/imp/marec.shtml`

	#queries (JP)	#relevance links (JP-EN)	#unique docs (EN)
train	107,061	1,422,253	888,127
dev	2,000	26,478	25,669
test	2,000	25,173	24,668

Table 4.1 Statistics of the query and document collections of BoostCLIR. Queries are extracted from Japanese patent applications. The second column lists the total number of relevance links between Japanese and English patents extracted from the citation graph. The third column lists the number of unique English relevant documents.

documents for training. Two patents are in the same family if they are granted by different authorities but related to the same invention. As family patents are often translations of the original document or at least closely comparable, we assign to them the highest relevance level (level 3). The extraction and validation of the patent citation relations was in part carried out in Ruppert (2013).

Information to reproduce our final data set has been published at `http://www.cl.uni-heidelberg.de/statnlpgroup/boostclir`. Training, development and test sets used by the experiments reported in Sokolov et al. (2013) are provided in this data set. The training set is restricted to data from the years 2003-2005, as this leaves data from previous years for training the patent translation system. The document collection for training contains all USPTO patents from 2003-2005 that were included in the MAREC collection. Development and test sets are sampled from applications submitted in 2006-2007. They each contain 2,000 queries. The document collections for development and test set have been obtained by randomly adding irrelevant documents from the NTCIR-10 collection until two pools of 100,000 documents are obtained. Table 4.1 contains summarizing statistics of the collection.

4.3 A Japanese-English Patent Translation System

Patent translation is a challenging task in itself. The jargon, technical vocabulary, and highly idiomatic sentence structure require a system trained on genre specific data and an elaborate pre-processing pipeline. At the same time, patents can be easier to translate than other genres, because their language is highly regulated and very repetitive, and does not contain named entities. Along with the large volume of parallel or highly comparable multi-lingual patent documents, these facts make SMT very well-suited for patent translation. Our system further exploits these characteristics by introducing sparse lexicalized

features. This section describes the HDU system submitted to the NCTIR-10 shared task on patent translation. It is based on the system description published as (Simianer et al., 2013). Work by other authors of this paper will be presented summarily and indicated as such.

4.3.1 System Setup

When translating from Japanese into English, the typical long sentences of patent text present an even more pressing problem than they would in languages with similar word order. We employ a hierarchical phrase-based translation model (Chiang, 2005) to better handle reordering problems. We use the implementation of the model in the CDEC toolkit (Dyer et al., 2010). During training, symmetrized word alignments are generated using GIZA++ (Och & Ney, 2003) on lowercased data in both directions and applying the `grow-diag-final-and` heuristic (Koehn et al., 2003). The SCFG-grammars required for CDEC are induced from the word alignments using the method described by Lopez (2007). We include a 5-gram language model trained with the SRILM toolkit (Stolcke, 2002). Given these settings, we train a baseline using 12 dense decoder features, which are tuned using minimum-error rate training (MERT) (Och, 2003). The final benchmark system includes two crucial modifications of this baseline: First, an extended, tailored pre-processing pipeline and second, a multi-task tuning method, which includes lexicalized sparse features. Both modifications and their impact are described below.

4.3.2 Preprocessing

The specialized vocabulary of patents, along with the particular challenges posed by Japanese-English translation requires task-specific pre-processing. In order to deal with this problem, our system draws from several pre-existing approaches to put together a tailored pre-processing pipeline.

The MeCab[5] toolkit is used for segmentation of Japanese text. We apply several post-processing steps to the segmented output to alleviate over-segmentation of ASCII-strings and under-segmentation of katakana terms. Due to their technical nature, the patent texts contain a large number of non-Japanese expressions, such as abbreviations, patent identifiers or English terms. Since the non-Japanese-characters in the provided data are in Fullwidth Latin, MeCab tends to heavily over-segment them at each character position, leading to faulty alignments. Therefore, we convert all fullwidth Latin characters

[5] `https://code.google.com/p/mecab/`

Baseline	In addition, リヤバンパービーム 111 is inclined in the direction of the arrow, and load is applied to
`+katakana-split`	In addition, the rear bumper beam 111 is inclined in the direction of the arrow and load applied to the rear bumper beam 111 escapes.
reference	In addition, if the rear bumper beam 111 inclines in the arrow direction, the load acting on the rear bumper beam 111 will escape.

Table 4.2 Example illustrating the effect of splitting katakana transliterations of English compounds.

to ASCII format before running the segmenter. Even with this format, tokenization of ASCII strings occurring in English and Japanese sentences is sometimes inconsistent. To avoid this problem for training, we follow Ma and Matsoukas (2011)'s approach for Chinese segmentation and tried to enforce one consistent tokenization to ASCII-strings in the Japanese training data and to their English counterparts. For the parallel training data, we use regular expressions to align ASCII-strings between Japanese and English and then replace strings in Japanese with their English counterparts. For the Japanese test data, we always use the tokenization which has been seen most often in the training data.

We also follow Feng et al. (2011), who apply a modified version of the compound splitter described in Koehn and Knight (2003) to katakana strings. Katakana characters are used in Japanese for foreign loan words, many of which are transliterations of English compound words. These katakana strings are not split by MeCab, causing additional out-of-vocabulary (OOV) words or faulty alignments. Applying compound splitting to katakana strings reduces the number of OOV terms in the devtest set from 98 to 34. Table 4.2 shows an example translation where katakana splitting is crucial to produce the correct translation.

In addition to the above steps, a modified version of the Moses tokenizer[6] is applied to the English side of the training data. The tokenizer's list of prefixes where no whitespace is inserted before a punctuation mark is extended to include patent-specific prefixes such as "*Chem*", "*FIG*" or "*Pat*". This pre-processing step has been implemented by one of the co-authors, while the other pre-processing steps have been implemented by the author of this thesis.

The combined pre-processing pipeline produces an improvement of 1 BLEU point over a baseline using the segmenter and tokenizer without modifications. After decoding, the

[6] `github.com/moses-smt/mosesdecoder`

system	% BLEU
RWTH-2	33.08
HDU-2	32.07
HDU-1	31.92
BASELINE-2	28.86
BASELINE-1	28.56
ONLINE-1	24.24

Table 4.3 Experimental results on the Intrinsic Evaluation set at the NTCIR10 Japanese-English patent translation subtask. HDU-1 and HDU-2 are identical, except for HDU-2 stopping early in the tuning phase. HDU-2 ranks second among systems which use a constrained data setup.

system output is detokenized and recased using the Moses tools, with the recaser trained on the truecased English side of the training data.

4.3.3 Tuning and Shared Task Results

For the final system we use the pairwise ranking optimizer DTRAIN (Simianer et al., 2012), which, unlike MERT, is able to handle the tuning of hundreds of thousands of sparse lexicalized features. DTRAIN also includes a framework for multi-task training with feature selection, thus allowing to tune on larger data sets. Multi-task tuning with sparse features produces a gain of 1.16 BLEU points over the baseline described above, which uses 12 dense features and MERT-tuning. For details on the tuning experiments, which were carried out by another co-author, refer to Section 4.3 of Simianer et al. (2013). Our benchmark system uses multi-task tuning with sparse feature selection on 5 development sets of approximately 1,000 sentences. Table 4.3 shows the official benchmark results. By combining a hierarchical phrase-based model with a tailored pre-processing pipeline and sparse features selected with multi-task tuning, our systems (HDU) are able to perform 2nd and 3rd best on the patent translation task, substantially outperforming the baselines supplied by the task organizers. For full listings of results for all subtasks and evaluations see Goto, Chow, Lu, Sumita, and Tsou (2013).

4.4 A Pairwise Ranking Framework for Retrieval-Optimized Translation

Given a strong translation system and a large data set for cross-lingual patent retrieval, we now propose a method to bridge the gap between end-to-end translation and monolingual retrieval by specifically optimizing query translation for retrieval performance. We present ROPE, a retrieval-optimized pairwise ranking perceptron algorithm for patent query translation.

Loss function We use the pairwise ranking perceptron for retrieval-based optimization. This is well-suited to this task as it is linear in the number of training examples and able to handle large amounts of sparse features. It is easy to implement and does not require a gold reference. We assume a log-linear SMT model with m feature functions $\phi(x, h, y)$ weighted by weights $\mathbf{w}$. The features are defined over in input $x = x_1, \ldots, x_i$ in the source language, a latent derivation h and the translation $y = y_1, \ldots, y_j$ in the target language yielded by h. We further require an external metric $\delta(y)$ which assigns a non-negative real-valued score to translations. The pairwise ranking perceptron optimizes the following structured hinge loss:

$$L_{perc\_rank} = \sum_{(y, y') \in \mathcal{P}(x)} \max\left(0, -\mathbf{w}^\top \Delta\phi(x, y, h, y', h')\right) , \tag{4.1}$$

with

$$\Delta\phi(x, y, h, y', h') = \phi(x, y, h) - \phi(x, y', h') .$$

y and y' are translations from the output space of x, $\mathcal{Y}(x)$, with their corresponding derivations h and h'. Let $\mathcal{P}(x)$ be a set of translation pairs such that y is preferred over y' according to the metric δ:

$$\mathcal{P}(x) = \{y, y' \in \mathcal{Y}(x) \mid \delta(y) > \delta(y')\}$$

This loss can be optimized using stochastic subgradient descent with the gradient being defined as

$$\nabla_{\mathbf{w}} L_{perc\_rank} = \begin{cases} -\Delta\phi(x, y, h, y', h') & \text{if } \Delta\phi(x, y, h, y', h') < 0 \\ 0 & \text{otherwise} . \end{cases}$$

The output space $\mathcal{Y}(x)$ is often too large to be efficiently enumerated, but can be approximated by a k-best list of translations which we designate by $\mathcal{K}(x)$. Further, heuristic methods have been used to restrict the set $\mathcal{P}(x)$ to the most informative pairs (see Hopkins

& May, 2011; L. Shen & Joshi, 2005; Simianer et al., 2012). The pairwise ranking perceptron has been successfully applied for the optimization of end-to-end translation models by Hopkins and May (2011) and Simianer et al. (2012). For end-to-end translation, the metric $\delta(y)$ is computed with respect to a reference translation y^*, using a sentence-level approximation of the BLEU score (Nakov et al., 2012).

Retrieval metric For the task of query translation there are no reference translations. Instead, we are given a set of multi-sentence queries q in the source language and documents D in the target language. For each query, a set of relevant documents $D^+(q) \subset D$ is provided by cross-lingual document-level links. We assume a monolingual retrieval model $\textsc{retrieve}(q, D)$ which, given a query, outputs a list of the top N highest-ranking documents $d \in D$ with respect to the query. We then define the metric $\delta(y)$ for query translation as

$$\delta(\tilde{q}) = \textsc{score}(\textsc{retrieve}(\tilde{q}, D), D^+(q)) \,,$$

where $\tilde{q}$ is a translated query containing y. For the $\textsc{score}$ operator, we use Normalized Discounted Cumulative Gain (NDCG) (Järvelin & Kekäläinen, 2002) For a query q_j, considering the top N returned results, NDCG is defined as

$$\mathrm{NDCG}(q_j) = Z_{Nj} \sum_{m=1}^{N} \frac{2^{R(j,m)} - 1}{\log_2(1+m)} \,,$$

where $R(j, m)$ refers to the relevance score assigned to the document returned at rank m with respect to query j, and Z_{Nj} is a normalizing factor, which ensures that a perfect ranking up to rank N will receive an NDCG score of 1. We use NDCG, as it is specifically designed to handle non-binary relevance.

Handling multi-sentence queries. A learning algorithm for the envisaged setup must address the issue that the translation system expects to process one sentence at a time, while the performance metric is defined with respect to a query. A simple solution would be to treat each translation as a search query in isolation and to ignore the given query context. But including the query context for each translation y would be a better match for the task of query translation. The query context is included by treating each multi-sentence query as a *minibatch*. Updates are accumulated over all training instances (pairs) generated from one input query before updating the weight vector. Our algorithm first generates k-best translations for each sentence in the query q, resulting in at most $k \cdot |q|$ translations. The query context $\hat{q}$ is then generated by concatenating the Viterbi translation $\hat{y}$ for each sentence in q. Search queries $\tilde{q}$ are constructed by replacing a Viterbi translation in $\hat{q}$ with a translation $y \neq \hat{y}$ from the corresponding k-best list. Retrieval is

Algorithm 2 ROPE: **R**etrieval-**o**ptimized pairwise ranking **pe**rceptron

Require:
Number of epochs T,
Set of target language documents D,
Set of source language queries Q with relevance annotations $D^+(q)$ for each $q \in Q$,
k-best list size k, learning rate η, document sample size s
$\mathbf{w} \leftarrow \mathbf{0}$
for T epochs **do**
  for each query $q \in Q$ **do**
    $\hat{q} \leftarrow \emptyset$
    for each sentence $x \in q$ **do**
      $\mathcal{K}(x) \leftarrow \text{DECODE}(x, k, \mathbf{w})$ ▷ Generate k-best list
      $\hat{q}$ += $\hat{y}$ ▷ Add top-1 translation to query context
    end for

    $\mathcal{P}(q) \leftarrow \emptyset$ ▷ Initialize preference pairs.
    $D' \leftarrow \text{SAMPLE}(s, D)$ ▷ Sample documents.
    for each sentence $x \in q$ **do**
      $\text{Ret}(x) \leftarrow \emptyset$ ▷ Initialize retrieval scores
      for each translation $y \in \mathcal{K}(x)$ **do** in parallel
        $\tilde{q} \leftarrow \text{REPLACE}(\hat{y}, y)$ ▷ Insert translation into query
        $\text{Ret}(x)$ += $\text{SCORE}(\text{RETRIEVE}(\tilde{q}, D'), D^+(q))$ ▷ Retrieval and scoring
      end for
      $\mathcal{P}(q)$ += $\text{GENPAIRS}(\mathcal{K}(x), \text{Ret}(x))$ ▷ Build preference pairs
    end for

    for $(y, y') \in \mathcal{P}(q)$ **do**
      $\mathbf{w}$ += $\eta \Delta\phi(y, y')$ ▷ Update weights.
    end for

  end for
end for

then performed for each candidate query, and the k-best lists are sorted according to retrieval scores. For each input sentence $x \in q$, a set of preference pairs is then created from the corresponding k-best list and added to the set of pairs for the current query, $\mathcal{P}(q)$. Finally, the weight vector is updated for each pair. Algorithm 2 contains pseudocode for our retrieval-optimized pairwise ranking perceptron (ROPE).

Implementation Notes. Using a retrieval-based metric requires running retrieval for $k \cdot |q|$ query translation candidates per input query. As we do not update the weights while processing one query, we can speed up this step by scoring hypotheses in parallel. We also do not use the full target document collection D for scoring candidate queries, but only use a sample D' with $|D'| \ll |D|$ of irrelevant documents. The sample size s is

a hyperparameter to be set by the user. We base our implementation on the code by Simianer et al. (2012).[7]

4.5 Experiments

4.5.1 SMT and Retrieval Baselines

The translation system described in Section 4.3 is used in our experiments with minor modifications. Our experimental system is trained only on the NTCIR-7 training data (1.8 million sentence pairs). This data has been extracted from Japanese and English patents published prior to 2003. We set aside patents published from 2003 to 2005 for retrieval-based optimization. Using this setup, we create three baseline SMT systems: The first system, *manual*, uses manually set weights for 12 default dense features: Language model probability and language model OOV count, five rule translation probabilities, word penalty and binary indicator features for source and source-target singletons, pass through rules and glue rules.[8] The second system, *mert*, optimizes weights for the 12 decoder features for translation quality using minimum error rate training (MERT). The *manual* weights are also used as input weights for MERT. The reported results are averaged over three runs of MERT to account for optimizer instability. The third baseline, *dtrain*, uses sparse lexicalized features (rule identifiers and target bi-grams) on top of the 12 decoder features. Feature weights are optimized for translation quality using the pairwise ranking optimizer DTRAIN[9] (Simianer et al., 2012). DTRAIN is trained for 10 epochs with a learning rate of $1e^{-5}$, a loss margin of 1, and a k-best size of 100. Final weights are averaged (Collins, 2002). This baseline is very similar to our experiments with ROPE, using the same pairwise ranking objective with the same sparse features, but optimizing for translation quality. The *mert* and *dtrain* baselines both use the NTCIR-8 development set (2,000 sentence pairs) for tuning. Unlike our benchmark system, we do not use multi-tasking while tuning.

[7] The code is no longer being developed, but a legacy fork is available on `https://github.com/pks/cdec-dtrain-legacy`.

[8] We use initial weights recommended by an earlier version of the CDEC documentation at `cdec-decoder.org/`, but no longer included in the current version. The features and weights are `LanguageModel=0.1, LanguageModel_OOV=-1, CountEF=0.1, EgivenFCoherent=-0.1, MaxLexFgivenE=-0.1, MaxLexEgivenF=-0.1, SampleCountF=-0.1, WordPenalty=-0.1, IsSingletonF=-0.01, IsSingletonFE=-0.01, PassThrough=-0.1, Glue=0.01`

[9] The code is no longer being developed, but a legacy fork is available on `https://github.com/pks/cdec-dtrain-legacy`.

At retrieval time, all queries are translated sentence-wise and subsequently re-joined to form one query per patent abstract. Our retrieval system uses the Okapi BM25 for document ranking (Spärck Jones, Walker, & Robertson, 2000), and returns the top $N = 1,000$ documents per query. We evaluate retrieval only on level 1 and 2 annotations (applicant and examiner citations). This setting more accurately represents the patent prior art search scenario, as family patents (level 3) contain published versions of a patent application at other patent organizations, rather than prior art. Additionally, retrieving family patents is much easier than retrieving cited patents. For metrics that focus on the top-ranked results, including family patents inflates scores. Besides NDCG, we evaluate retrieval performance according to three metrics: The total number of returned relevant documents (NUM_REL_RET), Mean Average Precision (MAP), and the Patent Retrieval Evaluation Score (PRES) (Magdy & Jones, 2010). MAP returns the averaged precisions over the top-1 up to n_j ranked documents for a query q_j, where n_j is equal to the number of relevant documents for q_j:

$$\text{MAP}(q_j) = \frac{1}{n_j} \sum_{k=1}^{n_j} \text{Precision}(R_k)\,,$$

where R_k refers to the set of the top-k ranked documents returned by the retrieval model (Definiton follows (Manning, Raghavan, & Schütze, 2008)). While MAP is precision-oriented, PRES has specifically been designed with patent prior art search in mind, and is more oriented towards recall, as finding all relevant prior art is crucial to this task. For a single query q_j with n_j relevant documents, considering the top k returned results, PRES is defined as

$$\text{PRES} = 1 - \frac{\frac{1}{n}\sum r_i - \frac{1}{2}(n+1)}{N}\,,$$

where r_i is the rank at which the i-th relevant document is retrieved. Magdy and Jones (2010) show experimentally that PRES prefers a system which returns more relevant documents within the top N results, regardless of their ranks, to a system which returns fewer relevant documents, but ranks them more highly.

4.5.2 Preliminary Analysis

ROPE approximates the search space over all possible translation hypotheses by using a k-best list. In this section, we conduct an analysis of the effect of several hyperparameters on the k-best list in order to motivate design choices and to gauge the oracle performance of our method. We analyze k-best lists generated by our baseline translation system. We compare retrieval performance of the oracle hypothesis and model-best translation, as measured by NDCG. As we are using multi-sentence queries, computing an oracle

weights	pop-limit	N	Viterbi NDCG	Oracle NDCG
manual	100	200	0.15473	0.194156
manual	100	1000	0.15473 ($\pm$0.0%)	0.209644 (+1.5%)
manual	200	1000	0.15286 ($-$0.2%)	0.213365 (+1.9%)
tuned (BLEU)	100	1000	0.15693 (+0.2%)	0.206901 (+1.3%)

Table 4.4 Effect of hyperparameters and tuning on one-best and oracle performance on 200 queries

with respect to retrieval quality is not straightforward. Finding the true oracle would require scoring all possible combinations of all k-best hypotheses, which is prohibitively expensive. We compute an approximate oracle using the following greedy approach: We first construct queries using the one-best translations of each sentence in a query. We then iterate over the sentences in the query, replacing the current sentence with each of the k-best translations and scoring this query. Finally, we combine the top-scoring translations for each sentence into a single search query. We generate k-best lists for 200 queries from the BoostCLIR training set. All k-best lists are generated using a filter which ensures that hypotheses differ by their surface string, and not just by their derivation.

Hyperparameters. Our first experiments investigate the effect of the length of the k-best list and the cube-pruning pop-limit. Table 4.4 compares retrieval quality on oracle queries and one-best queries. The difference between oracle and one-best NDCG is 4 percentage points. Increasing k-best size from 200 to 1,000 improved oracle NDCG by 1.5 percentage points. Raising the pop-limit and therefore allowing the model to explore a larger portion of the search space, produced an additional, but smaller, increase in oracle NDCG of 0.4 percentage points. One-best NDCG was actually decreased slightly. As both hyperparameters affect decoding speed, we increase k, but keep the pop-limit fixed at 100.

Initial Weights We then look at the difference between tuning decoder weights for translation quality and using manually set weights. When using weights tuned for translation quality, we observed a worse oracle NDCG compared to the manual weights, but a slightly higher one-best NDCG. This points to the conclusion that translation and retrieval quality are not independent. However, tuning for translation quality might produce a narrower range of k-best hypotheses, leading to decreased oracle NDCG. We therefore decide to initialize ROPE with manually set weights rather than weights optimized for translation quality.

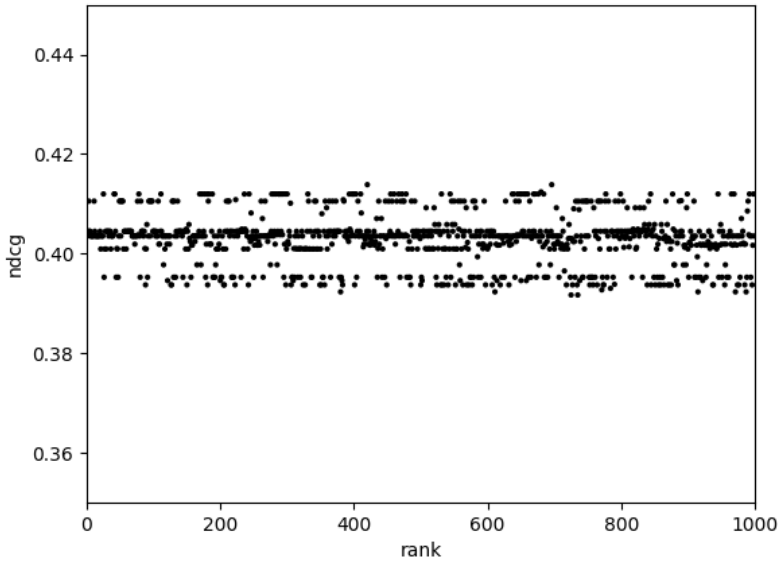
filtering	pop-limit	N	Viterbi NDCG	Oracle NDCG
`unique`	100	1000	0.15473	0.209644 (+1.5%)
`bow+sw`	100	1000	0.15473	0.231421 (+3.7%)
`bow+sw+patent`	100	1000	0.15473	0.231470 (+3.7%)

Table 4.5 Oracles for reranking on 200 queries

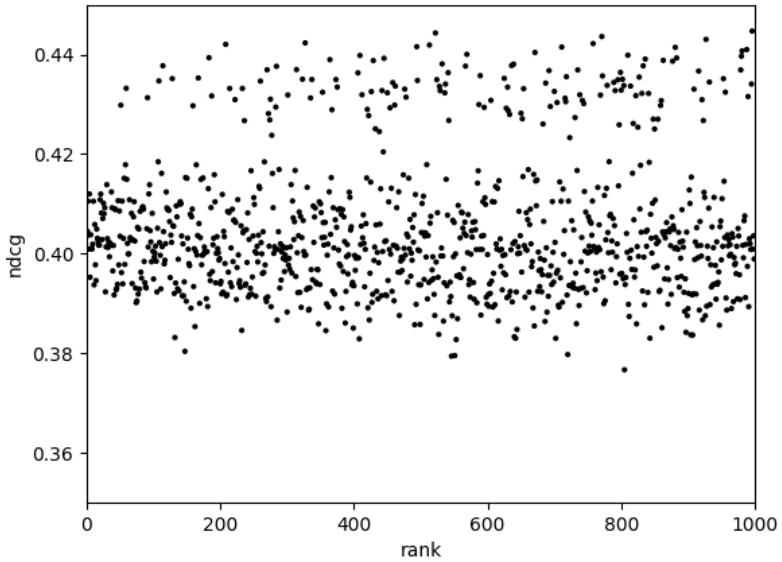
Filtering The baseline system uses a filtering scheme when producing k-best lists to ensure that all generated hypotheses differ by their surface string and not just by their latent derivation. Since retrieval uses stopword-filtering and bag-of-words representations of the queries, hypotheses differing only by word order will have identical scores. In order to generate more diverse queries, we introduce a modified filtering scheme. This filter only accepts a new hypothesis if its bag-of-words representation differs from previous hypotheses. Additionally, we incorporate stopword filtering to enforce the creation of hypotheses which differ by content words. We attempt to tailor the hypothesis generation even more to the task of patent retrieval by adding patent-specific terms to the stopword list. We construct a patent-specific stopword list by sorting all non-stopword terms according to frequency and adding the 500 most frequent terms to the stopword list.

Table 4.5 shows oracle NDCG for k-best lists generated with standard filtering (`unique`), bag-of-words filtering with general stopwords (`bow+sw`) and bag-of-words filtering with general and patent stopwords(`bow+sw+patent`). Gains in Oracle NDCG are calculated with respect to the first line in Table 4.4. We can see that introducing the `bow+sw` filter produced a substantially higher oracle NDCG, showing that it is important to explore the search space. Using additional stopwords from patents did not lead to an improvement. We explore both `unique` and `bow+sw` filtering experimentally.

Figure 4.1 illustrates the effectiveness of our filtering scheme. The figure shows an example plot of the NDCG-score for k-best translations by their ranks for an example k-best list with `unique` filtering (left) and `bow+sw` filtering (right). The left graph shows has a much larger range between maximum and minimum scoring hypothesis, and the top-scoring "tier" of hypotheses is not generated at all with `unique` filtering. This figure also shows no visible correlation between the ranking of hypotheses by the translation model and their retrieval performance.



(a) unique filtering



(b) bag-of-words + stopword filtering

Figure 4.1 Scatter plot of NDCG scores of top-1,000 translations for a single sentence according to model score with different k-best filters.

Lexical differences and retrieval quality. We qualitatively examine the discriminative power of individual lexical choices for retrieval quality using the same example as above. Figure 4.2 lists the best and worst scoring translations according to NDCG in the filtered k-best list from Figure 4.1 and marks content words, which differ between the translations.

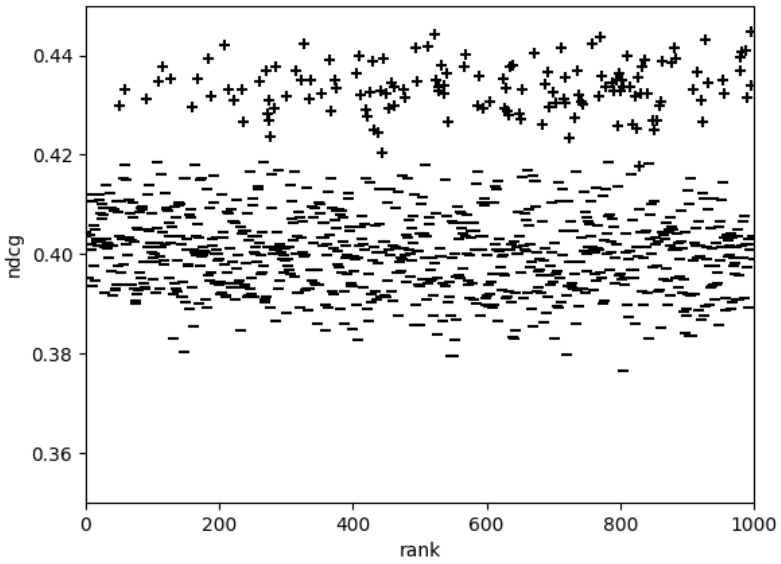
Best *solving* ***means*** *cast at least one portion of the rotor , the rotor and the rotor extending around rotor on both sides of the pair of side of the* ***superconducting***$^{+}$ *coil winding and having a coil winding of the rotor between the pair of side of at least one tension rod extending through the conduit and tension coil housing , and each of the rod ends of coil housing is* ***mounted*** *around the coil winding of a tension rod to* ***enclose*** *.*

Worst *solving cast at least one portion of the rotor , the rotor and the rotor extends around the rotor on both sides of the pair of side of the superconductive*$^{-}$ *coil winding and having a coil winding of the rotor between the pair of side of at least one tension rod extending through the conduit and tension coil housing coil housing is provided in each of the opposite ends of the rod of the coil winding tension position around the rod attached.*

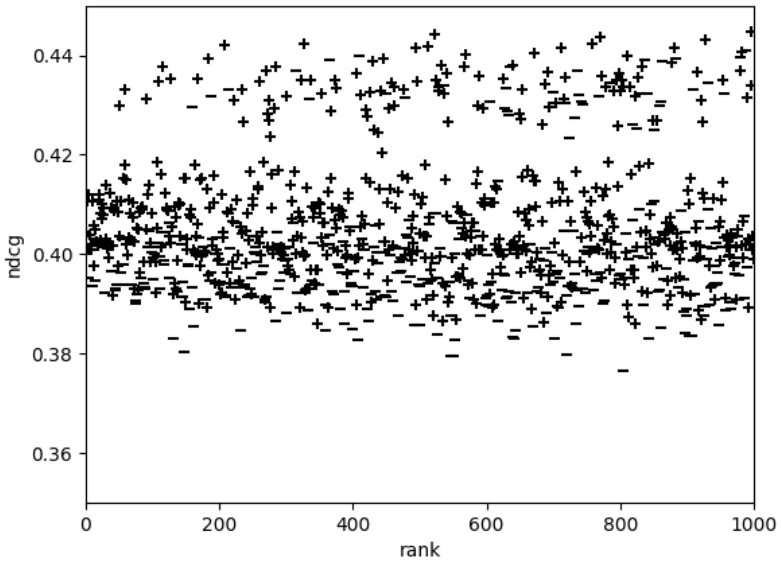
Figure 4.2 Best and worst scoring translations according to NDCG of a query using `bow+sw`-filtered k-best lists. **Bold-faced** terms occur only in the best translation, underlined terms occur only in the worst translation.

We can see that the translations differ by only few terms. Figure 4.3 on page 66 shows the same plot as Figure 4.1 (b), but marks each translation according to the presence or absence of a single term in the translation. Subfigure (a) shows figure that the presence or absence of the term "*superconducting*" in a translation almost perfectly separates the top and bottom clusters according to NDCG. In contrast, the presence or absence of the term "*mounted*" is not helpful. The term *superconducting* is very relevant for the patent at hand[10] , as can be seen by its appearance in the patent title ("*Rotor coil support for high-temperature* ***superconducting*** *synchronous machine equipped with tension rod, and method of assembling the coil support*"). Consequently, the translation choice of a single term can have a large impact on NDCG. We allow the model to learn the importance of certain words or phrases for retrieval by using sparse lexicalized features. As ROPE is able to handle large feature sets, we add them to our model.

[10]The cited patent's ID is JP-2003023766.



(a) superconducting



(b) mounted

Figure 4.3 Figure 4.1b, filtered according to the presence or absence of a single term ("*superconducting*" and "*mounted*"). "+" marks that the term is present in the translation, "-" marks its absence.

4.5.3 ROPE Training Setup and Hyperparameters

We train ROPE on 200 queries (798 sentences) from the BoostCLIR training set. We start from the same initial weights that are used for the *manual* baseline, as well as to initialize the *mert* baseline. Like the *dtrain* baseline we use rule identifiers and target bigram features. Following our analysis in Section 4.4, we use filtered k-best lists with k=1,000 and set the cube pruning pop-limit for k-best list generation to 100. For retrieval-based scoring during training, we sample 200 irrelevant documents per query and use MAP as retrieval metric. For ROPE, we use a constant learning rate η of $1e^{-6}$ and a loss margin of 1. We train ROPE for 10 epochs. Final weights are obtained by averaging over the updated weight vectors for each epoch. In order to investigate the effect of training on a weaker signal, we conduct an experiment using all relevance levels, as well as an experiment using only levels 1 and 2 in training.

4.5.4 Results

Table 4.6 on page 68 shows results for the three baselines and ROPE. These results are obtained by evaluating the systems on 200 heldout queries from the BoostCLIR training set. When comparing the baselines, we see that the *mert* baseline with tuned weights improves over the *manual* baseline according to the PRES metric and, very slightly, NDCG, while causing a decrease in MAP. This result indicates that there is only a weak correlation between translation quality and retrieval quality. Comparing *dtrain* to *manual*, we observe a reverse effect. Adding sparse features improves MAP, but leads to a slight decrease in PRES and NDCG. Since MAP is precision-oriented, while PRES takes recall into account more strongly, the baseline results indicate that *mert* tuning increases recall at the cost of precision , while DTRAIN with sparse features increases precision at the cost of recall.

Training ROPE only on levels 1 and 2 (line 4), produces worse results than the baselines, showing that the family citations are crucial for success. A possible explanation for this gap is that the weaker training signal from levels 1 and 2 is a very poor fit for the translation model, and guides the model to produce outputs that are no longer adequate translations. ROPE, trained on all levels, is able to outperform the baseline systems according to all retrieval metrics, leading to better precision, as well as recall. This result confirms that optimizing a query translation SMT system using downstream task performance leads to more task-appropriate translations. As the bag-of-words filtering scheme for generating k-best lists during training does not improve results on the heldout set, we revert to the faster unique filtering scheme for further experiments.

	train setting	NUM_REL_RET	MAP	NDCG	PRES
1	*manual* baseline	758^{3}	0.048^{2}	0.1498^{3}	0.2798^{3}
2	*mert* baseline	$761^{1,3}$	0.0466	$0.1503^{1,3}$	$0.2922^{1,3}$
3	*dtrain* baseline	730	$0.0515^{1,2}$	0.1496	0.2751
4	ROPE (levels 1,2)	712	0.042	0.1383	0.2717
5	ROPE (all levels)	798	$0.0567^{1,2,3}$	$0.1636^{1,2,3}$	$0.3043^{1,2,3}$
6	ROPE (all levels) `+bow`	$768^{1,2,3}$	$0.0562^{1,2,3}$	$0.1615^{1,2,3}$	$0.2962^{1,2,3}$

Table 4.6 Results for JP-EN patent retrieval on heldout data. Numbers in superscript indicate that the result is better than the baseline with this row number.

	train setting	NUM_REL_RET	MAP	NDCG	PRES
1	*manual* baseline	$13940^{2,3}$	$0.1486^{2,3}$	$0.3323^{2,3}$	$0.5505^{2,3}$
2	*mert* baseline	13936^{3}	0.1454^{3}	0.3287^{3}	0.5458^{3}
3	*dtrain* baseline	13349	0.1359	0.3136	0.5218
4	ROPE (levels 1,2)	13387	0.1262	0.3047	0.5199
5	ROPE (all levels)	$14070^{1,2,3}$	$0.1499^{1,2,3}$	$0.3346^{1,2,3}$	$0.5519^{1,2,3}$

Table 4.7 Results for JP-EN patent retrieval on test set. Numbers in superscript indicate that the result is better than the baseline with this row number.

Table 4.7 shows results on the BoostCLIR test data. Here, we observe a different ranking: The *mert* baseline clearly outperforms the *dtrain* baseline with respect to all evaluation metrics. Both tuned baselines perform worse than the manually selected weights used for the *manual* baseline. This result suggests that optimizing for BLEU will not yield the best translation system for cross-lingual retrieval. ROPE, again, deteriorates in all metrics if only on levels 1 and 2 are used as a training signal. When ROPE is trained on all levels, however, the highest overall scores are achieved. However, the difference to the *manual* baseline is small. This points to the conclusion that ROPE shows promise, but more training might help to achieve larger gains.

Conclusions

This chapter has presented a case study where full supervision is available for end-to-end translation, but where document meta-information is used to adapt an SMT model to a different multilingual application with no explicit task-specific supervision. Document meta-information in the form of cross-lingual citations between documents has been used to simulate a downstream task loss for the task of cross-lingual patent prior art search. In order to set up this task, the citation information from a large collection of Japanese-English patent documents has been exploited to build a large-scale training and evaluation set for simulating patent prior art search. A competitive baseline translation system has also been trained, which uses available parallel training data from the patent domain.

A tailored retrieval-optimized version of the pairwise ranking perceptron algorithm, ROPE, has been introduced, based on pairwise ranking optimizers for SMT. While the theoretic changes to the algorithm are straightforward, employing the algorithm with a retrieval metric instead of a translation quality metric requires solutions for two issues related to the task. The first issue, the high computational complexity of retrieval-based scoring, was solved by sampling a small set of documents from the document collection. The second issue, the use of multi-sentence queries, was solved by treating queries as a variably-sized minibatch, where an update is only computed after scoring the entire query, and by using the Viterbi context for scoring individual sentences in a query.

The choice of features and hyperparameters for ROPE have been motivated by analyzing oracles found in k-best lists generated by our baseline systems. Unsurprisingly, increasing the number of generated hypotheses or the size of the search space, has produced better oracles. A bag-of-words based filtering scheme for hypotheses has also produced more diverse hypotheses, leading to a further jump in oracle performance. We finally presented an example were the translation of a single word was responsible for a significant jump in NDCG score. This motivated our decision to include sparse lexicalized features in our system.

ROPE has been tested in a proof-of-concept evaluation on a training set of 200 queries. Even with this small training set, numeric improvements with respect to all retrieval metrics over the baselines have been observed, showing the promise of retrieval-optimized training for query translation. When comparing retrieval performance of different baselines, the baselines optimized for translation quality are not unequivocally better than the baseline with manually set weights. This result suggests that tuning for translation quality does not necessarily improve retrieval performance. However, it was essential for

good performance to include family patents in the training sets used with ROPE. This result suggests that ensuring good translation quality still matters for query translation.

The improvements achieved with ROPE have been smaller than in other recent work on cross-lingual patent retrieval, which pursues a more integrated approach to patent translation (Hieber & Riezler, 2015). However, this approach leads to an increase in complexity at test time, as the translation model's search space is used for scoring documents in retrieval, requiring D decoding runs per query in the worst case. With our approach, complexity is high during training as translation hypotheses have to be evaluated by conducting a retrieval step, but, as the approach is agnostic to the retrieval system, complexity at test time is reduced to decoding the search query once and running efficient monolingual retrieval.

Applicability to Neural Machine Translation (NMT). While our experiments have been conducted using an SMT system, ROPE itself is easily adaptable to an NMT system. As NMT systems use left-to-right beam search for translation, generating $\mathcal{K}(x)$ is straightforward. Equation 4.1 can be replaced by the more general

$$L_{perc\_rank} = \sum_{(y,y')\in\mathcal{P}(x)} \max\left(0, s_w(x,y') - s_w(x,y)\right) , \tag{4.2}$$

where $s_w(x,y)$ is the score assigned to translation y, given input x by a neural network with parameters w. The generation of preference pairs (y,y') only uses the retrieval scoring and is thus independent of the translation model.

CHAPTER 5

Bipolar Ramp Loss for Learning from Weak and Negative Supervision

Document-level links can serve as a weak surrogate training signal when strong supervision from parallel data is not available. We have shown this for patent query translation, where document-level links can be used to simulate a downstream application. But could cross-lingual document-level links also be used as a weak supervision signal for end-to-end translation? This chapter presents a case study where in-domain full supervision is unavailable, but where a large bilingual in-domain document collection with a cross-lingual link structure exists. Our goal is to utilize this weak in-domain supervision directly to adapt an out-of-domain system to the target domain. We employ a structured ramp loss objective, which promotes a *hope* translation and demotes a *fear* translation using the prediction of a pre-trained model along with an external metric, which incorporates weak supervision from document-level links. By updating towards a *hope* output and not towards a reference, structured ramp loss is particularly well-suited to a weak supervision scenario. By including the model prediction and the external metric in selecting the *hope* and *fear* output, the objective allows us to trade off between the model's predictions and an external metric. It also lets us strengthen the supervision signal by including negative supervision from irrelevant documents. The characteristic of deliberately selecting positive and negative outputs to promote and demote sets the structured ramp loss apart from other objectives. We refer to an objective, which has this ability, as "bipolar".

This chapter studies the translation of articles from Wikipedia[1] as an example scenario. Wikipedia is a large, publicly available resource, which contains stronger and weaker cross-lingual links. As articles across languages are being edited independently, Wikipedia is, at best, a comparable data set, making our weak supervision scenario plausible. Better automatic translation of Wikipedia articles is desirable, as it aids the propagation of knowledge across languages. While we focus on Wikipedia, there are many other scenarios where a cross-lingual link structure in a large multilingual collection could be exploited: Patent documents are linked by citations; social media posts are linked by a user network or by user-generated tags; a product taxonomy can be used to infer stronger or weaker links for product descriptions or reviews in e-commerce data. Unlike previous chapters where the solutions are tailored to a specific scenario, this study aims to provide a more general approach, which is applicable across data sets. Hence, no information specific to only Wikipedia is included in the approaches.

Given the above scenario and a general, model-agnostic ramp loss framework for learning from document-level links, we spell out approaches for statistical and neural machine translation (SMT/NMT) models. For SMT, we investigate the potential of different link strengths for tuning sparse lexicalized features. A loss-augmented inference method is described, which directly incorporates the external metric into the candidate generation step and therefore leads to more diverse *hope* and *fear* outputs. This method is compared to a metric-agnostic approximation of the search space via k-best lists. In order to learn from weaker links, a weighted sampling strategy is introduced. This strategy makes use of unsupervised cross-lingual document similarity to boost the sampling of documents with higher similarity. Both loss-augmented inference and weighted sampling are found to be crucial for successful learning. For NMT, the focus is on investigating the potential of combined positive and negative supervision from relevant and irrelevant documents in this extremely weak supervision scenario. Several versions of the ramp loss, which incorporate either only positive, only negative, or both types of supervision, are evaluated experimentally. In line with prior work, these objectives are compared to minimum risk training. Experiments show that for both ramp loss and minimum risk training, it is crucial to include positive and negative supervision. Bipolar ramp loss, which intelligently selects a specific positive output to promote and a specific negative output to demote, is found to be superior to minimum risk training, which computes an expectation over a sample of outputs. As structured ramp loss has not been applied to NMT before, we also conduct an experiment, which compare ramp loss and minimum risk training in a fully supervised setting, where we are able to confirm the superiority of bipolar ramp loss.

[1] https://en.wikipedia.org

The chapter makes the following contributions: (1) Bipolar structured ramp loss objectives for a weak supervision scenario, where supervision is provided via document-level links, including the definition of a new external metric, which can take into account positive and negative feedback. (2) A learning procedure and evaluation based on this loss in an SMT system using loss-augmented inference, large-scale parallel training and weighted sampling. (3) Transfer of the bipolar ramp loss to a neural sequence-to-sequence model and comprehensive analysis of different ramp loss variants in comparison to minimum risk training in weakly supervised and fully supervised scenarios. To our knowledge, we are the first to apply this objective to neural machine translation. (4) The definition of a weakly supervised Wikipedia translation task, along with the construction of a small in-domain evaluation set from automatically aligned sentences and an analysis of the potential of extracting parallel data depending on the strength of document-level links. All contributions were made by the author after consultation with the supervisor of this thesis. The experiments build on the WikiCLIR data set created by Schamoni, Hieber, Sokolov, and Riezler (2014). The approach and experiments with an SMT system were published in Jehl and Riezler (2016). The work on NMT is also included in Jehl et al. (2018) (under submission). The general, model- and metric-agnostic ramp loss formulation, as well as the token-level variant of the ramp loss evaluated in the NMT experiments, were both first suggested by a co-author of this work.

The chapter is organized as follows: Section 5.1 discusses related work. Section 5.2 contains a model-agnostic definition of the structured ramp loss and its transfer to the weakly supervised setting. Section 5.3 describes the Wikipedia data set and its potential for parallel data extraction, and describes the construction of the evaluation data. Section 5.4 describes how to tune sparse lexicalized feature weights using structured ramp loss with weak supervision from document-level links for a hierarchical phrase-based system, and reports experimental results on Wikipedia. Section 5.5 describes ramp loss variants and the MRT baseline for NMT in the weakly supervised scenario and presents experimental results on the Wikipedia task. Section 5.6 describes our fully supervised NMT experiment.

5.1 Related Work

Our learning objective is a modified version of the structured ramp loss objective. The ramp loss was originally proposed for binary classification by Collobert, Sinz, Weston, and Bottou (2006) as an alternative to hinge loss. Chapelle, Do, Teo, Le, and Smola (2009) then adapted it to structured prediction. For linear models, ramp loss has been shown to be consistent, and learning can be easily implemented using a *perceptron-like* update (Hazan, Keshet, & McAllester, 2010). Gimpel and Smith (2012) and Chiang (2012) apply ramp

loss objectives for SMT tuning under full supervision. Gimpel and Smith (2012) compare known loss functions in a *structured ramp loss* framework to several ramp loss formulations. Running approximate loss-augmented inference over a translation lattice or forest, as we describe for weakly supervised SMT, has also been explored for full supervision by Tromble, Kumar, Och, and Macherey (2008) and Chiang (2012). We apply structured ramp loss to weakly supervised SMT, and, to our knowledge, are the first to apply it to NMT. Most previous approaches to training neural translation models with sequence-level objectives and feedback from external metrics has focused on minimum risk training (MRT) (S. Shen et al., 2016) or approaches (Ranzato, Chopra, Auli, & Zaremba, 2016, inter alia) based on REINFORCE (Williams, 1992). Edunov, Ott, Auli, Grangier, and Ranzato (2018) compare MRT to other classical structured prediction losses, which they lift to a fully supervised neural machine translation setting, and find MRT to perform best. However, their best-performing setup relies on interpolation with a token-level maximum likelihood objective, which is not applicable in our weak supervision scenario. Nevertheless, we compare our structured ramp loss objectives to MRT.

There are different ways to use large cross-lingually linked document collections for adaptation: Monolingual language model adaptation for SMT ignores cross-lingual links, but uses the target side of the collection (see e.g. Koehn et al. (2007) or Foster and Kuhn (2007)). Cross-lingual links have also been used to filter the multilingual collection for sentence-parallel data. Besides our work in Chapter 3, Munteanu and Marcu (2005) extract parallel sentences from online news, Smith et al. (2013) filter the Common Crawl, and Wołk and Marasek (2015) describe an approach for finding parallelism in Wikipedia. Our aim in this chapter is not to replace these methods but rather to provide an approach, which can be used with less strongly linked documents, as well as in combination with these other methods. For SMT, we experimentally test different strengths of links between document, and analyze what strength is required to extract parallel data from Wikipedia.

5.2 Structured Ramp Loss

The structured ramp loss aims at promoting a good *hope* output y^+ and demoting a bad *fear* output y^-. We use the following general definition of the ramp loss

$$L_{RAMP} = \frac{1}{M} \sum_{m=1}^{M} -s_w(y_m^+, x_m) + s_w(y_m^-, x_m)\,, \tag{5.1}$$

where $s_w(x, y)$ is the scoring function defined by the translation model under parameters w. The loss is averaged over a minibatch of M inputs. Different definitions of y^+ and y^- then give rise to different concrete loss functions. y^+ and y^- are defined with respect to

the scoring function and a metric $\delta(y)$, which measures the quality of y with respect to external supervision. Ramp loss is particularly well-suited to a weak supervision setting for three reasons. First, it does not require a gold standard reference translation to update towards.[2] Second, by incorporating both the model prediction and the external metric in the selection of y^+ and y^-, it further allows to trade off between both scores. This is especially useful if the external metric is unreliable. Third, and most importantly, it not only selects a particular positive translation y^+ to update towards, but also selects a particular negative translation y^- to move away from. This property, which we call "*bipolar*", allows us to also include negative feedback, which can be a useful supplement for a weak positive supervision signal.

5.2.1 Fully Supervised Setting

For the sake of clarity, we first define the structured ramp loss for a fully supervised setting, before describing our adaptations for the weakly supervised setting. In a fully supervised translation task, we have access to one or more gold standard reference translations y^* for each training input x. We can then use a smoothed per-sentence approximation of BLEU as our metric $\delta(y)$. We refer to this metric as $\text{BLEU}_{+1}(y, y^*)$. We then define y^+ and y^- as

$$y^+ = \operatorname*{argmax}_y s_w(y, x) - \alpha(1 - \text{BLEU}_{+1}(y, y^*)) \tag{5.2}$$

and

$$y^- = \operatorname*{argmax}_y s_w(y, x) + \alpha(1 - \text{BLEU}_{+1}(y, y^*))\,. \tag{5.3}$$

This version of the structured ramp loss was also applied to fully supervised SMT by Gimpel and Smith (2012) and Chiang (2012). Note that the per-sentence BLEU metric is converted into a cost function by subtracting it from 1. We follow Gimpel and Smith (2012) in including the non-negative scaling parameter α, which controls the importance of the metric, compared to the model score.

Gimpel and Smith (2012) use a concave-convex batch procedure to train their model with structured ramp loss. As batch training is inefficient for larger training sets, we instead follow (Hazan et al., 2010) and (Chiang, 2012) and use a *perceptron-like* online learning procedure using stochastic sub-gradient descent. Algorithm 3 on page 76shows the general training procedure, which is independent of the translation model and the

[2] Coincidentally, this property of the ramp loss is useful for SMT tuning in general. In an SMT model, an external reference can often not be generated by the model, and therefore, computing a feature representation of such a reference is not possible. For other objectives, which require a feature representation of the reference, a common strategy is to resort to using a surrogate pseudo-reference, which can be generated by the model. For ramp loss, the issue does not arise at all.

Algorithm 3 Ramp Loss Training

Require: training set X, Y; minibatch size M, initial parameters w, learning rate η

while stopping criterion not met **do**
  $\{x_m, y^*_m\}_{m=1}^{M} \leftarrow$ sample minibatch of size M
  $B \leftarrow \emptyset$ ▷ initialize new minibatch
  for $m = 1 \dots M$ **do**
    Obtain y^+_m as specified by ramp loss function
    Obtain y^-_m as specified by ramp loss function
    $B \mathrel{+}= \{(x_m, y^+_m), (x_m, y^-_m)\}$ ▷ add y^+ and y^- to minibatch
  end for
  $w \leftarrow w + \eta \frac{1}{M} \sum_{m=1}^{M} \nabla_w s_w(y^+_m, x_m) - \nabla_w s_w(y^-_m, x_m)$
end while

specific loss function, in a minibatch version. An online version can be obtained by setting the minibatch size M to 1.

In machine translation, search for y^+ and y^- in the space of all possible outputs $\mathcal{Y}(x)$ has to be approximated as the output space is too large. This is often done by restricting search to a candidate set $\mathcal{K}(x)$ of k-best hypotheses under the current model.

5.2.2 Weakly Supervised Setting

In the weakly supervised scenario considered here, an out-of-domain model is to be adapted to a target domain without access to gold standard references in the target domain. The weak supervision comes from relevance information provided by cross-lingual document-level links, such that we can obtain for each input x a set of *relevant* target documents $D^+(x)$. The search for y^+ and y^- is then guided by a relevant document d^+ sampled from $D^+(x)$. Unlike a reference translation, a relevant document is an imperfect label. Guiding the translation model to reproduce d^+ would not generate good translations. We construct a more informative supervision signal by additionally sampling an irrelevant document d^- from a set of contrast documents in the target language. For each input x, we then obtain a pair (d^+, d^-) as our weak supervision signal.

Computing a BLEU score between a translation y and a document d does not make sense. We therefore define two new metrics with respect to the weak supervision:

1. $\delta_1(y, d)$ computes how well a translation matches a single document d.

2. $\delta_2(y, d^+, d^-)$ computes how well a translation differentiates between a document pair.

We first define $\delta_1(y, d)$. This metric should evaluate how closely an output matches this document, while still ensuring that the output has the correct length. It should also be similar to the BLEU score, which will be ultimately used for evaluation. We define $\delta_1(y, d)$ as the average n-gram precision between an output and a document.

$$\delta_1(y, d) = \frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{u_n} c(u_n, y) \cdot \mathbb{1}_{u_n \in d}}{\sum_{u_n} c(u_n, y)} \cdot BP\,, \tag{5.4}$$

where u_n are the n-grams present in y, $c()$ counts the occurrences of an n-gram in y and N is the maximum order of n-grams used. $\mathbb{1}_{u_n \in d}$ returns 1 if u_n is present in document d and 0 otherwise. BP is a brevity penalty controlling the length of the translation, $|y|$. Since we cannot use the reference length, we use the source length as reference length $|x|$, multiplied by an average source-target length factor r, which can be empirically determined on the training set. The brevity penalty is then defined as

$$BP = \min(1, \frac{r \cdot |y|}{|x|})\,.$$

$\delta_2(y, d^+, d^-)$ is defined as the difference between $\delta_1(y, d^+)$ and $\delta_1(y, d^-)$, subject to a linear transformation to allow values to lie between 0 and 1:

$$\delta_2(y, d^+, d^-) = 0.5 \cdot (\delta_1(y, d^+) - \delta_1(y, d^-) + 1)\,. \tag{5.5}$$

Our intuition behind this metric is that it should measure how well a translation differentiates between the relevant and irrelevant document, leading to domain-specific translations being weighted higher than domain-agnostic translations.

We are now equipped to define our concrete ramp losses for the weakly supervised case. Using δ_1, we define y^+ analogously to the full supervision task as

$$y^+_{\delta_1} = \operatorname*{argmax}_y s_w(y, x) - \alpha(1 - \delta_1(y, d^+))\,. \tag{5.6}$$

$y^+_{\delta_1}$ only includes d^+, hence we would like to define $y^-_{\delta_1}$ in such a way that it includes d^-. As d^- is a source of negative supervision, we propose to use as $y^-_{\delta_1}$ a translation that matches d^- well. We define $y^-_{\delta_1}$ as

$$y^-_{\delta_1} = \operatorname*{argmax}_y s_w(y, x) - \alpha(1 - \delta_1(y, d^-))\,. \tag{5.7}$$

Algorithm 4 Ramp Loss Training with Weak Supervision

Require: Inputs X with relevant documents $D^+(x)$ for all $x \in X$
1: Contrast document collection D^-
2: Minibatch size M, learning rate η, initial weights w
3: **while** stopping criterion not met **do**
4: $\{x_m\}_{i=1}^M \leftarrow$ sample minibatch of size M
5: $B \leftarrow \emptyset$ ▷ initialize new minibatch
6: **for** $m = 1 \dots M$ **do**
7: Sample d^+ from $D^+(x_m)$ and d^- from D^-
8: Obtain y_m^+ as specified by loss function
9: Obtain y_m^- as specified by loss function
10: $B \mathrel{+}= \{(x_m, y_m^+), (x_m, y_m^-)\}$ ▷ add y^+ and y^- to minibatch
11: **end for**
12: $w \leftarrow w + \eta \frac{1}{M} \sum_{m=1}^M \nabla_w s_w(y^+, x_m) - \nabla_w s_w(y_m^-, x_m)$
13: **end while**

Note that unlike the supervised case, the cost is still subtracted from the model score. As δ_2 already includes d^-, using this metric we can define y^+ and y^- analogously to the supervised case as

$$y_{\delta_2}^+ = \operatorname*{argmax}_y s_w(y, x) - \alpha(1 - \delta_2(y, d^+, d^-)) \tag{5.8}$$

and

$$y_{\delta_2}^+ = \operatorname*{argmax}_y s_w(y, x) + \alpha(1 - \delta_2(y, d^+, d^-)) \,. \tag{5.9}$$

Algorithm 4 shows the adapted training algorithm in the weak supervision setup. The main difference to Algorithm 3 is line 5, where the weak supervision signal is sampled from the relevant documents and contrast documents.

5.3 Experimental Setup and Data Analysis

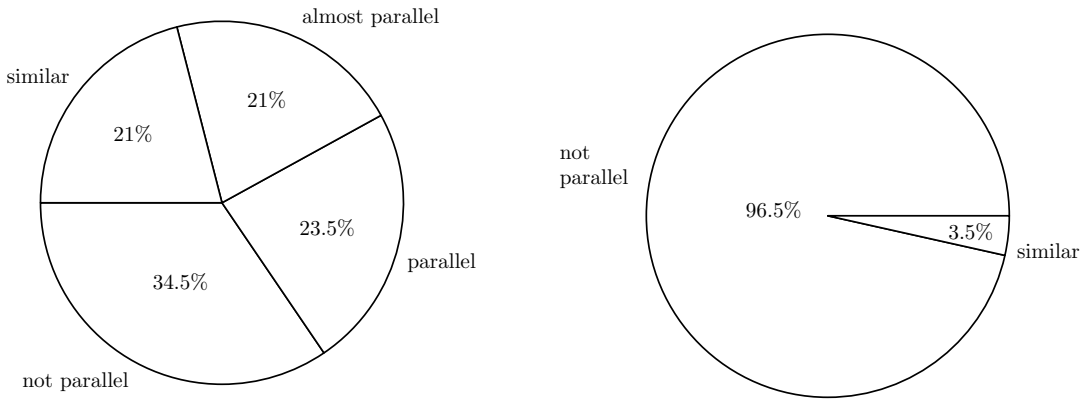
We apply our approach to the task of translating Wikipedia entries. Being able to automatically translate Wikipedia entries would allow the proliferation of knowledge across more languages by providing a starting point for editors if an article already exists in another language. It could also be used to augment existing articles with missing parts translated from another language. We choose Wikipedia, because it fits our envisaged scenario perfectly: It is a large multilingual collection containing document-level cross-lingual links, and there are no parallel data available for this task. Wikipedia is internally structured by interlanguage links and inter-article links. *Interlanguage links* connect arti-

cles on the same topic across languages. While entries connected by these links often have the same subject, they are not necessarily parallel. Nonetheless, these connections provide a good starting point for automatic parallel data extraction. *Inter-article links* connect articles in the same language. These articles can be closer or more distantly related, but they do not have the same subject. It is not clear whether these articles contain any parallel data. In this work, we focus on interlanguage and inter-article links, but additional link structure could also be extracted from images embedded in text or the categorization of articles.

We use the German-English WikiCLIR collection by Schamoni et al. (2014)[3], along with their definition of cross-lingual relevance levels: A target language document is highly relevant to a source document if there exists an interlanguage link between the source and target document. WikiCLIR calls this the *mate* relation. A target document is weakly relevant to a source document, if there exists a bidirectional link between the source-document's cross-lingual mate and the target document. A bidirectional link exists between two documents, if both reference each other via an inter-article link. WikiCLIR calls this the *link* relation. The corpus contains a total of 225,294 mate relations with one average mate per English document, and over 1.7 million link relations, with on average 8.5 links per English document. The search queries provided in WikiCLIR are designed for a cross-lingual retrieval task. They are truncated to the first 200 words of a document, and words occuring in the article title have been removed. As our task is article translation, we use the full Wikipedia documents rather than the truncated queries.

We use the linked Wikipedia data to run an automated pseudo-parallel data extractor. We do this for three purposes: First, to identify nearly parallel document pairs for the construction of an in-domain evaluation set without having to rely on manual translation. Second, to examine whether the *mate* and *link* relation provide a strong enough signal for extracting pseudo-parallel training data. Third, to compare our method to automatic parallel data extraction based on cross-lingual document-level links. We use the modified `yalign` method described by Wołk and Marasek (2015) for pseudo-parallel data extraction.[4] We adapt their software to handle the WikiCLIR format. `yalign` requires a bilingual dictionary with translation probabilities. Following Wołk and Marasek (2015), we use a lexical translation table created from the IWSLT parallel training data[5] as our bilingual dictionary. We filter the dictionary for punctuation and numerals and discard all entries whose lexical translation probability is smaller than 0.3. We run the sentence aligner twice, using both the mate and link relation to align documents.

[3] `cl.uni-heidelberg.de/wikiclir/`
[4] `github.com/krzwolk/yalign`
[5] `wit3.fbk.eu/`



(a) Parallelism in mates **(b)** Parallelism in links

Figure 5.1 Sentence aligner precision for mates and links.

Figure 5.1 shows an analysis of `yalign`'s precision for the mate and link relations.For each case, a sample of 200 automatically aligned sentence pairs was manually evaluated. The sentence pairs were annotated using four categories: "parallel", "almost parallel" – this category contains sentence pairs that have strictly parallel segments, with other segments missing from the aligned part, "similar" – for sentence pairs that have similar content or wording but are not strictly parallel –, and "not parallel". While 65.5% of sentence pairs from the mate relation are similar or parallel, the link relation yields only 3.5% similar sentence pairs. We conclude that the bidirectional link relation is too weak to extract useful pseudo-parallel data.

To gain an idea of the yield of the aligner, we also look at the number of sentence pairs that were extracted from the paired documents. Figure 5.2 shows the frequency histogram of the number of extracted lines per document pair for document pairs with a mate relation. For most document pairs, only a single sentence pair was extracted. However, there were a few pairs that yielded several hundred pseudo-parallel sentence pairs. In total, 533,516 sentence pairs were extracted.

To construct our in-domain evaluation data, we sorted all automatically aligned documents by the number of aligned sentences up to a limit of 10,000 sentences. We then selected eight document pairs, discarding other document pairs which appeared to have been machine-translated, only contained few parallel sentences, or consisted of lists of proper names. We manually corrected sentence splitting errors in the selected documents, and removed image captions and references. We split the documents into two groups of four, making sure to keep the sets topically diverse. Table 5.1 shows the two sets of extracted documents. They are similar in length (1,712 sentences for Wiki1, 1,526 sentence for Wiki2), and contain

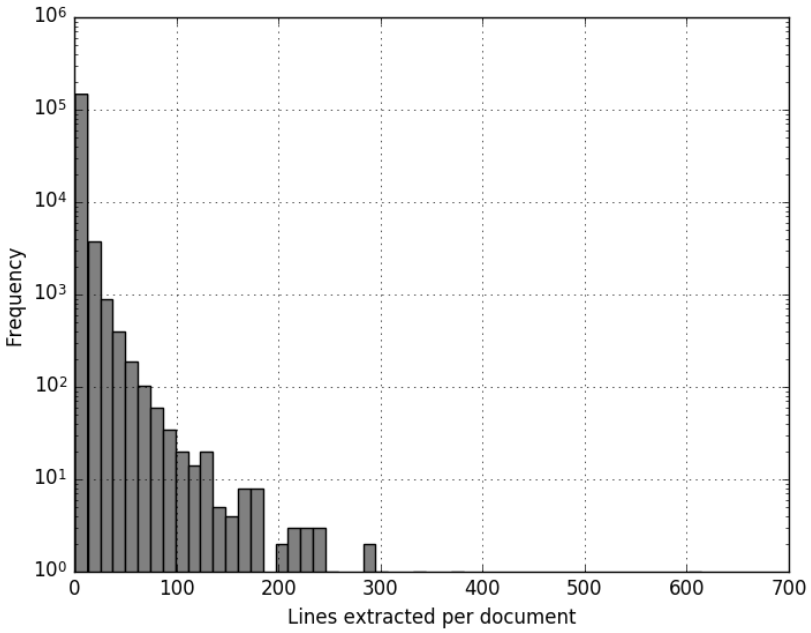


Figure 5.2 Number of documents (y-axis, on log-scale) from which n lines were extracted (x-axis).

set	total sentences	parallel sentences	article title
WIKI1	323	285	"*Polish culture during World War II*"
	710	677	"*Black-figure pottery*"
	457	375	"*Ulm Hauptbahnhof*"
	587	375	"*Characters of Carnivàle*"
WIKI2	360	268	"*J-pop*"
	501	388	"*Schüttorf*"
	549	438	"*Military history of Australia during World War II*"
	676	432	"*Arab citizens of Israel*"

Table 5.1 Wikipedia development and test documents.

a considerable percentage of parallel sentences. We used WIKI1 as heldout validation set, and WIKI2 for testing.

5.4 Exploring Relevance Levels and Combined Adaptation Methods for SMT

5.4.1 Objective and Implementation

We assume a log-linear translation model, where each output sequence y is associated with a latent derivation h. A triple of input sequence x, translation y, and latent derivation h is represented by a feature vector $\phi(x, y, h)$, which is weighted by a weight vector $\mathbf{w}$ of equal length. The model score of a translation y yielded by a derivation h is computed as

$$s_w(x, y, h) := \frac{\exp(\mathbf{w}^\top \phi(x, y, h))}{\sum_{y', h'} \exp(\mathbf{w}^\top \phi(x, y', h'))} \propto \mathbf{w}^\top \phi(x, y, h)\,.$$

For the experiments in the weakly supervised scenario, the two variants of the ramp loss described in Section 5.2 are explored. Table 5.2 lists the two variants, which we call RAMP^- and RAMP_{δ_2} for weakly supervised SMT. Here, we use the *cost* functions $\Delta_1(y, d^+)$ and $\Delta_2(y, d^+, d^-)$ as shorthand for $\alpha(1 - \delta_1(y, d^+))$ and $\alpha(1 - \delta_1(y, d^+, d^-))$, respectively.

After obtaining the translations y^+ and y^- according to the ramp loss objective along with the derivations h^+ and h^-, the stochastic subgradient update is computed as

$$\mathbf{w} \leftarrow \mathbf{w} + \eta(\phi(x, y^+, h^+) - \phi(x, y^-, h^-))\,.$$

We use a hierarchical phrase-based translation model as implemented in the CDEC toolkit (Dyer et al., 2010), which we augment by our implementation of the weakly supervised ramp loss objectives. Below, we describe some details of our implementation.

Forest search. The process of finding *hope* and *fear* hypotheses is called *loss-augmented inference*, as the model score is augmented by a loss function based on external performance metrics. Loss-augmented inference is often conducted over the k-best outputs. We test this approximation in our experiments. However, k-best lists cover only a very small portion of the possible hypothesis space, and the external metric does not influence their generation. We therefore implement *forest search*, a version of loss-augmented inference,

Loss	y^+	y^-
RAMP$^-$	$\text{argmax}_y\, s_w(y,h,x) - \Delta_1(y,d^+)$	$\text{argmax}_y\, s_w(y,h,x) - \Delta_1(y,d^-)$
RAMP$_{\delta_2}$	$\text{argmax}_y\, s_w(y,h,x) - \Delta_2(y,d^+,d^-)$	$\text{argmax}_y\, s_w(y,h,x) + \Delta_2(y,d^+,d^-)$

Table 5.2 Configurations for y^+ and y^- for weakly supervised SMT adaptation.

which integrates an approximation of the external metric into the search for the model-best hypothesis.

CDEC takes a two-pass approach to decoding. In the first pass, an unpruned hypergraph structure is built up from all potential parses of the input including only rule-local features. In the second pass or rescoring pass, this hypergraph is scored with the non-local n-gram language model via finite state intersection. As the language model requires storing the previous $n-1$ words at each gap, copies need to be made for each hyperedge according to different histories. Without pruning, this would cause the hypergraph to grow to unmanageable size very quickly. This problem can be solved by cube pruning (Chiang, 2007), which rescores the hypergraph bottom-up, lazily computing only the most promising candidates at each intersection step.

As our metric is based on n-gram precision, we make use of the fact that this information is present locally in the rescored forest. By using a rule-local approximation of our metric, which can be computed independently at each rescored edge, we can obtain y^+ or y^- directly by running the Viterbi algorithm over the rescored hypergraph (see also Section 5 of Chiang (2012)). We approximate the sentence-level metric $\delta_1(y,d)$, which computes a macro-average over the n-gram precision in a translation by a localized version $\delta_1(e,d)$ which computes a micro-average over the n-gram precision at each hyperedge e. δ_2 is then computed by combining $\delta_1(e,d^+)$ and $\delta_1(e,d^-)$ according to Equation 5.5.

The *cost* of traversing a hyperedge according to the metric δ, $\Delta(e)$, is then computed by subtracting the metric from 1 and multiplying the result by the scaling factor α. Finally, the global cost of a translation is computed by summing over the costs of all hyperedges participating in the translation's derivation. The hyperedges are annotated with the localized metric during the rescoring pass over the translation forest, when the language model intersection is computed. This ensures that derivations which have lower cost, when searching for *hope*, or higher cost, when searching for *fear*, will not be pruned

by cube pruning. Finally, running the Viterbi algorithm over the rescored forest, will then find an approximation of:

$$\underset{y,h}{\operatorname{argmax}} \sum_{e \in h} (\mathbf{w}^\top \phi(e) + w_\Delta \cdot \Delta(e)),$$

where w_Δ is a dummy weight that is set to 1 or -1, and is used to control whether we search for *hope* or *fear*. Running forest search produces an overhead in complexity compared to k-best search, as $\Delta(e)$ needs to be computed for each edge, and forest rescoring and Viterbi search have to be run twice to search for *hope* and *fear*.

Sampling For each input x, we need to sample a document pair (d^+, d^-). Our first strategy is to randomly sample a pair. We also implement a second weighted sampling strategy based on document similarity. For weighted sampling, our intuition is that a relevant document d^+ should be sampled with higher weight if it is more similar to the input document. Similarity between an input document and a target document is measured by an unsupervised cross-lingual document similarity metric. This metric is calculated by computing document representations from bilingual word embeddings. The embeddings are learned from the aligned parallel training corpus using the bilingual skipgram model of T. Luong et al. (2015).[6] Document representations are computed by averaging over all word representations in the document, weighted by the inverse document frequencies of the words. Cosine similarity is then used to measure similarity between the current source document and all relevant documents. We use weighted reservoir sampling by Efraimidis and Spirakis (2006) to sample a document weighted by its cosine similarity to the current source document. The contrast document d^- is drawn randomly from D^-, but is re-drawn if it has higher similarity to the input than d^+.

Parallelization and Feature Selection In order to be able to train on thousands of documents, we use the parallelization method described in Algorithm 4 (IterSelSGD) of Simianer et al. (2012). This method splits training data into shards, trains one epoch on each shard and then applies feature selection by ℓ_1/ℓ_2 regularization before starting the next epoch. While the method was originally applied with a pairwise ranking perceptron objective, it can be transferred to the online ramp loss training procedure (our Algorithm 4 on page 78). Additionally, we add a regularization term, which prevents weights to stray too far from the initial weight vector. This term is obtained by adding $C\frac{1}{2|X|}\|(w-w_0)^2\|$ to the ramp loss objective, where C is a hyperparameter controlling the amount of regularization.

[6] `github.com/lmthang/bivec`

Pre-training and weight freezing As previously mentioned, document-level links can be a much weaker source of supervision than parallel reference translations. In order to ensure that the translation model continues to produce fluent translations, we first pre-train it on out-of-domain parallel data and only use our in-domain data for fine tuning the model. In the hierarchical phrase-based paradigm, we accomplish this fine tuning effect by freezing the weights for the dense translation model features and only tuning weights for additional sparse features, which emphasize individual rule shapes, as well as single word and phrase pairings.

5.4.2 Experimental Setup

Our initial out-of-domain English-German translation system is trained on 2.1 million sentence pairs (61/59 million English/German tokens) from the Europarl v7 corpus[7] (Koehn, 2005), the News Commentary v10 corpus[8], and the MultiUN v1 corpus[9] (Eisele & Chen, 2010). Bidirectional word alignments are computed using MGIZA++[10] and symmetrized using the `grow-diag-final-and` heuristic (Koehn et al., 2003). A 4-gram count-based n-gram language model is estimated from the target side of the training data using KenLM (Heafield, Pouzyrevsky, Clark, & Koehn, 2013). All experiments use the hierarchical phrase-based decoder cdec (Dyer et al., 2010). Hierarchical phrase rules are extracted using cdec's implementation of the suffix array extractor by (Lopez, 2007) with default settings. Our baselines use 21 decoder features – 7 translation model features, 2 language model features, 7 pass through features, 3 arity penalty features, word penalty and glue rule count features –, which are implemented in cdec. Decoder feature weights are optimized on the WMT Newstest 2014 data set (3003 sentence pairs) using the pairwise ranking optimizer dtrain[11] (Simianer et al., 2012). We run dtrain for 15 epochs, with the hyperparameters k-best size=100, loss-margin=1, and a learning rate of $1e^{-5}$. The final weights are averaged over all epochs.

Our method is trained on 10,000 input sentences sampled from the English WikiCLIR documents. Each input sentence is annotated with a document identifier in order to sample positive and negative examples. The relevant document collection includes all German documents linked to an English document by a *mate* or *link* relation. For the contrast documents we use the News Commentary corpus, which we split into documents. In a pre-processing step, we extract n-grams up to order 3 from each document, which are

[7] `statmt.org/europarl/`

[8] `statmt.org/wmt15/training-parallel-nc-v10.tgz`

[9] `euromatrixplus.net/multi-un/`

[10] `cs.cmu.edu/~qing/giza/`

[11] The code is no longer being developed, but a legacy fork is available on `https://github.com/pks/cdec-dtrain-legacy`.

needed to calculate n-gram precision. Experiments with a larger training set of 200,000 input sentences have shown no significant improvement. All experiments use the baseline weights for the 21 dense translation model features, learning only weights for additional sparse features. We use sparse rule identifiers and rule source and target bigram features as described in (Simianer et al., 2012), as well as lexical alignment features as described in (Sokolov et al., 2013). All sparse feature functions are implemented in CDEC. Parallelized training is done on 10 shards, followed by an ℓ_1/ℓ_2 feature selection step which keeps at most 100,000 features. We use a constant learning rate of $\eta = 1e^{-4}$ and regularization strength $C = 1$. The ramp loss objective uses scaling factor $\alpha = 10$. Hyperparameters were selected in preliminary experiments on the development set. Experiments are run for up to 20 epochs, with early stopping being determined by performance on the in-domain development set (WIKI1).

5.4.3 Experimental Results

All results are reported on the test set (WIKI2). Significance was determined by running the `multeval` script[12] (Clark et al., 2011). The main dimensions for experimental comparison are the two ramp loss objectives RAMP$^-$ and RAMP$_{\delta_2}$, the different search variants (forest and k-best) and the different link strengths (*mate* and *link*). For the link relation we also compare random and weighted sampling strategies. We do not test weighted sampling for experiments with *mates* only, as there is usually only one mate document d^+ for each input document. In this setting, only irrelevant documents are sampled.

Table 5.3 reports BLEU scores for applying variants of our approach to the unadapted baseline model. Row 2 shows that for the SMT setup, searching only over the k-best list is too restricted with no significant improvement over the baseline. Row 3, on the other hand, which uses the same setting (RAMP$_{\delta_2}$ with mate documents) but with forest search, produces our best result with 0.6 points increase in BLEU. While RAMP$^-$ (row 4) still improves over the baseline, it falls short of RAMP$_{\delta_2}$. Training on weaker supervision from the link relation (rows 5-7), does not lead to the same increase in BLEU. Combining mates and links still improves significantly over the baseline (row 5). But using only links does not (row 6). This decrease in performance can be alleviated by using weighted sampling (row 7), which produces a significant increase by 0.3 BLEU points over the baseline.

Figure 5.3 shows learning curves over epochs on the in-domain heldout data for training on mates and links (both with random sampling). Both experiments show gradual

[12] `https://github.com/jhclark/multeval`

		search method	levels	sampling method	BLEU	Δ
1	baseline	-	-	-	12.46	
2	RAMP_{δ_2}	k-best	mates	random	12.57	+0.11
3	RAMP_{δ_2}	forest	mates	random	**13.05**	+0.59
4	RAMP^{-}	forest	mates	random	**12.81**	+0.34
5	RAMP_{δ_2}	forest	mates+links	random	**12.85**	+0.38
6	RAMP_{δ_2}	forest	links	random	12.67	+0.21
7	RAMP_{δ_2}	forest	links	weighted	**12.77**	+0.31

Table 5.3 Result with an out-of-domain baseline for different ramp loss variants, relevance levels, search methods and sampling methods. Boldfaced results are significantly better than the baseline at a significance level of 0.05.

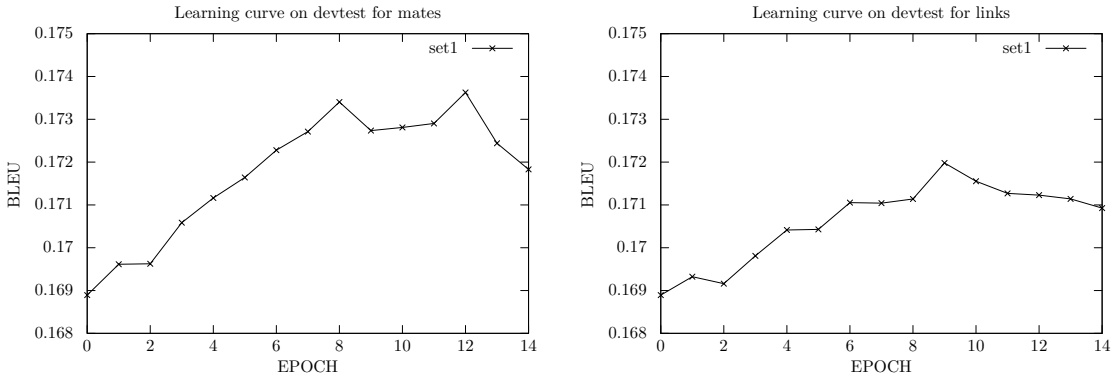


Figure 5.3 Performance on heldout set WIKI1 for mates and links.

improvements up to epoch 10-12 with declining results after that. The decline probably occurs due to overfitting.

Combining adaptation approaches As mentioned in the introduction of this chapter, our approach can be integrated with other methods for adaptation from non-parallel data. We first build an adapted language model by sampling 500,000 sentences from the German Wikipedia data, which we add to the out-of-domain language model data. Experiments using an additional Wikipedia language model are marked by `+wiki-lm`. We then adapt the translation model by adding pseudo-parallel Wikipedia data extracted by `yalign` (see Section 5.3) to our baseline training data and re-training the translation model. Experiments using an adapted translation model are marked as `+wiki-tm`. Both adapted models are then combined with our approach using the best configuration from the unadapted experiments above (row 3 in Table 5.3). Experiments using biploar ramp loss training

Experiment	BLEU	Δ
baseline `+wiki-lm`	13.62	
`+wiki-ramp`	**13.93**	+0.31
baseline `+wiki-lm +wiki-tm`	14.96	
`+wiki-ramp`	**15.17**	+0.21

Table 5.4 Results for combining different adaptation methods. * indicates a significant difference to the adapted baseline at a significance level of $p = 0.05$.

on Wikipedia are marked as `+wiki-ramp`. Table 5.4 shows results for all adaptation approaches. Both language model and translation model adaptation boost performance significantly. The good performance of TM adaptation leads to the conclusion that if there is a strong signal for potential parallelism like in the Wikipedia data, automatic pseudo-parallel data extraction works well. In both experiments, our method is able to produce an additional small, yet significant, gain over the adapted model, showing that additional information can be learned from the relevance signal.

To summarize, for SMT we test the ramp loss objectives RAMP_{δ_2} and RAMP^{-} to train sparse lexicalized features for adapting an out-of-domain model to a new domain without supervision from parallel references. Forest search, an approximate loss-augmented inference over the entire hypothesis space, is proposed in order to find more diverse hope and fear hypotheses. Parallelized training is applied to scale training to larger data sets, especially considering the added complexity of forest search. This training setup also introduces regularization via ℓ_1/ℓ_2 feature selection. Furthermore, weighted sampling by document similarity is introduced to inform supervision from weaker links. Experimentally, we find significant improvements over an unadapted baseline when using forest search and the stronger mate relation. Both ramp loss objectives produce significant gains, with RAMP_{δ_2} slightly outperforming RAMP^{-}. With weighted sampling, significant gains have also been produce by using only the weaker document links. Our best approach still leads to small improvements on top of language model or translation model adaptation.

5.5 Exploring the Impact of Negative Supervision for Weakly Supervised NMT Adaptation

We now study the performance of the structured ramp loss for neural machine translation (NMT) in the same weak supervision scenario as above (see Section 5.3). NMT systems are normally trained by a token-level maximum likelihood estimation (MLE) objective. But the in-domain supervision available in our scenario is much weaker and cannot be broken down to the token level. We therefore again employ the structured ramp loss to incorporate sequence-level feedback. We also again take advantage of the structured ramp loss's property of deliberately selecting a good output to promote and a bad output to demote, rather than only promoting a good output without differentiating between the other outputs or computing an expectation over all outputs. We refer to this property of the structured ramp loss as "*bipolar*". We take this bipolar principle one step further by incorporating positive and negative supervision when selecting y^+ and y^-. The NMT experiments conducted here focus on evaluating the contribution of weak positive and negative supervision via a bipolar objective. We therefore evaluate not only the neural equivalent of RAMP$^-$ and RAMP$_{\delta_2}$, but other versions of the structured ramp loss. We also compare the performance of structured ramp loss to minimum risk training.

5.5.1 Objectives

For NMT, the score $s_w(y, x)$ for an output y is defined as the likelihood of y, $p_w(y|x)$. This likelihood is the product of the token-level probabilities at each step j as emitted by a recurrent encoder-decoder model with attention (Bahdanau et al., 2015) under parameters w:

$$s_w(y, x) = p_w(y \mid x) = \prod_{j=1}^{J} p_w(y_j \mid y_{<j}, x)\,,$$

where $y_{<j}$ refers to the first $j-1$ tokens in y.

Maximum likelihood. With full supervision from reference translations y^*, the model is trained using the following maximum likelihood objective

$$L_{MLE} = -\frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{J} \log\, p_w(y^*_{m,j} \mid y^*_{m,<j}, x_m)\,. \tag{5.10}$$

This objective computes an average over minibatches of M inputs. $y^*_{m,<j}$ refers to the first $j-1$ tokens in the reference. In our weakly supervised scenario there are no reference

translations available, precluding us from applying MLE training. Instead, we turn to sequence-level objectives which incorporate weak supervision via an external sentence-level metric.

Structured ramp loss. The ramp losses used in the NMT experiments are again exemplifications of the general ramp loss objective in Equation 5.1. Table 5.5 shows the different definitions for y^+ and y^-. RAMP$^-$ and RAMP$_{\delta_2}$ are defined analogously to the SMT setup, with the same metrics δ_1 and δ_2. We again use the the *cost* functions $\Delta_1(y, d^+)$ and $\Delta_2(y, d^+, d^-)$ as shorthand for $\alpha(1-\delta_1(y, d^+))$ and $\alpha(1-\delta_1(y, d^+, d^-))$, respectively. As the main interest of the NMT experiments lies in evaluating the contribution of the negative supervision from d^-, three additional ramp loss versions are explored experimentally. First, RAMP is the classical ramp loss objective as defined for structured prediction by Gimpel and Smith (2012) and Chiang (2012), obtained by simply replacing the BLEU metric with $\delta_1(y, d^+)$. RAMP2 corresponds to Gimpel and Smith (2012)'s equation 7, where the model-best hypothesis $\hat{y}$ is used as y^-. Both RAMP and RAMP2 only use positive in-domain supervision from d^+. RAMP1$^-$ is an adapted version of Gimpel and Smith (2012)'s equation 6, where the model-best hypothesis $\hat{y}$ is used as y^+ and updated against a fear hypothesis. In analogy to RAMP$^-$, we use the hope hypothesis with respect to d^- as y^- in our version of this objective. RAMP1$^-$ only uses in-domain supervision from d^-.

In addition to the sequence-level ramp loss objectives, we also test a token-level ramp loss objective, which only assigns rewards to tokens that differ between y^+ and y^-

$$\begin{aligned} L_{RAMP\text{-}T} = \frac{1}{M}\sum_{m=1}^{M} & -\sum_{j=1}^{J} \tau^+_{m,j} \log p_w(y^+_{m,j} \mid y^+_{m,<j}, x_m) \\ & + \sum_{j=1}^{J} \tau^-_{m,j} \log p_w(y^-_{m,j} \mid y^-_{m,<j}, x_m)\,, \end{aligned} \tag{5.11}$$

where

$$\tau^+_{m,j} = \begin{cases} 0 & \text{if } y^+_m \in y^-_m \\ 1 & \text{otherwise} \end{cases}$$

and

$$\tau^-_{m,j} = \begin{cases} 0 & \text{if } y^-_m \in y^+_m \\ -1 & \text{otherwise}\,. \end{cases}$$

This objective was proposed by the co-author of Jehl et al. (2018). The contribution of this thesis lies in its application to the weakly supervised MT scenario.

Loss	y^+	y^-
RAMP	$\text{argmax}_y\, s_w(y,x) - \Delta_1(y,d^+)$	$\text{argmax}_y\, s_w(y,x) + \Delta_1(y,d^+)$
RAMP$^-$	$\text{argmax}_y\, s_w(y,x) - \Delta_1(y,d^+)$	$\text{argmax}_y\, s_w(y,x) - \Delta_1(y,d^-)$
RAMP1$^-$	$\hat{y}$	$\text{argmax}_y\, s_w(y,x) - \Delta_1(y,d^-)$
RAMP2	$\text{argmax}_y\, s_w(y,x) - \Delta_1(y,d^+)$	$\hat{y}$
RAMP$_{\delta_2}$	$\text{argmax}_y\, s_w(y,x) - \Delta_2(y,d^+,d^-)$	$\text{argmax}_y\, s_w(y,x) + \Delta_2(y,d^+,d^-)$

Table 5.5 Configurations for y^+ and y^- for weakly supervised MT adaptation. $\hat{y}$ is the highest-probability model output. The argmax_y is taken over the k-best list $\mathcal{K}(x)$, which we omit due to space constraints.

Minimum risk training. For sequence-level, metric-augmented training of NMT systems, the most successful method to date has been minimum risk training (MRT) (Edunov et al., 2018; S. Shen et al., 2016). We therefore compare our ramp loss objectives with MRT. The MRT objective minimizes the expected risk of the output distribution, which is approximated by sampling from the output space. We use the following objective:

$$L_{MRT} = \sum_{s=1}^{S} p_w(y_s|x) \cdot (\delta(y_s) - b(x))\,, \tag{5.12}$$

where S is the number of samples drawn to approximate the output space and $\delta(y_s)$ is the reward for output y_s. Further, a baseline $b(x)$ average reward computed over S' sampled outputs can be subtracted from the model distribution for more stability in training, with

$$b(x) = \frac{1}{S'} \sum_{s'=1}^{S'} \delta(y_{s'})\,.$$

We test two variants of MRT. MRT$_{\delta_1}$ uses the metric $\delta_1(y,d^+)$ as reward function. This objective only uses supervision from d^+. MRT$_{\delta_2}$ uses $\delta_2(y,d^+,d^-)$ as reward function. The metric allows us to include positive and negative supervision in the MRT objective.

The objectives are implemented in the neural sequence-to-sequence toolkit Nematus (Sennrich et al., 2017). This toolkit implements the recurrent neural encoder-decoder architecture with attention, as well as the MLE and MRT objectives. We extend the toolkit by the ramp loss objectives and the metrics δ_1 and δ_2. We approximate the search for y^+ and y^- over the output space $\mathcal{Y}(x)$ by a k-best list $\mathcal{K}(x)$.

5.5.2 Experimental Setup

The baseline out-of-domain model is trained on the same data as the SMT baseline, using 2.1M sentences from Europarl, News Commentary, and the MultiUN corpus (see Section 5.4). Training progress is measured on 7,000 heldout sentences that are randomly sampled from the training data. All input data are split into subword units using byte-pair encodings (Sennrich et al., 2016c) with 30,000 merge operations. Sentences longer than 80 words are filtered from the training data. The baseline model uses 500-dimensional word embeddings and hidden layer dimension of 1,024. Encoder and decoder use GRU units (Cho et al., 2014). The baseline is trained using the MLE objective and AdaDelta (Zeiler, 2012) using minibatch size $M = 64$. Dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) is applied for regularization with a dropout rate of 0.2 for embedding and hidden layers and 0.1 for source and target layers. The baseline is validated every 2,500 updates. As training did not reach the stopping criterion for a long time, but BLEU scores on heldout also had not increased for a long time either, training was stopped manually. We use the baseline model at iteration 650,000 (after 20 epochs) as an input to our tuning method.

Our adaptation experiments use the same training and evaluation data as Section 5.4. Hyperparameters are selected on the development set. For structured ramp loss, we use SGD with a learning rate of 0.005 and no dropout. We use minibatch size $M = 40$, α is set to 10. The k-best list size is set to 16. Results are validated every 200 updates. For MRT, we use SGD with a learning rate of 0.05 and no dropout. We use sampling size $S = 16$ and compute the baseline over 10 samples. As all samples have to be considered in the update and thus loaded into GPU RAM, an update is made after each input for MRT training. Progress on the development set is evaluated every 8,000 updates to match the validation frequency of the structured ramp loss. The stopping point is determined by the BLEU score on the development set. We report scores computed with Moses' `multi-bleu.perl` on tokenized, truecased output. Each experiment is run two times and results are averaged over the runs.

5.5.3 Experimental Results

Table 5.6 shows results for the different ramp loss versions. The losses RAMP, RAMP1$^-$ and RAMP2 in rows 2, 3 and 4 perform worse than the baseline. These losses do not incorporate both positive and negative supervision, but use supervision either only from d^+ (RAMP and RAMP2) or only from d^- (RAMP1$^-$). RAMP performed worst, an observation which we explain by the extremely weak signal from relevant documents.

		M	% BLEU	Δ
1	baseline	64	15.59	
2	RAMP	40	15.03 $\pm$0.01	$-$ 0.56
3	RAMP1$^-$	40	15.12 $\pm$0.02	$-$ 0.47
4	RAMP2	40	15.19 $\pm$0.01	$-$ 0.40
5	MRT$_{\delta_1}$	1	15.37 $\pm$0.04	$-$ 0.22
6	MRT$_{\delta_2}$	1	15.70 $\pm$0.04	+ 0.11
7	RAMP$^-$	40	**15.85** $\pm$0.02	+ 0.26
8	RAMP$_{\delta_2}$	40	**15.86** $\pm$0.04	+ 0.27
9	RAMP$^-$-T	40	**16.03***$\pm$0.02	+ 0.44
10	RAMP$_{\delta_2}$-T	40	**15.84** $\pm$0.02	+ 0.25

Table 5.6 BLEU scores for weakly supervised MT experiments. Boldfaced results are significantly better than the baseline at $p < 0.05$. The * (line 9) marks a significant difference over the corresponding RAMP$^-$ objective (line 7).

Trying to perfectly re-create d^+ will not lead to good translations. Looking at the MRT experiments (rows 5 and 6), MRT$_{\delta_1}$ also causes a deterioration in BLEU compared to the baseline. This loss function also computes the reward only according to d^+. MRT$_{\delta_2}$, on the other hand, shows a nominal improvement over the baseline. This objective incorporates bipolar supervision by using δ_2 as its metric. RAMP$^-$ and RAMP$_{\delta_2}$ also incorporate bipolar supervision. Both of these objectives are able to significantly outperform the baseline. This result shows that bipolar feedback is crucial for success when learning from weak supervision. Further, the ramp loss is superior to MRT at learning from bipolar supervision, because, unlike MRT, which uses random samples, the ramp loss intelligently selects specific positive and negative outputs to encourage or discourage. Both objectives perform at the same level, showing that in the setup investigated here it does not matter how the bipolar supervision is integrated. Finally, RAMP$^-$-T, the token-level version of RAMP$^-$, adds another significant improvement. This result shows that token-level updates are helpful in this weak supervision scenario.

To summarize, only objectives which take into account bipolar feedback are able to improve over the baseline, while objectives which fail to do so have a deteriorating effect on translation quality. This observation holds both for ramp loss and minimum risk training. For ramp loss, it is irrelevant whether the positive and negative supervision is incorporated through the metric (RAMP$_{\delta_2}$) or the objective (RAMP$^-$). Using a bipolar ramp loss objective outperforms minimum risk training. The best result is achieved using the

Loss	y^+	y^-
RAMP	$\text{argmax}_y\, s_w(y,x) - \Delta(y,y^*)$	$\text{argmax}_y\, s_w(y,x) + \Delta(y,y^*)$
RAMP1	$\hat{y}$	$\text{argmax}_y\, s_w(y,x) + \Delta(y,y^*)$
RAMP2	$\text{argmax}_y\, s_w(y,x) - \Delta(y,y^*)$	$\hat{y}$
PERC1	y^*	$\hat{y}$
PERC2	$\text{argmax}_y\, \text{BLEU}_{+1}(y,y^*)$	$\hat{y}$

Table 5.7 Configurations for y^+ and y^- for fully supervised MT. $\hat{y}$ is the highest-probability model output, $y*$ is a gold standard reference. The argmax_y is taken over the k-best list $\mathcal{K}(x)$, which we omit due to space constraints.

token-level version of RAMP$^-$. While the overall improvements are small, an evaluation of the same objectives on a weakly supervised semantic parsing task has yielded a similar ranking of objectives with larger gains, showing the general validity of the bipolar principle for weakly supervised scenarios. The semantic parsing experiments are described in Jehl et al. (2018). We surmise that the larger improvements in semantic parsing are possible, because the available weak supervision - gold answers for natural language questions - is stronger than in our case.

5.6 Comparing Structured Ramp Loss to Other Objectives for Fully Supervised NMT

While our work focuses on weakly supervised tasks, we are also the first to investigate structured ramp loss for NMT. We therefore also conduct experiments using a fully supervised MT task. These experiments are motivated on the one hand by lifting the findings of Gimpel and Smith (2012) to the neural MT paradigm, and on the other hand to expand the work by Edunov et al. (2018) on applying classical structured prediction losses to neural MT.

5.6.1 Objectives

For fully supervised MT, we assume access to one or more reference translations y^* for each input x. The reward $\text{BLEU}_{+1}(y,y^*)$ is a per-sentence approximation of the BLEU score.[13] Table 5.7 shows the different losses for y^+ and y^-. RAMP, RAMP1, and RAMP2

[13]We use the BLEU score with add-1 smoothing for $n > 1$ as proposed by B. Chen and Cherry (2014).

are defined analogously to the ramp loss variants proposed by Gimpel and Smith (2012). RAMP is a bipolar objective, which encourages a *hope* output, which has a high probability according to the model and high BLEU_{+1}, while discouraging a *fear* output, which has high probability according to the model and low BLEU_{+1}. This objective is also used by Chiang (2012). RAMP1 replaces the hope output with the model-best translation $\hat{y}$, while RAMP2 replaces the fear with $\hat{y}$. Like in the weakly supervised setting, the external metric is again subtracted from 1 and scaled by a hyperparameter $\alpha > 0$ to obtain a *cost* function $\Delta(y, y^*) = \alpha(1 - \text{BLEU}_{+1}(y, y^*))$.

Gimpel and Smith (2012) also include the perceptron loss in their analysis. PERC1 is a re-formulation of the Collins perceptron (Collins, 2002) where the reference is used as y^+ and $\hat{y}$ is used as y^-. A comparison with PERC1 is not possible for the weakly supervised tasks in the previous sections, as gold structures are not available for these tasks. With neural MT and subword methods we are able to compute this loss for any reference without running into the problem of *reachability* that was faced by phrase-based MT (Liang et al., 2006). However, using sequence-level training towards a reference can lead to degenerate solutions where the model gives low probability to all its predictions (S. Shen et al., 2016). PERC2 addresses this problem by replacing y^* by a surrogate translation which achieves the highest BLEU_{+1} score in $\mathcal{K}(x)$. This approach is also used by Edunov et al. (2018) for the loss functions which require an oracle. PERC1 corresponds to Equation 9, PERC2 to Equation 10 of (Gimpel & Smith, 2012).

5.6.2 Experimental Setup

We conduct experiments on the IWSLT 2014 German-English task, which is based on Cettolo et al. (2012) in the same way as Edunov et al. (2018). The training set contains 160K sentence pairs. We set the maximum sentence length to 50 and use BPE with 14,000 merge operations. Edunov et al. (2018) sample 7K sentences from the training set as heldout data. We do the same, but only use 1/10th of the data as heldout set to be able to validate often without having to wait long. Our baseline system is a BiLSTM encoder-decoder with attention. Word embedding and hidden layer dimensions are set to 256. We use batches of 64 sentences for baseline training and batches of 40 inputs for training RAMP and PERC variants. MRT makes an update after each input using all sampled outputs and resulting in a batch size of 1. All experiments use dropout for regularization, with dropout probability set to 0.2 for embedding and hidden layers and to 0.1 for source and target layers. During MLE-training, the model is validated every 2500 updates and training is stopped if the MLE loss on the heldout set worsens for 10 consecutive validations.

For metric-augmented training, we use SGD for optimization with learning rates optimized on the development set. Ramp losses and PERC2 use a k-best list of size 16. For ramp loss training, we set $\alpha = 10$. RAMP and PERC variants both use a learning rate of 0.001. A new k-best list is generated for each input using the current model parameters. We compare ramp loss to MRT as described above. For MRT, we use SGD with a learning rate of 0.01 and set $S = 16$ and $S' = 10$. As Edunov et al. (2018) observe beam search to work better than sampling for MRT, we also run an experiment in this configuration, but find no difference between results. As beam search runs significantly slower, we only report the experiments which use sampling. The model is validated on the development set after every 200 updates for experiments with batch size 40 and after 8,000 updates for MRT experiments with batch size 1. The stopping point is determined by the BLEU score on the heldout set after 25 validations. As we are training on the same data as the baseline, we also apply dropout during ramp loss training to prevent overfitting. BLEU scores are computed with Moses' `multi-bleu.perl` on tokenized, truecased output. Each experiment is run 3 times and results are averaged over the runs.

5.6.3 Experimental Results

Table 5.8 shows experimental results for the different training losses. All experiments, except for PERC1 (row 3), yield improvements over the baseline, confirming that sequence-level losses which update towards the reference can lead to degenerate solutions. For MRT (row 2), our findings show similar performance to the initial experiments reported by Edunov et al. (2018) who gain 0.24 BLEU points on the same test set.[14] PERC2 (row 4) and RAMP2 (row 6) improve over the baseline and PERC1, but perform on a par with MRT and each other. Both RAMP (row 7) and RAMP1 (row 5) are able to outperform MRT, PERC2 and RAMP2, with the bipolar objective RAMP also outperforming RAMP1 by a narrow margin. The main difference between RAMP and RAMP1, compared to PERC2 and RAMP2, is the fact that the latter objectives use $\hat{y}$ as y^-, while the former use a *fear* translation with high probability and low BLEU_{+1}. We surmise that for this fully supervised task, selecting a y^- which has some known negative characteristics is more important for success than finding a good y^+. RAMP, which fulfills both criteria, still outperforms RAMP2. This result re-confirms the superiority of bipolar objectives compared to non-bipolar ones. While still improving over the baseline, token-level ramp loss RAMP-T (row 8) is outperformed by RAMP by a small margin. This result suggests that when employing a metric-augmented objective on top of an MLE-trained model in a full supervision scenario without domain shift, there is little room for improvement from

[14] See their Table 2. Using interpolation with the MLE objective, Edunov et al. (2018) achieve +0.7 BLEU points, but as we are only interested in the effect of the sequence-level loss functions, we look at these losses in isolation.

		M	% BLEU	Δ
1	baseline	64	31.99	
2	MRT	1	**32.17** $\pm$ 0.02	+ 0.18
3	PERC1	40	31.91 $\pm$ 0.02	− 0.08
4	PERC2	40	**32.22** $\pm$ 0.03	+ 0.23
5	RAMP1	40	**32.36*** $\pm$ 0.05	+ 0.37
6	RAMP2	40	**32.19** $\pm$ 0.01	+ 0.20
7	RAMP	40	**32.44**** $\pm$ 0.00	+ 0.45
8	RAMP-T	40	**32.33*** $\pm$ 0.00	+ 0.34

Table 5.8 BLEU scores for fully supervised MT experiments. Boldfaced results are significantly better than the baseline at $p < 0.01$ according to `multeval` (Clark et al., 2011). * marks a significant difference to MRT and PERC2, and ** marks a significant difference to RAMP1.

token-level supervision, while gains can still be obtained from additional sequence-level information captured by the external metric, such as information about the sequence length.

To summarize, our findings on a fully supervised task show the same small margin for improvement as Edunov et al. (2018), without any further tuning of performance, e.g. by interpolation with the MLE objective. Bipolar RAMP is found to outperform the other losses. This observation is also consistent with the results by Gimpel and Smith (2012) for phrase-based MT. We conclude that for fully supervised MT, deliberately selecting a *hope* and *fear* translation is beneficial.

Conclusions

This chapter has examined the potential of structured ramp loss objectives for learning from weak and negative supervision. Ramp loss has been selected because of its potential to deliberately select positive and negative outputs to encourage and discourage, rather than averaging over a sample of outputs or promoting a reference structure, which may not be available in the weakly supervised case. We have called this characteristic "bipolar". The bipolar nature of the ramp loss also allows to incorporate both positive and negative supervision. In our scenario of adapting a translation system to a new domain using only supervision from cross-lingual document-level links, we have paired positive supervision

from relevant documents with negative supervision from irrelevant documents to obtain a more reliable signal. We proposed adapted bipolar ramp loss, which allow the model to take into account both sources of supervision. Negative supervision is integrated into the objective either directly, by modifying the definition of y^-, or indirectly, via the new δ_2 metric.

We have defined a new Wikipedia task, including a small development and test set. A small manual evaluation shows that while the stronger cross-lingual *mate* relation between articles can also be used for pseudo-parallel data extraction, the weaker *link* relation does not produce any parallel data.

Experiments using an SMT system have focused on learning sparse lexicalized features from links of different strength. Using forest search instead of a k-best approximation was vital to the experimental success, leading to significant gains over the baseline when using bipolar ramp loss with stronger cross-lingual links. As expected, learning was found to be slower and less successful when using supervision only from weak links, but by employing a weighted sampling scheme based on unsupervised document similarity, significant gains could be obtained. Finally, we showed that small gains still persisted when our approach was combined with traditional adaptation methods.

For NMT, experiments focused on exploring the significance of the bipolar principle and the contribution of positive and negative supervision. Five different ramp loss versions were tested on the weakly supervised task. Structured ramp loss was further compared to two variants of minimum risk training, one of which incorporated positive and negative supervision by using the δ_2 metric as a reward function. Results showed that positive and negative supervision was crucial to the experimental success, both for minimum risk training and ramp loss. The best result was obtained by using a token-level version of bipolar ramp loss with the δ_1 metric.

The importance of the bipolar principle was further confirmed in a fully supervised NMT experiment, which lifted the structured prediction losses suggested by Gimpel and Smith (2012) to NMT and featured the first use of structured ramp loss for NMT.

CHAPTER 6

Document Metadata as Side Constraints for Fully Supervised Translation

So far, we have considered document meta-information as a source of supervision for tasks where full supervision is not available. But could document meta-information also be useful as an additional supervision signal on top of full supervision? This chapter proposes to integrate document meta-information into a fully supervised machine translation (MT) system as *side constraints*. We use the term "side constraints" to mean information that is not present in a source string, but can influence translation choice in the target string. This follows the definition of the term by Sennrich, Haddow, and Birch (2016a) who explore politeness as a side constraint for translation.

Patent translation lends itself particularly well to our endeavor since patent documents are annotated with rich metadata containing different types of information, from hierarchical categorization to information about individual inventors. We are interested in seeing whether patent translation can be improved by leveraging this metadata and if so, which kind of metadata and which model integration is most useful. For example, patent documents are categorized according to their topic. Depending on the overall topic, different translation choices can be required, and the correct choice will not always be apparent from the sentence context. By providing the categorization information to the translation model as a side constraint, we enable the model to select the correct translation, given the constraint.

First, we define side constraints for a statistical machine translation (SMT) system. We use the metadata to inform phrase selection by adding phrase-level side constraint match features to the SMT system. These features reflect the amount of matching metadata between an input document and the training documents that a phrase pair was extracted from. In order to combat sparsity, different feature aggregation methods are introduced. Further, a TFIDF-based weighting scheme is defined to better distinguish the importance of different metadata. We evaluate the features along with the aggregation and weighting schemes and perform an ablation for different feature categories on Japanese-English and German-English patent translation tasks. Second, we train a neural machine translation (NMT) system for both patent translation tasks, as NMT has been shown to surpass SMT on many translation tasks. We present two approaches to source-side integration of side constraints for neural patent translation, as word-attached and sentence-attached side constraints. We adapt these approaches to accommodate multi-category, multi-valued metadata. We then again perform an experimental evaluation of the different approaches and different categories.

The main contributions presented in this chapter are: (1) A method for an SMT system to capture matching document metadata between a test document and phrase pairs in the translation model via side constraint match features. (2) Adaptation of two methods by Kobus, Crego, and Senellart (2017) for passing document metadata to an NMT system via sentence-attached and word-attached side constraints. (3) An in-depth evaluation of all proposed methods on patent translation for two language pairs (Japanese-English and German-English), along with a comparison of SMT and NMT performance on the task. All contributions were made by the author after consultation with the supervisor of this thesis. An earlier version of this work was published as Jehl and Riezler (2018). Note that this chapter includes a different and much more comprehensive evaluation of the SMT features, leading to slightly divergent results compared to the publication, where no improvements over the baseline were found. The more comprehensive experiments reported here show indeed modest improvements for Japanese-English patent translation.

The remainder of the chapter is organized as follows: Related work is reviewed in Section 6.1. Section 6.2 contains the general problem and task definitions. Section 6.3 describes the approach to integrating metadata as side constraints into an SMT system and presents experimental results. Section 6.4 describes two approaches to integrating metadata as side constraints into an NMT system and presents experimental results.

6.1 Related Work

For SMT, Niehues and Waibel (2010) and Bisazza, Ruiz, and Federico (2011) both modify the phrase table to include corpus identifiers or binary in-domain/out-of-domain markers, which they find beneficial for domain adaptation. However, they do not use more fine-grained information. For phrase-based patent translation, Wäschle and Riezler (2012b) use patent section labels to partition training data for multi-task learning, but do not look into the more fine-grained classification information. Khadivi, Wilken, Dahlmann, and Matusov (2017) investigate integrating category information into a hybrid SMT/NMT system using sparse features. Their approach resembles our SMT features, but they only use six product categories, whereas we have thousands of different annotations, requiring aggregation and weighting to make features more robust.

Our work on NMT is inspired by the work of Kobus et al. (2017) on multi-domain adaptation. This work uses the domain label as a side constraint for translation in a multi-domain setup: A combined NMT model is trained on subcorpora from different domains, and each training sentence is marked with its subcorpus information. Test data come from one of the known domains and are marked in the same way. We take the idea of using sentence-attached and word-attached source side features to represent side constraints from this paper and modify it to fit our scenario of multi-category, multi-valued patent annotations. Kobus et al. (2017) observed no improvements for sentence-attached features, but saw an improvement of 0.8% BLEU for word-attached features when testing on all but the largest subdomain.

Incorporating side constraints via *sentence-attached* features has also been applied in other work: Originally, this method was proposed by Sennrich et al. (2016a) to model politeness as a side constraint. Johnson et al. (2016) have used it to indicate the desired target language for multilingual NMT. Chu et al. (2017) apply it to neural domain adaptation in combination with fine tuning methods (M.-T. Luong & Manning, 2015). Passing additional information to a neural network via *word-attached* features was first introduced by Collobert et al. (2011) as a way to add linguistic annotation for various NLP tasks using feed-forward and convolutional networks. Sennrich and Haddow (2016) transferred this idea to neural translation models. The word-attached features used by Kobus et al. (2017) were first presented by Crego et al. (2016), where they were used to encode case information.

The idea of leveraging document meta-information as a side constraint for translation was recently investigated by W. Chen, Matusov, Khadivi, and Peter (2016). They focus on integrating product category information for translation of product descriptions in

e-commerce, and also apply their method to online lecture translation (Cettolo et al., 2012), where the lectures are annotated with topic keywords. They also experimented with attaching document information as an artificial token to the source sentence, but found no gains for this method. They then propose to integrate topic information on the target side by including it as an additional read-out layer in the decoder before the softmax layer. This method improved e-commerce data translation by 1.4% BLEU and lecture translation by 0.3% BLEU.

6.2 Task

6.2.1 General Task Definition

In our scenario, weak supervision from document meta-information is available alongside strong supervision from parallel data. Given a source input $x = x_1, x_2, \ldots, x_{|x|}$, we assume that we can obtain a set of side constraints $S(x)$ from metadata of the source document d containing x. Our goal is to learn a parameterized function $s_w(y, x, S(x))$, which allows us to run inference as follows:

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, s_{[w, w_S]}(y, x, S(x)), \tag{6.1}$$

where $[w, w_S]$ indicates that using side constraints may require additional parameters related to the document information. In a log-linear SMT model, we define side constraint match features, which require additional weights. In the case of NMT, the necessity of adding parameters depends on the approach to integrating side constraints, as will be discussed in Section 6.4.

6.2.2 Side Constraints from Patent Metadata

Figure 6.1 shows the first page of an example patent. This page contains the patent's abstract and metadata. There are different metadata categories, as marked by the different item numbers in parentheses. Metadata include the application and disclosure dates (items 22 and 43), the classification according to the hierarchical International Patent Classification system[1] (item 51), the company filing the patent (item 71) and the inventors (item 72). In our example, the inventors are "*Ralf Schuler*" and "*Uwe Tellermann*", the company filing the patent is "*Robert Bosch GmbH*", and the IPC classifications are "*B60K*

[1] see `wipo.int/classifications/ipc/en/`, Sep 01, 2018.

(19) Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 10 2006 051 931 A1** 2008.06.05

(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2006 051 931.0**
(22) Anmeldetag: **03.11.2006**
(43) Offenlegungstag: **05.06.2008**

(51) Int Cl.[8]: ***B60K 6/387*** (2007.10)
B60K 6/48 (2007.10)

(71) Anmelder:
Robert Bosch GmbH, 70469 Stuttgart, DE

(72) Erfinder:
Schuler, Ralf, 73728 Esslingen, DE; Tellermann, Uwe, 70499 Stuttgart, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Antriebsstrang für Hybridfahrzeug**

(57) Zusammenfassung: Die Erfindung betrifft einen Antriebsstrang für ein Hybridfahrzeug zur mechanischen Kopplung eines Verbrennungsmotors mit einer elektrischen Maschine, gekennzeichnet durch eine Kupplung, die den Verbrennungsmotor auswählbar vollständig mit der elektrischen Maschine selektiv verbindet oder von dieser vollständig trennt. Ferner betrifft die Erfindung einen Antrieb für Hybridfahrzeuge mit einem Verbrennungsmotor, einer elektrischen Maschine und einem erfindungsgemäßen Antriebsstrang, wobei der Verbrennungsmototr selbststartfähig ist.

Figure 6.1 The title page of a patent document from the German Patent and Trade Mark Office (DPMA). The pre-amble contains information about the patent's date of application, the inventors and filing company and the IPC classification.

B	Section	Performing Operations, Transporting
B60	Class	Vehicles in General
B60K	Subclass	Arrangement or Mounting of Propulsion Units in Vehicles (…)

Table 6.1 IPC classification example.

6/387" and "*B60K 6/48*". Note that there are multiple classifications and inventors. The classification number encodes a hierarchical classification, whose levels can be obtained by truncating the string to a fixed number of positions. The first letter ("*B*") indicates the patent section, the first three letters ("*B60*") indicate the patent class and the first four letters ("*B60K*") indicate the patent subclass. The hierarchy is illustrated in Table 6.1. The digits in subsequent positions indicate an even more fine-grained classification.

Our experiments use up to five types of side constraints extracted from patent metadata. The first three types of constraints concern the patent classification. Patents are classified by their subject matter as belonging to one or more of 8 different sections (A-H). The sections are divided into 130 classes and 639 subclasses. The hierarchy branches out further, but we only use at most the top three hierarchy levels to avoid running into sparsity issues. We refer to these constraints as `IPC1` (patent section), `IPC2` (patent class) and `IPC3` (patent subclass). The fourth and fifth type of constraint considered in this work are the filing company, abbreviated as `COMP`, and the inventors, that we abbreviate by `INV`. Since each patent can be assigned to more than one section, class and subclass, and be filed by more than one company or inventor, side constraints need to be integrated into a translation in a way which allows multiple values for each constraint.

6.2.3 Data

Experiments are run on Japanese-English and German-English patent translation task. We train Japanese-English models on the NTCIR-7 Patent Translation training set[2] (Utiyama & Isahara, 2007). We use the NTCIR-8 development and test sets for development, and the NTCIR-9 intrinsic evaluation set for testing. Table 6.2 contains information about the data sets. Japanese data is segmented using MeCab[3]. For German-English, we use the abstracts section of the PatTR corpus (Wäschle & Riezler, 2012a). Patents published before 2008 are used for training. Development, devtest and test set contain about 2,000 sentences each and are randomly selected from the remaining patents. Sentences from the same document are always assigned to the same set. We apply a length-ratio based filter to the test set prior to evaluation to filter out noise from automatic sentence alignment, which was used to create the PatTR corpus (see Wäschle and Riezler (2012a)). Table 6.3 contains information about the data sets. All English data is tokenized and true-cased using the Moses toolkit[4].

6.3 Side Constraints for SMT

In phrase-based translation, we want to use side constraints from document metadata to influence phrase selection. Given an input document with metadata, the model should prefer translations that have been extracted from patents with similar metadata. For example, the polysemous German word "*Speicher*" translates into "*memory*", "*storage*" or

[2] `research.nii.ac.jp/ntcir/permission/ntcir-7/perm-en-PATMT.html`
[3] `taku910.github.io/mecab/`
[4] `github.com/moses-smt/mosesdecoder`

Data set	Origin	# sentence pairs	# documents
train	NTCIR-7 train	1,798,571	51,040
dev	NTCIR-8 pat-dev-2006-2007	2,000	115
devtest	NTCIR-8 Test Intrinsic	1,251	114
test	NTCIR-9 Test Intrinsic	2,001	417

Table 6.2 Data sets used in Japanese-English translation.

Data set	Origin	# sentence pairs	# documents
train	PatTR abstracts before 2008	694,609	280,009
dev	PatTR abstracts since 2008	2,000	899
devtest	PatTR abstracts since 2008	2,001	864
test (filtered)	PatTR abstracts since 2008	1,716	724

Table 6.3 Data sets used in German-English translation.

"*reservoir*".[5] "*memory*", but not "*reservoir*", occurred in patent classes related to computing, which belong to IPC section G, physics. On the contrary, "*reservoir*", but not "*memory*", occurred in patent class F24, Heating; Ranges; Ventilating. In this case, patent section and class provide clues for selecting the correct translation. When translating a new patent document categorized as F24, "*reservoir*" should therefore be preferred over "*memory*". The presented *side constraint match features* quantify to what degree the annotations of the current input document match the annotations of the documents that a phrase pair was extracted from. These features allow the model to give more weight to phrases with matching annotations. We first specify our approach to these features, before we present an experimental evaluation.

6.3.1 Side Constraint Match Features

We propose to capture side constraints via *side constraint match features* defined between side constraints of a phrase pair $\bar{x}, \bar{y}$ and an of an input x. With these additional features, the translation model score is computed as

$$s_w(x,y) = \max_h \sum_{\{\bar{x},\bar{y} \in h\}} \mathbf{w}^\top \phi(\bar{x},\bar{y}) + \mathbf{w}_{match}^\top \phi_{match}(S(\bar{x},\bar{y}), S(x)) \,,$$

[5] These examples were found in the phrase table and do not represent all meanings of "*Speicher*".

Algorithm 5 Extraction of raw overlap features

1: *At training time:*
2: **for all** phrase pairs $\bar{x}, \bar{y}$ **do**
3: $S(\bar{x}, \bar{y}) \leftarrow$ UNION(LOOKUP(x)) for all training pairs x, y containing $\bar{x}, \bar{y}$
4: **end for**
5:
6: *At test time:*
7: Obtain test input x
8: $S(x) \leftarrow$ LOOKUP(x)
9: **for all** phrase pairs $\bar{x}, \bar{y}$ such that $\bar{x} \in x$ **do**
10: $\phi_{match}(S(\bar{x}, \bar{y}), S(x)) \leftarrow$ AGGREGATE(INTERSECT($S(\bar{x}, \bar{y}), S(x)$))
11: **end for**

where $\mathbf{w}_{match}$ are additional feature weights, which can be trained on the heldout set along with the other feature weights.

Given a phrase pair $\bar{x}, \bar{y}$, the phrase pair side constraints $S(\bar{x}, \bar{y})$ are computed as the union over the side constraints of all training sentences that contain $\bar{x}, \bar{y}$. The side constraint match features are then computed by taking the intersection between the side constraints for test input x, $S(x)$, and the phrase pair side constraints $S(\bar{x}, \bar{y})$. Algorithm 5 shows the feature extraction process. Lines 2-4 show the extraction of phrase pair side constraints at training time. The LOOKUP procedure maps the sentence identifier of an input x to the side constraints extracted from the document-level metadata of the patent document containing x. Lines 7-11 then contain the extraction of the side constraint match features at test time. Side constraints for input x are obtained via LOOKUP, and the intersection is computed for each phrase pair in x. The new features are *dynamic features*, as they depend on the input document side constraints, whose values are only obtained at test time. The final feature values are computed by intersecting $S(x)$ with $S(\bar{x}, \bar{y})$, and feeding the intersection to an aggregation procedure AGGREGATE. Below, we motivate the need for feature aggregation and describe different aggregation schemes.

We propose three different ways of aggregating matching side constraints: no aggregation, aggregation by category and aggregation by sum. More aggregation produces fewer, more general features, while no aggregation allows the model to capture specific information, but can lead to sparsity issues. When doing *no aggregation*, each matching side constraint is treated as a separate binary feature. This method produces sparse, fine-grained features. However, the model will only learn weights for annotations it has seen during tuning. If the tuning set only contains few documents, few annotation features will be learned. The maximum number of features added by this method is equal to the number of different annotations present in the training data. We use *aggregation by category* to define more general features. This method aggregates all side constraints belonging to the same type

– `COMP, INV, IPC1, IPC2, IPC3` – in one single feature by summing over all side constraint matches in this category. For example, if a test document has three inventors, and a phrase was found in documents from two of the same inventors, this phrase will be annotated with the feature `INVMatch=2`. Aggregating by category allows the model to differentiate between types of side constraints, while avoiding the sparsity issues arising without aggregation. This method of aggregation always adds F features to the model, where F is the number of categories. Finally, *aggregation by sum*, is the most aggressive aggregation scheme. This method sums over all side constraint matches, adding only a single additional feature to the model. For example, if a phrase pair and an input document have two matching inventors and a matching IPC section, the phrase would be annotated with `MatchSum=3`.

The features we have defined so far capture whether a phrase pair shares the same side constraints as an unseen document. One weakness of this approach lies in its potential bias for more common phrases. Phrases that occur in more documents are more likely to have more matching side constraints. However, a phrase pair occuring frequently, but only within a single IPC class, for example, could be crucial for producing a correct translation for inputs from this class, but would only receive low aggregated match counts. We counter this weakness by introducing a weighting scheme inspired by *term frequency - inverse document frequency* (TFIDF) weighting. Originally, TFIDF weighting has been designed to weight individual terms for information retrieval: Terms are given high weight if they occur frequently in few documents. In our case, the weighting scheme should assign more weight to a phrase pair with respect to a side constraint, if the phrase pair is strongly associated with this side constraint. Less weight should be assigned to a phrase pair with respect to a side constraint, if this phrase pair also occurs with many other side constraints. We therefore define *documents* by the side constraints, e.g., the patent class or the name of the inventor. The *terms* are phrase pairs occurring in patents with this side constraint. Note that as one patent can have multiple side constraints, the same phrase pair occurrence can be assigned to more than one "*document*". In this regard our approach differs from classical TFIDF, where each term occurrence is assigned to exactly one document. The idea of using a TFIDF weighting scheme is inspired by the vector space adaptation model of B. Chen et al. (2013), who apply TFIDF weighting to measure the importance of phrase pairs with respect to different corpora for domain adaptation. In our setting, the term frequency TF of a phrase pair $\bar{x}, \bar{y}$ with respect to a given side constraint $s_i \in S(\bar{x}, \bar{y})$ is computed as follows:

$$\mathrm{TF}(\bar{x}, \bar{y}; s_i) = \frac{\mathrm{count}(\bar{x}, \bar{y}; s_i)}{\max_{\bar{x}', \bar{y}'}(\mathrm{count}(\bar{x}', \bar{y}'; s_i))},$$

using maximum TF-normalization (Manning et al., 2008, Section 6). The inverse document frequency IDF of $\bar{x}, \bar{y}$ is:

$$\mathrm{IDF}(\bar{x}, \bar{y}) = \log \frac{N}{\mathrm{DF}(\bar{x}, \bar{y})} + \beta \,,$$

where the document frequency $\mathrm{DF}(\bar{x}, \bar{y})$ is the number of side constraints $\bar{x}, \bar{y}$ occurs with, and N is the total number of side constraints. β is a smoothing parameter to avoid zero-values if $\mathrm{DF}(\bar{x}, \bar{y}) = N$. The TFIDF weight of $\bar{x}, \bar{y}$ is then computed as

$$\mathrm{TFIDF}(\bar{x}, \bar{y}; s_i) = \mathrm{TF}(\bar{x}, \bar{y}; s_i) \cdot \mathrm{IDF}(\bar{x}, \bar{y}) \,.$$

As an illustration, Figure 6.2 shows the 5 patent classes with the highest TFIDF weights for three translation options for "*Speicher*". We can see that "*storage*" has highest weight in classes concerning physical storage, while "*memory*" ranks highest in classes on computing and electronics. Both options are relevant for class G11, "INFORMATION STORAGE". For "*reservoir*", the top-ranked classes concern heating, thermal insulation and liquids.

Combining the variants described above, we conduct four sets of experiments:

(1) `single`: Sparse binary match features without aggregation or weighting.

(2) `single weighted`: Sparse match features with TFIDF weighting.

(3) `aggregated`: Binary match features from setup (1), either aggregated by side constraint type, or by summing over all matches.

(4) `aggregated weighted`: Weighted match features from setup (2), either aggregated by side constraint type, or by summing over all matches.

Top IPC classes for "*storage*"
F17, STORING OR DISTRIBUTING GASES OR LIQUIDS
E03, WATER SUPPLY; SEWERAGE
F24, HEATING; RANGES; VENTILATING
G11, INFORMATION STORAGE
E02, HYDRAULIC ENGINEERING; FOUNDATIONS; SOIL-SHIFTING

Top IPC classes "*memory*"
G06, COMPUTING; CALCULATING; COUNTING
H04, ELECTRIC COMMUNICATION TECHNIQUE
G11, INFORMATION STORAGE
G05, CONTROLLING; REGULATING
G01, MEASURING; TESTING

Top IPC classes for "*reservoir*"
F24, HEATING; RANGES; VENTILATING
F16, ENGINEERING ELEMENTS OR UNITS; GENERAL MEASURES FOR PRODUCING AND MAINTAINING EFFECTIVE FUNCTIONING OF MACHINES OR INSTALLATIONS; THERMAL INSULATION IN GENERAL
F03, MACHINES OR ENGINES FOR LIQUIDS; WIND, SPRING, OR WEIGHT MOTORS; PRODUCING MECHANICAL POWER OR A REACTIVE PROPULSIVE THRUST, NOT OTHERWISE PROVIDED FOR
F02, COMBUSTION ENGINES; HOT-GAS OR COMBUSTION-PRODUCT ENGINE PLANTS
E05, LOCKS; KEYS; WINDOW OR DOOR FITTINGS; SAFES

Figure 6.2 Most highly relevant patent classes for translation options for "*Speicher*".

6.3.2 Experimental Evaluation

We train Japanese-English and German-English patent translation systems using the data sets as described in Section 6.2. All SMT experiments use the CDEC toolkit (Dyer et al., 2010) for training and decoding. The baseline uses 21 built-in dense features, consisting of five translation model features, seven pass through features, two singleton features, one glue rule feature, one word penalty, three arity penalty features, and two language model features. We use a single 5-gram language model built with KENLM (Heafield et al., 2013) from the target side portion of the training data. For all experiments, we use the DTRAIN pairwise ranking optimizer (Simianer et al., 2012) to learn feature weights on the development set. We run DTRAIN with k-best size = 100, learning rate = $1e^{-7}$ and loss margin = 1 for 15 epochs and obtain final weights by averaging over all epochs. Cube pruning with a pop-limit of 200 is used for tuning and evaluation. We conduct feature ablations for the different types of side constraints on the devtest set. For each experiment group described above, we perform an ablation for the five side constraint types described in Section 6.2: Company (`COMP`), inventor (`INV`), IPC section (`IPC1`), class (`IPC2`), and subclass (`IPC3`). We test each constraint type on its own, as well as the combination of all constraint types. The best performing configuration within each experiment group is then evaluated on the test set. Performance is evaluated according to the BLEU score (Papineni et al., 2002). BLEU scores, as well as significance tests, were computed using the MULTEVAL tool (Clark et al., 2011) with tokenized, lowercased input.

Figure 6.3 shows the results of the feature ablation on the devtest set for the Japanese-English task. The `single` features without weighting perform worst overall, with only `IPC1` and `all` experiments outperforming the baseline. This result shows that apart from `IPC1` constraints, where only eight different labels are possible, other types of constraints could be too sparse for reliable weight estimation from a small development set of 2000 sentences when binary features are used. This is especially notable for the `INV` constraints which even decreases performance. The `single weighted` setting improves in performance for `COMP`, `IPC2` and `IPC3` constraints, showing that the added information by weighting is helpful. For the `aggregated` features without weighting, we see an increase over the baseline for all constraints with the largest increase for `COMP` and `INV` constraints. This result indicates that aggregating constraints leads to more helpful features by reducing sparsity. The `aggregated weighted` experiments achieved the best performance overall by either summing over all weighted match features or by using only the weighted `IPC2` constraints. As in the `single weighted` setting, we observe that weighting is more helpful for more fine-grained IPC constraints. However, unlike the `single weighted` setting, the features using `IPC3` constraints actually deteriorated over the baseline. Table 6.4 shows BLEU scores on the test set for the best configuration in each experimental setting.

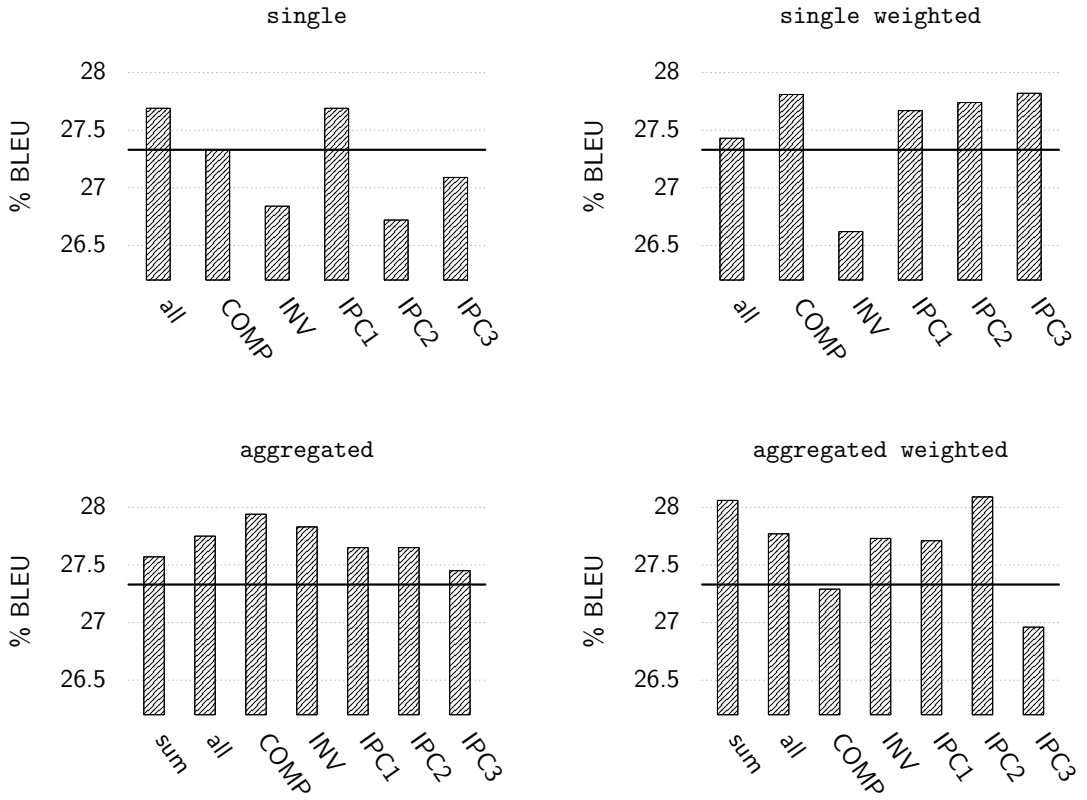


Figure 6.3 Devtest results for different configurations on Japanese-English patent translation. The horizontal line marks the baseline at 27.33% BLEU.

The improvements observed on the devtest set carry over to the test set, with the best configuration – `aggregated weighted IPC2` constraints – increasing the BLEU score by 0.6 BLEU points over the baseline.

	setting	constraint	% BLEU devtest	% BLEU test	Δ test
	baseline		27.3	26.2	
1	`single`	`IPC1`	**27.7**	26.3	+0.1
2	`single weighted`	`IPC3`	**27.8**	**26.5**	+0.3
3	`aggregated`	`COMP`	**27.9**	**26.7**	+0.5
4	`aggregated weighted`	`IPC2`	**28.1**	**26.8**	+0.6

Table 6.4 Test results for best experiments in each category on Japanese-English task. Results on devtest are shown for comparison. Boldface marks a significant improvement over the baseline.

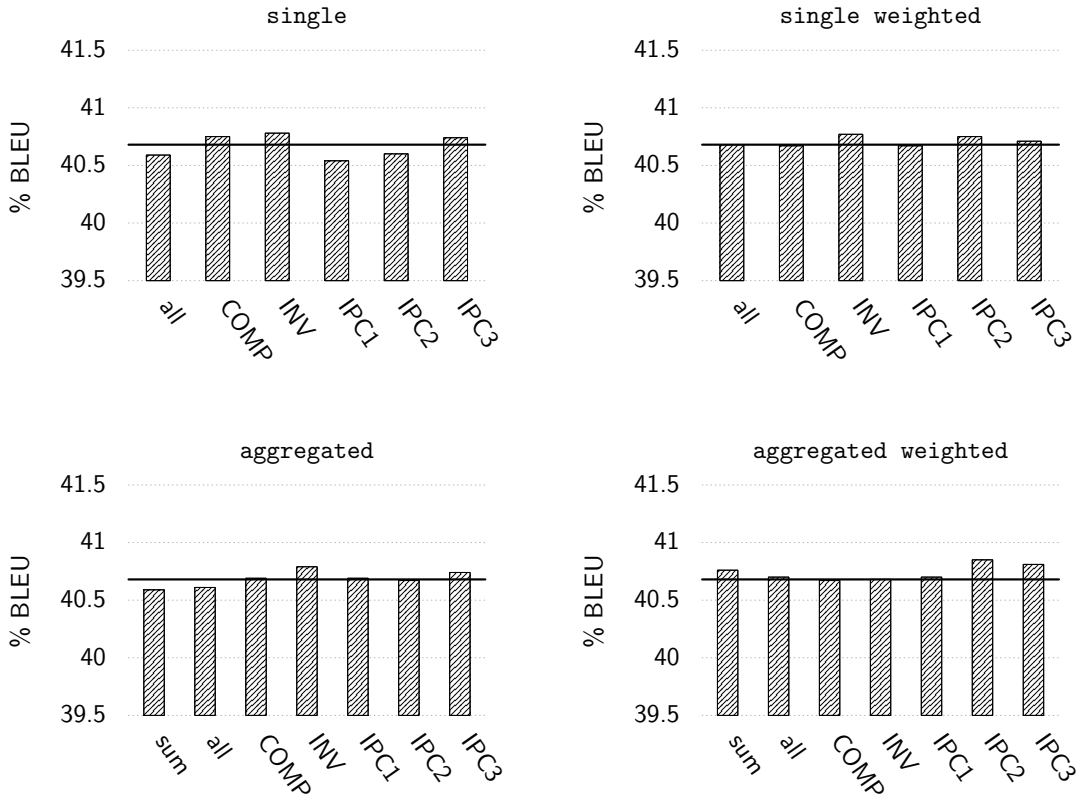


Figure 6.4 Feature ablations for different experiment settings on German-English patent translation. The horizontal line marks the baseline at 40.68% BLEU.

	setting	constraint	% BLEU devtest	% BLEU test	Δ test
	baseline		40.7	43.1	
1	`single`	`INV`	**40.8**	43.1	+0.0
2	`single weighted`	`INV`	**40.8**	43.0	+0.0
3	`aggregated`	`INV`	**40.8**	43.0	−0.1
4	`aggregated weighted`	`IPC2`	**40.9**	43.0	−0.1

Table 6.5 Test results for best experiments in each category on German-English task. Results on devtest are shown for comparison. Boldface marks a significant improvement over the baseline.

Figure 6.4 shows the ablation experiments for the German-English system. For German-English, all results are very close to the baseline with very small differences between runs. Unlike the Japanese experiments, the `INV` constraints produce a small gain for the `single`, `single weighted` and `aggregated` settings. The largest overall gain is obtained with `IPC2` constraints in the `aggregated weighted` setting. This result matches the Japanese experiments. Table 6.5 reports test results for the best-performing configurations on German-English. However, the gains on devtest are very small at only 0.1 to 0.2 BLEU points over the baseline and do not carry over to the test set.

Comparing both language pairs, we observe different patterns in the `single` and `single weighted` experiments. For example, the `INV` constraints hurt performance for Japanese-English in the `single` setup, while it produces small gains for German-English. This indicates that the success of sparse features from side constraints is very dependent on the data conditions. On the other hand, `IPC1` constraints work best for Japanese-English in the `single` setup, but perform below the baseline for German-English. The results are most consistent for the `aggregated weighted` setup, where `IPC2` constraints perform best overall in both language pairs. This points towards larger robustness of those features. Overall, results for German-English are very close to the baseline, while for Japanese-English there are improvements of up to 0.8 BLEU points on the devtest set. One potential explanation for the greater success on the Japanese-English task could be the lower overall performance of the Japanese-English system in terms of BLEU score, where the German-English baseline system is very strong already. A possible explanation for this performance gap could be the enhanced difficulty of Japanese-English translation due to word order differences. These differences are difficult to model, even for a hierarchical SMT system. This is particularly true for patent translation, where sentences tend to be very long and complex with dependencies ranging over larger spans than can be captured by SCFG rules, given the necessary constraints to make rule extraction computationally tractable.

To summarize, adding side constraint match features to an SMT system leads to mixed results. Ablation experiments on the devtest set show that feature aggregation by category in combination with TFIDF-weighting produces the largest gains over the baseline when `IPC2` constraints are used. For Japanese-English, this configuration also produces a significant improvement of +0.6 BLEU points on the test set. For German-English, no significant improvements are observed on the test set. We surmise that this difference is due to the strong baseline for German-English.

6.4 Side Constraints for NMT

Since 2014, neural machine translation (NMT) has become the state of the art in machine translation (Jean et al., 2015; M.-T. Luong & Manning, 2015). We hypothesize that NMT is well-suited to the integration of document meta-information. Since the model works on sentence representations, it can decide whether or not to pay attention to this information for each particular translation in context. What is more, side constraints could be highly correlated, such as patent class and subclass information. Deep models are capable of learning these correlations. Inspired by Kobus et al. (2017), two ways of integrating document information as side constraints into a neural machine translation system are explored: (1) by attaching side constraints as special tokens to the source sentence, and (2) by attaching side constraints as tags to each source word. We consider for both methods the need to handle multiple side constraints with multiple values for each constraint. We then show experimentally that side constraints can improve a neural patent translation system.

6.4.1 Sentence- and Word-attached Side Constraints

Our first method integrates side constraints by encoding them as special tokens that are attached at the beginning or end of a source sentence, or in both positions. Since the method has to be able to handle multiple values for each type of constraint, each value is appended to the sentence in alphabetical order. For example, if a patent document has been assigned to IPC sections B and C, the marked up input could look as follows:

```
<IPC1:B> <IPC1:C> in die Matrix sind Verstärkungsfasern (5, 6, 7)
                  eingebettet. <IPC1:B> <IPC1:C>
```

The advantages of this approach are its simplicity and efficiency, as it does not require any changes in the translation model. It also allows the model to learn to pay attention to the special token as needed. Further, the number of model parameters also does not change if vocabulary size remains the same. Previous work has differed on the attachment position, leading us to run experiments for attaching at the front, back, or both front and back of a sentence.

Our second method integrates side constraints by attaching them to each source word, as shown in the example below:

```
in|IPC1:B die|IPC1:B Matrix|IPC1:B sind|IPC1:B …
```

Note that in our case, every word can have more than one side constraint attached to it. Side constraints are encoded as F word-attached features where each feature represents one type of side constraints (e.g., `IPC1, IPC2` etc.). For each side constraint feature, a vector representation $\mathbf{s}_f, f = 1 \dots F$ is constructed and appended to the input word embedding $\mathbf{x}_j$, leading to the following equation for updating the hidden RNN encoder state:[6]

$$\mathbf{h}_j^{enc} = \tanh(\mathbf{W}^{enc}([\mathbf{x}_j, \mathbf{s}_1, \dots, \mathbf{s}_F]) + \mathbf{U}^{enc}\mathbf{h}_{j-1}^{enc}),$$

where $\mathbf{W}^{enc}$ and $\mathbf{U}^{enc}$ are the hidden layer parameters, $\mathbf{h}_{j-1}^{enc}$ is the previous hidden state, and $[\mathbf{v}_1, \mathbf{v}_2, \dots]$ signifies vector concatenation. Note that $\mathbf{s}_f$ is identical for each token in a document d. Each side constraint feature represents all side constraints belonging to type f, e.g. all `IPC1` constraints. We write individual constraints as $s_{f,k}$.

For constructing $\mathbf{s}_f$, we could either use a sparse vector representation, as is done by Kobus et al. (2017) for multi-domain adaptation, or a dense "feature embedding", as is done by Sennrich and Haddow (2016) to encode token-level linguistic annotation. We choose the dense approach, as a sparse representation requires the length of $\mathbf{s}_f$ to match the number of values for the f-th feature. In our setting, some side constraint types have thousands of possible values, which will become unwieldy for the recurrent model. The dense representation is computed by looking up the row corresponding to a particular side constraint in an embedding matrix, which is trained along with the model. This is much more efficient than computing a full matrix vector multiplication. For constraint $s_{f,k}$, the lookup operation is

$$\mathbf{E}_f[\cdot, s_{f,k}],$$

where $\mathbf{E}_f \in \mathbb{R}^{|emb| \times |V_f|}$ is an embedding matrix for side constraint feature f, and $\mathbf{M}[\cdot, n]$ designates column-wise lookup in matrix $\mathbf{M}$.

This description has not yet specified how to handle multiple side constraints $s_{f,1} \dots s_{f,k}$ for one feature. This occurs, for example, if a patent document belongs to more than one IPC class. Treating those side constraints as separate features would ignore the fact that they belong to the same side constraint type. We solve this issue by looking up all side constraints of the same type in the same embedding matrix $\mathbf{E}_f$ and summing over the embeddings as

$$\mathbf{s}_f = \sum_{k=1}^{K_f} \mathbf{E}_f[\cdot, s_{f,k}].$$

For efficiency reasons, we restrict the number of possible side constraints per side constraint feature, K_f, to a fixed size, which is selected by the experimenter.

[6] We omit bias terms and bidirectional RNNs here for the sake of readability.

Compared to sentence-attached side constraints as proposed above, the word-attached approach provides a more straightforward solution for combining multiple side constraints - by first aggregating all side constraints belonging to the same feature and then concatenating the aggregated embeddings. For the sentence-attached approach, there is no clear way how to aggregate individual side constraints, making it difficult to incorporate a large number of them. The concatenated embeddings also allow to model correlations between side constraints. The disadvantages of this method are an increase in computational complexity caused by additional lookup operations and vector concatenation and, if the token embedding $\mathbf{x}_j$ remains unchanged, an increase in parameters.

6.4.2 Experimental Evaluation

Experimental Setup. We use the NEMATUS NMT system[7] (Sennrich et al., 2017) to train a recurrent encoder-decoder network with attention (Bahdanau et al., 2015). After pre-processing, we train a subword model using Byte Pair Encoding (BPE) (Sennrich et al., 2016c) with 29,500 merge operations on the training set. For training, we only keep sentences up to length 80. Unless specified otherwise we use 500-dimensional word embeddings and a single-layer encoder and decoder with 1,024 hidden units each. All models are trained using cross entropy loss. ADADELTA (Zeiler, 2012) is employed for optimization on minibatches of 80 sentences. Gradients are re-normalized if their L2 norm exceeds a threshold of 1.0. Training is stopped if cross entropy does not decrease for 10 validations. Results are reported on the model with lowest cross entropy. BLEU score and significance tests are computed using MULTEVAL Clark et al. (2011).

As models with side constraints have to be trained from scratch, performing an ablation like in Section 6.3 would take months with limited GPU resources available. We therefore only explore `IPC1`, `IPC2` and `COMP` constraints, which have been more successful in the SMT experiments. For the sentence-attached side constraints, we experiment with IPC section (`IPC1`) and class (`IPC2`) constraints, as well as `COMP` for Japanese-English. We consider attachment at the `front`, `back`, or `front+back` of the source sentence. For the word-attached side constraints we combine IPC section and class constraints (`IPC1+IPC2`). The hyperparameters K_f are determined by the distribution of the number of values for each side constraint in the training documents. K_f is chosen to ensure that less than 5% of training documents have more than K_f side constraints of type f. If a document has more side constraints of type f, a subset of size K_f is sampled randomly. For documents with fewer side constraints of type f, empty values are marked by an extra dummy token. For `IPC1`, we set $K = 2$, for `IPC2`, $K = 3$. The embedding dimensions are set to 5 for `IPC1`

[7] `github.com/EdinburghNLP/nematus`

	setting	constraint	% BLEU	Δ BLEU
1	baseline		36.9	
	sentence-attached			
2	`front`	`IPC1`	37.4	+0.5
3	`back`	`IPC1`	**37.5**	+0.6
4	`front+back`	`IPC1`	**37.6**	+0.7
5	`front`	`IPC2`	37.2	+0.3
6	`front`	`COMP`	37.0	+0.1
	word-attached			
7	$K = \{2, 3\}$	`IPC1,IPC2`	37.2	+0.3

Table 6.6 Japanese-English translation results. Boldface marks a significant improvements over the NMT baseline at $p \leq 0.05$.

and 20 for `IPC2`. Embedding sizes are chosen to reflect the fact that `IPC2` has over an order of magnitude more different annotations than `IPC1`. In order to avoid improvements from merely increasing the number of parameters, we use 475-dimensional source word embeddings when testing word-attached side constraints. The concatenated embedding vectors then have the same length (500) as the original word embedding vectors.

Experimental Results. Table 6.6 shows results for Japanese-English neural patent translation. Switching to a neural machine translation model leads to a striking improvement in BLEU of about 10 points compared to the SMT experiments in Section 6.3. This is most likely due to the neural model's superior ability to deal with differences in word order between the two languages. These differences create long-range dependencies, which could be exasperated by the long, complex sentences common in the patent domain. Sentence-attached `IPC1` constraints produce a small, significant, improvement over the NMT baseline (+0.7% BLEU), when attached at the `front+back` of the sentence or at the *back* of the sentence. Sentence-attached constraints (`IPC1, IPC2, COMP`) at the `front` of the sentence do not improve significantly over the baseline. Word-attached constraints (`IPC1+IPC2`) also do not improve the baseline significantly.

Table 6.7 shows results for German-English patent translation. For this language pair, baseline performance is on a par with SMT. This points to SMT being well-suited to patent translation when there is no large difference in word order. We surmise that the formulaic nature of patent language along with the absence of named entities and the large amount of available training data play a role in this result. Using document infor-

	setting	constraint	% BLEU	Δ BLEU
1	baseline		42.5	
	sentence-attached			
2	`front`	`IPC1`	**43.5**	+1.0
3	`back`	`IPC1`	**43.2**	+0.7
4	`front+back`	`IPC1`	42.7	+0.2
5	`front+back`	`IPC2`	**43.9**	+1.4
	word-attached			
6	$K = \{2, 3\}$	`IPC1,IPC2`	**43.5**	+1.0

Table 6.7 German-English translation results. Boldface marks a significant improvement over the NMT baseline at $p \leq 0.05$.

mation as side constraints produces larger improvements over the NMT baseline than for Japanese-English. Sentence-attached `IPC1` constraints at the `front` of the sentence improve the NMT baseline significantly (+1% BLEU). Sentence-attached `IPC2` constraints at the `front and back` also produce a significant improvement (+1.4% BLEU). Word-attached constraints lead to a significant improvement of +1% BLEU, but do not outperform sentence-attached side constraints. These results differ from previous work, where sentence-attached domain or topic labels have produced no gains.[8]

When comparing attachment location for sentence-attached `IPC1` constraints, Japanese-English experiments show improvements for attachment at the `back` and `front+back`. For German-English, on the other hand, attachment at the `front` produces better BLEU scores than attachment at the `back`, and `front+back` is worst. We can conclude from our experiments that there is no general recommendation on which attachment location is best and that it is worth experimenting with different attachment locations.

Table 6.8 shows example input sentences from the German-English test set and their translations by different models. In Example 1, the NMT model with side constraints correctly translated the German word "*Kupplung*" as "*clutch*", which was incorrectly translated as "*coupling*" by the SMT and NMT baseline. The correct phrase translation for "*elektrischen Maschine*", "*electric machine*", was also only selected by the model with side constraints. In Example 2, the correct translation "*impact plates*" for German "*Prallplatten*" was produced by all NMT models. However, the word "*Wasserschleiers*" was only translated correctly as "*water curtain*" by the model with side constraints. It was passed through the

[8] We performed an experiment combining sentence- and word-attached constraints but did not see additional gains.

Example 1	
Source	(…) mit einer **Kupplung** (8) , die den Verbrennungsmotor (2) auswählbar vollständig mit der **elektrischen Maschine** (10) selektiv verbindet (…)
SMT baseline	(…) with a coupling (8) , the internal combustion engine (2) can be completely selectively connects with the electrical machine (10)
NMT baseline	(…) comprising a coupling (8) which can be selectively connected to the electric motor (10) (…)
`IPC1 front`	(…) comprising a **clutch** (8) which selectively connects the internal combustion engine (2) to the **electric machine** (10) (…)
Reference	(…) having a **clutch** (8) which selectively connects the internal combustion engine (2) completely to the **electric machine** (10) (…)
Example 2	
Source	Sie weist (…) **Prallplatten** (531) auf zur Erzeugung eines flächigen **Wasserschleiers** (25) zumindest im oberen Bereich der **Dampfkabine** (…)
SMT baseline	it has (…) deflector plates (531) for generating a flat wasserschleiers (25) at least in the upper region of the cubicle (…)
NMT baseline	(…) **impact plates** (531) , at least in the upper region of the steam booth (…)
`IPC1` front	(…) it has (…) **impact plates** (531) for producing a flat **water curtain** (25) at least in the upper region of the **steam cabin** (…)
Reference	(…) said steam cabin comprises (…) **impact plates** (531) for producing a flat **water curtain** (25) at least in the upper area of the **steam cabin** . (…)

Table 6.8 Examples for German-English translation. **Bold-faced** portions highlight correct translations. Incorrect translations are underlined.

decoder in SMT and omitted entirely by the NMT baseline. The NMT model with side constraints also selected the correct translation "*steam cabin*" for German "*Dampfkabine*", where SMT produced "*cubicle*" and the NMT baseline produced "*steam booth*".

Conclusions

In this chapter we have investigated the usefulness of document meta-information as a means to inject knowledge about the document context into a fully supervised scenario. More specifically, document metadata are integrated into a translation model via source side constraints, which contain information that is not explicit in the source sentence but may be relevant to the target translation. The side constraints were evaluated on patent translation, as patents contain rich document metadata. Experiments were conducted for Japanese-English and German-English patent translation. Side constraints were defined from five metadata categories – patent section, class, subclass, inventor and company, where a patent could be annotated with more than one value from each of these categories.

For SMT, we proposed phrase-level side constraint match features, along with two aggregation strategies and a TFIDF-inspired weighting scheme. The weighting scheme assigns different weights to side constraint matches, depending how important a phrase is with respect to a side constraint. The side constraints were evaluated individually and in combination, with and without aggregation and weighting. Models using the `aggregated weighted` setting were found to be most promising with a significant improvement on the Japanese-English translation task for the best-performing setup using the `IPC2` constraint.

We then moved to an NMT system, taking into account the superior performance of neural sequence-to-sequence models on many translation tasks. The experiments confirmed the superiority of NMT for Japanese-English patent translation, showing impressive gains of almost 10% BLEU over the SMT baseline. For German-English patent translation, we found the two systems performing on par - showing that SMT can handle patent translation well if source and target language do not differ much in word order. For NMT, we proposed two methods of integrating side constraints, which are inspired by previous work on multi-domain adaptation: as sentence-attached tokens or word-attached features. Sentence-attached side constraints are very easy to implement and do not require additional model parameters, while word-attached features are better at combining different side constraints. For word-attached features we devised a method to handle multiple side constraints of the same type by summing over individual embeddings from a joint embedding matrix. We evaluated sentence- and word-attached side constraints, and found modest improvements for Japanese-English with sentence-attached `IPC1` constraints. For German-English we

observed larger improvements with the highest improvement of 1.4% BLEU achieved by sentence-attached `IPC2` constraints. Word-attached side constraint features also produced a significant improvement of 1 BLEU point.As for the attachment location of sentence-attached features, we found that results varied between the two translation tasks, showing the need for careful experimentation with these features. Comparing SMT and NMT we conclude that NMT is better-suited for integrating side constraints, especially in the German-English task where no improvements were observed in the SMT system.

While this chapter has explored side constraints for patent translation, the presented approaches are not restricted to patent translation, but could also be applied to other domains where suitable document meta-information is available, such as e-commerce.

CHAPTER 7

Summary and Concluding Remarks

Even though significant progress has been made in machine translation, two issues limit its potential: First, machine translation relies on supervision from sentence-parallel data. Second, machine translation in its current setup only takes into account sentence-level context. By requiring supervision from sentence-parallel data, performance degrades in low-resource conditions. By ignoring the document context, translation quality is limited, as some ambiguities cannot be resolved, even with human-level translation quality.

This thesis has aimed to contribute towards the solution of both issues by integrating document meta-information into data-driven machine translation. Such information exists in many multilingual data collections, but is rarely used in machine translation. Our intuition has been that the first issue can be addressed by using document meta-information as a source of weak supervision in low-resource settings, compensating to some extent for the lack of supervision from parallel data. Further, that the second issue can be addressed by using weak supervision from document meta-information to supplement full sentence-level supervision with document-level context. We have explored these intuitions in four cumulatively conducted case studies where data-driven machine translation was improved by incorporating document meta-information as weak supervision. Below, we summarize the findings of each case study and draw general conclusions.

Our first case study concerned topically constrained Arabic-English Twitter translation. In this scenario no in-domain supervision is available. Document meta-information from user-generated hashtags was used to identify topically relevant messages and to automatically extract in-domain pseudo-parallel data using a cross-lingual information retrieval (CLIR) model. The shortness of the messages due to the character limit allowed us to treat them as single sentences and integrate pseudo-parallel message pairs directly into a statistical machine translation (SMT) system without performing an additional sentence alignment step. By applying a simple, bold alignment strategy to these noisy pairs, and combining the extracted phrases with an out-of-domain phrase table, we were able to significantly improve performance on a crowd-sourced test set. Further analysis showed a large reduction in out-of-vocabulary words. Our work was the first to apply CLIR techniques for the extraction of pseudo-parallel data to the Twitter domain. Document meta-information was leveraged in a novel way and even though the extracted pairs were noisy, using a simple, bold alignment strategy produced much better results than more sophisticated, but conservative approaches to extracting phrases from the aligned pairs.

Our second study examined the case where translations were generated for use in a downstream task, which has different requirements from end-to-end translation. For our example task, cross-lingual patent prior art search, supervision in the form of task-appropriate query translations was not available. We therefore used document meta-information in the form of cross-lingual patent citations to simulate a task loss. This loss was then used in a pairwise ranking algorithm to tune sparse and dense feature weights of an SMT system for the task of cross-lingual patent retrieval. In experiments our method performed numerically better according to various CLIR metrics than systems with manually set weights or weights adapted for translation quality.

In our third study, we considered a setup for end-to-end translation under extremely weak supervision. We conducted domain adaptation experiments where no in-domain parallel data was available, using Wikipedia translation as our example task. Supervision was only available from document-level links of stronger or weaker relevance. We presented bipolar structured ramp loss training objectives, which take into account both weak positive supervision from relevant documents and negative supervision in the form of irrelevant documents. We also defined two new reward metrics for the specific weak supervision setup. For SMT, we successfully trained sparse lexicalized in-domain features using our new objectives and metric. We investigated the difference between stronger and weaker links, as well as the difference between loss-augmented inference and inference over k-best lists. We found that even weaker links provided helpful supervision, although stronger links produced better results. We also found loss-augmented inference over the translation forest to be necessary for successful learning. Finally, we showed improvements even over a strong adapted baseline. We then lifted our approach to the neural machine

translation (NMT) paradigm, focusing on comparing bipolar variants of the ramp loss to non-bipolar ones and on measuring the contribution of negative supervision. In the weakly supervised scenario, we demonstrated that gains over the unadapted system were only obtained when using a bipolar objective which incorporated both positive and negative supervision, while the objectives using only positive or only negative supervision failed to learn. This finding was also confirmed for a minimum risk training objective. We further confirmed the success of ramp loss on a fully supervised translation task.

Our final case study returned to the patent translation setup where supervision from sentence-parallel data is available. In this study we incorporated document metadata as side constraints on top of the sentence-level supervision from parallel data. These side constraints allowed us to include information about the document context. We studied the performance of both SMT and NMT systems on a patent translation task. For SMT, we integrated side constraints as additional phrase features. We experimented with different aggregation and weighting schemes for the phrase features and performed an ablation of different feature groups. We found modest improvements for Japanese-English patent translation. For NMT, side constraints were implemented both as special tokens attached to the source sentence, and as tags attached to the source words. We devised a method for efficiently learning dense representations for word-attached side constraints belonging to the same constraint category. Our experiments showed improvements for both German-English and Japanese-English patent translation, with the sentence-attached tokens outperforming the word-attached tags.

To summarize, in this thesis, we have cumulatively conducted four case studies which successfully made use of document meta-information as weak supervision for data-driven machine translation. We have shown success on tasks as diverse as Twitter translation and patent prior art search. We have identified not only document metadata, but also document citations, document-level links and other document characteristics as useful information. The information has been incorporated into machine translation systems in several ways: First, indirectly, via pseudo-parallel data or as a downstream task simulation. Second, directly, via a substitute training objective in an incomplete supervision scenario or in the form of source side constraints, which allow the model to take into account document context information. As the indirect methods are agnostic to the translation system, we have only presented findings using the SMT paradigms. For the direct ways, we have adapted the approaches originally designed for SMT to the NMT paradigm, taking into account the differences between both paradigms.

Our work set out to leverage document meta-information to address two issues of data-driven machine translation: First, its reliance on parallel data, leading to translation quality decay in low-resource domains, and second, its focus on the sentence level, lead-

ing to translation quality restrictions even at human performance levels. We have contributed to the first issue by (a) leveraging document meta-information for obtaining pseudo-parallel in-domain training data and (b) by using weak supervision from document meta-information directly in a weakly supervised training objective, either via simulating a downstream task loss or as a direct replacement for reference translations. Approach (a) was very successful on the investigated task, while approach (b) produced smaller improvements. However, (a) was designed with a specific task in mind, incorporating characteristics of the data set, and could not easily be transferred to any other domain. Approach (b), on the other hand, is generally applicable where document-level links exist and, for SMT, also explicitly shows the success of using even very weak sources of supervision. It also makes apparent the interesting principle that negative supervision is helpful in a weak supervision scenario. Finally, both approaches are orthogonal and could be combined for further gains, as we demonstrated for SMT on Wikipedia translation. We have addressed the second issue by proposing to pass document meta-information to the translation model as side constraints. Experimentally, we have shown the success of these constraints for neural patent translation on top of fully supervised training. Orthogonally to our work, recent work has also tackled these issues, but has focused on weak supervision from textual information rather than from meta-information, leaving an investigation of the benefits of combining both research directions as a promising future outlook.

Bibliography

Alegria, I., Aranberri, N., España Bonet, C., Gamallo, P., Gonçalo Oliveira, H., Martínez Garcia, E., … Zubiaga, A. (2015). Overview of TweetMT: A shared task on machine translation of tweets at SEPLN 2015. In *Proceedings of the Tweet Translation Workshop 2015.*

Artetxe, M., Labaka, G., Agirre, E., & Cho, K. (2018). Unsupervised neural machine translation. In *Proceedings of the International Conference on Learning Representations (ICLR).*

Axelrod, A., He, X., & Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR).* Retrieved from `http://arxiv.org/abs/1409.0473`

Bakhshaei, S., Safabakhsh, R., & Khadivi, S. (2019). Extracting parallel fragments from comparable documents using a generative model. *Computer Speech & Language*, *53*, 25 - 42.

Bentivogli, L., Bisazza, A., Cettolo, M., & Federico, M. (2016). Neural versus phrase-based machine translation quality: A case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Bertoldi, N., & Federico, M. (2009). Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation (WMT).*

Bisazza, A., Ruiz, N., & Federico, M. (2011). Fill-up versus interpolation methods for phrase-based SMT adaptation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT).*

Brown, P. F., Della Pietra, V. J., Della Pietra, S. A., & Mercer, R. L. (1993, June). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, *19*(2), 263–311.

Buckland, M. K. (2015). Document theory: An introduction. In *Records, Archives and Memory: Selected Papers from the Conference and School on Records, Archives and Memory Studies.*

Cettolo, M., Federico, M., & Bertoldi, N. (2010). Mining parallel fragments from comparable texts. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT).*

Cettolo, M., Girardi, C., & Federico, M. (2012). WIT^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT).*

Chapelle, O., Do, C. B., Teo, C. H., Le, Q. V., & Smola, A. J. (2009). Tighter bounds for structured estimation. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in Neural Information Processing Systems 21* (pp. 281–288). Curran Associates, Inc.

Chen, B., & Cherry, C. (2014). A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT).*

Chen, B., Cherry, C., Foster, G., & Larkin, S. (2017). Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation.*

Chen, B., & Huang, F. (2016). Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL).*

Chen, B., Kuhn, R., & Foster, G. F. (2013). Vector space model for adaptation in statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics.*

Chen, W., Matusov, E., Khadivi, S., & Peter, J. (2016). Guided alignment training for topic-aware neural machine translation. In *Proceedings of the Twelfth Conference of the Association for Machine Translation in the Americas (AMTA).*

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL).*

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, *33*(2), 201–228.

Chiang, D. (2012). Hope and fear for discriminative training of statistical translation models. *The Journal of Machine Learning Research*, *13*(1), 1159–1187.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for

statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Chu, C., Dabre, R., & Kurohashi, S. (2017). An empirical comparison of simple domain adaptation methods for neural machine translation. *CoRR*, *abs/1701.03214*. Retrieved from `http://arxiv.org/abs/1701.03214`

Clark, J. H., Dyer, C., Lavie, A., & Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short papers (ACL).*

Collins, M. (2002). Discriminative training methods for hidden Markov models: Theory and experiments with Perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Collobert, R., & Bengio, S. (2004). Links between perceptrons, MLPs and SVMs. In *Proceedings of the twenty-first international conference on Machine learning (ICML).*

Collobert, R., Sinz, F., Weston, J., & Bottou, L. (2006). Trading convexity for scalability. In *Proceedings of the 23rd International Conference on Machine Learning (ICML).*

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493–2537.

Crego, J. M., Kim, J., Klein, G., Rebollo, A., Yang, K., Senellart, J., … Zoldan, P. (2016). SYSTRAN's pure neural machine translation systems. *CoRR*, *abs/1610.05540*.

Criscuolo, P., & Verspagen, B. (2008). Does it matter where patent citations come from? Inventor vs. examiner citations in European patents. *Research Policy*, *37*(10), 1892 - 1908.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, 1–38.

Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., … Resnik, P. (2010). cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL).*

Edunov, S., Ott, M., Auli, M., Grangier, D., & Ranzato, M. (2018). Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).*

Efraimidis, P. S., & Spirakis, P. G. (2006). Weighted random sampling with a reservoir. *Information Processing Letters*, *97*(5), 181 - 185.

Eidelman, V., Boyd-Graber, J., & Resnik, P. (2012). Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL).*

Eisele, A., & Chen, Y. (2010). MultiUN: A multilingual corpus from united nation documents. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC).*

Eisenstein, J. (2013). What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).*

Feng, M., Schmidt, C., Wuebker, J., Peitz, S., Freitag, M., & Ney, H. (2011). The RWTH Aachen system for NTCIR-9 PatentMT. In *Proceedings of the 9th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access.*

Foster, G., & Kuhn, R. (2007). Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation (WMT).*

Freitag, M., & Al-Onaizan, Y. (2016). Fast domain adaptation for neural machine translation. *CoRR*, *abs/1612.06897*. Retrieved from `http://arxiv.org/abs/1612.06897`

Fung, P., & Cheung, P. (2004). Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML).*

Gimpel, K., & Smith, N. A. (2012). Structured ramp loss minimization for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).*

Goldberg, Y. (2015). A primer on neural network models for natural language processing. *CoRR*, *abs/1510.00726*. Retrieved from `http://arxiv.org/abs/1510.00726`

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* Cambridge, Massachusetts, USA: MIT Press.

Goto, I., Chow, K. P., Lu, B., Sumita, E., & Tsou, B. K. (2013). Overview of the patent machine translation task at the NTCIR-10 workshop. In *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies.*

Graf, E., & Azzopardi, L. (2008). A methodology for building a patent test collection for prior art search. In *Proceedings of the Second International Workshop on Evaluating Information Access (EVIA).*

Grave, E., Joulin, A., & Usunier, N. (2017). Improving neural language models with a continuous cache. In *Proceedings of the International Conference on Learning Representations (ICLR).*

Green, S., Cer, D., & Manning, C. (2014). An empirical comparison of features and tuning for phrase-based machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT).*

Gupta, R., Pal, S., & Bandyopadhyay, S. (2013). Improving MT system using extracted parallel fragments of text from comparable corpora. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora.*

Habash, N., & Rambow, O. (2005). Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL).*

Hardmeier, C., & Federico, M. (2010). Modelling pronominal anaphora in statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT).*

Hardmeier, C., Nivre, J., & Tiedemann, J. (2012). Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL).*

Hasler, E., Haddow, B., & Koehn, P. (2014). Dynamic topic adaptation for SMT using distributional profiles. In *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT).*

Hazan, T., Keshet, J., & McAllester, D. A. (2010). Direct loss minimization for structured prediction. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, & A. Culotta (Eds.), *Advances in Neural Information Processing Systems 23* (pp. 1594–1602). Curran Associates, Inc.

Heafield, K., Pouzyrevsky, I., Clark, J. H., & Koehn, P. (2013, August). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL).*

Hewavitharana, S., Bach, N., Gao, Q., Ambati, V., & Vogel, S. (2011). CMU Haitian Creole-English translation system for WMT 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation (WMT).*

Hewavitharana, S., Mehay, D., Ananthakrishnan, S., & Natarajan, P. (2013). Incremental topic-based translation model adaptation for conversational spoken language translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL).*

Hieber, F., & Riezler, S. (2015). Bag-of-words forced decoding for cross-lingual information retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hopkins, M., & May, J. (2011). Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Hu, C., Resnik, P., Kronrod, Y., Eidelman, V., Buzek, O., & Bederson, B. B. (2011). The value of monolingual crowdsourcing in a real-world translation scenario: Simulation

using Haitian Creole emergency SMS messages. In *Proceedings of the Sixth Workshop on Statistical Machine Translation (WMT).*

Hutchins, J. (2007). Machine translation: A concise history. *Computer aided translation: Theory and practice*, *13*, 29–70.

Järvelin, K., & Kekäläinen, J. (2002, October). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, *20*(4), 422–446.

Jean, S., Firat, O., Cho, K., Memisevic, R., & Bengio, Y. (2015). Montreal neural machine translation systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation (WMT).*

Jehl, L., Hieber, F., & Riezler, S. (2012). Twitter translation using translation-based cross-lingual retrieval. In *Proceedings of the Seventh Workshop on Statistical Machine Translation.*

Jehl, L., Lawrence, C., & Riezler, S. (2018). *Learning Neural Sequence-to-Sequence Models from Weak Feedback with Bipolar Ramp Loss.* (Under Submission)

Jehl, L., & Riezler, S. (2016). Learning to translate from graded and negative relevance information. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers.*

Jehl, L., & Riezler, S. (2018). Document information as side constraints for improved neural patent translation. In *Proceedings of the 13th Conference of The Association for Machine Translation in the Americas (AMTA).*

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., … Dean, J. (2016). Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, *abs/1611.04558*. Retrieved from `http://arxiv.org/abs/1611.04558`

Khadivi, S., Wilken, P., Dahlmann, L., & Matusov, E. (2017). Neural and statistical methods for leveraging meta-information in machine translation. In *Proceedings of the 16th machine translation summit (MT Summit XVI).*

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR).*

Kobus, C., Crego, J., & Senellart, J. (2017). Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP).*

Koehn, P. (2004). Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (AMTA).*

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the tenth machine translation summit (MT summit X).*

Koehn, P. (2010). *Statistical machine translation.* Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., … Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *ACL 2007 Proceedings of the Interactive Poster and Demonstration Sessions.*

Koehn, P., & Knight, K. (2003). Empirical methods for compound splitting. In *Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics (EACL).*

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL).*

Koehn, P., & Schroeder, J. (2007). Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation (WMT).*

Lample, G., Denoyer, L., & Ranzato, M. (2018). Unsupervised machine translation using monolingual corpora only. In *Proceedings of the International Conference on Learning Representations (ICLR).*

Lee, Y.-S., Papineni, K., Roukos, S., Emam, O., & Hassan, H. (2003). Language model based Arabic word segmentation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL).*

Liang, P., Bouchard-Côté, A., Klein, D., & Taskar, B. (2006). An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL-COLING).*

Ling, W., Marujo, L., Dyer, C., Black, A. W., & Trancoso, I. (2014). Crowdsourcing high-quality parallel data extraction from Twitter. In *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT).*

Ling, W., Xiang, G., Dyer, C., Black, A., & Trancoso, I. (2013). Microblogs as parallel corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL).*

Linn, A. (2018, March 14). *Microsoft reaches a historic milestone, using AI to match human performance in translating news from Chinese to English* [Blog post]. Retrieved from `https://blogs.microsoft.com/ai/machine-translation-news-test-set-human-parity/`

Lopez, A. (2007). Hierarchical phrase-based translation with suffix arrays. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL).*

Lopez, A. (2008). Statistical machine translation. *ACM Computing Surveys (CSUR)*, *40*(3), 8.

Luong, M.-T., & Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT).*

Luong, T., Pham, H., & Manning, C. D. (2015). Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing.*

Ma, J., & Matsoukas, S. (2011). BBN's systems for the Chinese-English sub-task of the NTCIR-9 PatentMT evaluation. In *Proceedings of the 9th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access.*

Maamouri, M., Bies, A., Buckwalter, T., & Mekki, W. (2004). The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR conference on Arabic language resources and tools.*

Magdy, W., & Jones, G. J. (2010). PRES: A score metric for evaluating recall-oriented information retrieval applications. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR).*

Magdy, W., & Jones, G. J. (2011). An efficient method for using machine translation technologies in cross-language patent search. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM).*

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval.* Cambridge University Press.

Matsoukas, S., Rosti, A.-V. I., & Zhang, B. (2009). Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH-2010.*

Munteanu, D. S., & Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, *31*(4), 477–504.

Munteanu, D. S., & Marcu, D. (2006). Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL).*

Nakov, P., Guzman, F., & Vogel, S. (2012). Optimizing for sentence-level BLEU+1 yields short translations. In *Proceedings of COLING 2012: Technical Papers.*

Neubig, G. (2017). Neural machine translation and sequence-to-sequence models: A tutorial. *CoRR*, *abs/1703.01619*. Retrieved from `http://arxiv.org/abs/1703.01619`

Ney, H. (1995, February). On the probabilistic interpretation of neural network classifiers and discriminative training criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(2), 107–119.

Niehues, J., & Waibel, A. (2010). Domain adaptation in statistical machine translation using factored translation models. In *Proceedings of the Annual Conference of the European Association for Machine Translation (EAMT).*

Nikoulina, V., Kovachev, B., Lagos, N., & Monz, C. (2012). Adaptation of statistical machine translation model for cross-lingual information retrieval in a service context. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL).*

Noreen, E. W. (1989). *Computer intensive methods for testing hypotheses. an introduction.* New York: Wiley.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL).*

Och, F. J., & Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL).*

Och, F. J., & Ney, H. (2003, March). A systematic comparison of various statistical alignment models. *Computational Linguistics*, *29*(1), 19–51.

Och, F. J., & Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, *30*(4), 417-449.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL).*

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning (ICML).*

Piroi, F., & Tait, J. (2010). *CLEF-IP 2010: Retrieval experiments in the intellectual property domain* (Tech. Rep.). Vienna, Austria: Information Retrieval Facility.

Post, M., Callison-Burch, C., & Osborne, M. (2012). Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation (WMT).*

Quirk, C., Udupa, R., & Menezes, A. (2007). Generative models of noisy translations with applications to parallel fragment extraction. In *Proceedings of the Eleventh Machine Translation Summit (MT Summit XI).*

Ranzato, M., Chopra, S., Auli, M., & Zaremba, W. (2016). Sequence level training with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR).*

Riezler, S., & Maxwell, J. (2005). On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL-05 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization.*

Roda, G., Tait, J., Piroi, F., & Zenz, V. (2010). CLEF-IP 2009: Retrieval experiments in the intellectual property domain. In C. Peters et al. (Eds.), *Multilingual Information Access Evaluation I. Text Retrieval Experiments* (pp. 385–409). Berlin, Heidelberg: Springer Berlin Heidelberg.

Rosen, A. (2017, November 7). *Tweeting made easier [Blog post]* [Blog post]. Retrieved from `https://blog.twitter.com/official/en_us/topics/product/2017/tweetingmadeeasier.html`

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533.

Ruppert, E. (2013). *Cross-lingual patent retrieval for large data collections* (Unpublished master's thesis). Heidelberg University, Germany.

Schamoni, S., Hieber, F., Sokolov, A., & Riezler, S. (2014). Learning translational and knowledge-based similarities from relevance rankings for cross-language retrieval. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL).*

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., … Nadejde, M. (2017). Nematus: A toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL).*

Sennrich, R., & Haddow, B. (2016). Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation (WMT).*

Sennrich, R., Haddow, B., & Birch, A. (2016a). Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).*

Sennrich, R., Haddow, B., & Birch, A. (2016b). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL).*

Sennrich, R., Haddow, B., & Birch, A. (2016c). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL).*

Shen, L., & Joshi, A. (2005). Ranking and reranking with perceptron. *Machine Learning*, *60*(1), 73–96.

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., & Liu, Y. (2016). Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL).*

Simianer, P., Riezler, S., & Dyer, C. (2012). Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL).*

Simianer, P., Stupperich, G., Jehl, L., Wäschle, K., Sokolov, A., & Riezler, S. (2013). The HDU discriminative SMT system for constrained data PatentMT at NTCIR10. In *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies.*

Smith, J. R., Saint-Amand, H., Plamada, M., Koehn, P., Callison-Burch, C., & Lopez, A. (2013). Dirt cheap web-scale parallel text from the Common Crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL).*

Sokolov, A., Jehl, L., Hieber, F., & Riezler, S. (2013). Boosting cross-language retrieval by learning bilingual phrase associations from relevance rankings. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Spärck Jones, K., Walker, S., & Robertson, S. (2000). A probabilistic model of information retrieval: Development and comparative experiments: Part 2. *Information Processing & Management*, *36*(6), 809 - 840.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929-1958.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing.*

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 27* (pp. 3104–3112). Curran Associates, Inc.

Syria unrest: Who are the shabiha? (2012, May 29). Retrieved from `https://www.bbc.com/news/world-middle-east-14482968`

Tiedemann, J. (2010). Context adaptation in statistical machine translation using models with exponentially decaying cache. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing.*

Tillmann, C. (2004). A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers.*

Tillmann, C., & Xu, J. (2009). A simple sentence-level extraction algorithm for comparable data. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistic (NAACL-HLT).*

Tromble, R. W., Kumar, S., Och, F., & Macherey, W. (2008). Lattice minimum bayes-risk decoding for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Ture, F., & Lin, J. (2012). Why not grab a free lunch? Mining large corpora for parallel sentences to improve translation modeling. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).*

Ture, F., Lin, J., & Oard, D. (2012). Combining statistical translation techniques for cross-language information retrieval. In *Proceedings of COLING 2012.*

Ueffing, N., Haffari, G., & Sarkar, A. (2007). Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL).*

Utiyama, M., & Isahara, H. (2007). A Japanese-English patent parallel corpus. In *Proceedings of the eleventh machine translation summit (MT summit XI).*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In I. Guyon et al. (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 5998–6008). Curran Associates,

Inc.

Vogel, S., & Hewavitharana, S. (2011). Extracting parallel phrases from comparable data. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web.*

Wang, L., Tu, Z., Way, A., & Liu, Q. (2018). Learning to jointly translate and predict dropped pronouns with a shared reconstruction mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Wang, R., Utiyama, M., Liu, L., Chen, K., & Sumita, E. (2017). Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Wäschle, K., & Riezler, S. (2012a). Analyzing parallelism and domain similarities in the MAREC patent corpus. *Multidisciplinary Information Retrieval*, 12–27.

Wäschle, K., & Riezler, S. (2012b). Structural and topical dimensions in multi-task patent translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL).*

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*(3), 229–256.

Wołk, K., & Marasek, K. (2015). Unsupervised comparable corpora preparation and exploration for bi-lingual translation equivalents. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT).*

Xu, J., Weischedel, R., & Nguyen, C. (2001). Evaluating a probabilistic model for cross-lingual information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR).*

Xue, X., Jeon, J., & Croft, B. (2008). Retrieval models for question and answer archives. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR).*

Zaidan, O. F., & Callison-Burch, C. (2011a). The Arabic Online Commentary Dataset: An annotated dataset of informal Arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL).*

Zaidan, O. F., & Callison-Burch, C. (2011b). Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th annual meeting of the association for computational linguistics (ACL).*

Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. *CoRR*, *abs/1212.5701*. Retrieved from `http://arxiv.org/abs/1212.5701`

Zhang, J., Li, L., Way, A., & Liu, Q. (2016). Topic-informed neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers.*

Zhao, B., Eck, M., & Vogel, S. (2004). Language model adaptation for statistical machine translation with structured query models. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING).*

Zhou, Z.-H. (2017). A brief introduction to weakly supervised learning. *National Science Review*, *5*(1), 44–53.

Ziemski, M., Junczys-Dowmunt, M., & Pouliquen, B. (2016). The United Nations Parallel Corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC).*

Zuccon, G., Koopman, B., Bruza, P., & Azzopardi, L. (2015). Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th australasian document computing symposium.*

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Dr. Stefan Riezler, for his professional input and keen insights, his encouragement and support, his unrelenting enthusiasm, as well as his sharp criticism and high standards, throughout my time in his group. A profound thank you also goes to Prof. Dr. Michael Gertz for agreeing to act as second supervisor.

I would also like to acknowledge the members and former members of MLGroup. Whether cooperating on projects, attending conferences and thanksgivings together, plotting insubordination, or just having a quick chat, you have all been a source of inspiration to me and have made my (rather long) time at the institute worthwhile. A very special thank you to Carolin Lawrence who read most of this dissertation and provided invaluable input.

Finally, and most of all, it is hard to put into words the amount of support I have received from my family. Without you, the completion of this work would not have been possible.