Quantifying Cross-lingual Semantic Similarity for Natural Language Processing Applications

Dissertation zur Erlangung der Doktorwürde der Neuphilologischen Fakultät der
Ruprecht-Karls-Universität Heidelberg
vorgelegt von

Katharina Wäschle

1. Juni 2015

Erstgutachter:	Prof. Dr. Stefan Riezler Institut für Computerlinguistik, Universität Heidelberg
Zweitgutachterin:	PD Dr. Karin Haenelt Fraunhofer-Gesellschaft e.V.
Datum der Disputation:	21. Juli 2015

Abstract

Translation and cross-lingual access to information are key technologies in a global economy. Even though the quality of machine translation (MT) output is still far from the level of human translations, many real-world applications have emerged, for which MT can be employed. Machine translation supports human translators in computer-assisted translation (CAT), providing the opportunity to improve translation systems based on human interaction and feedback. Besides, many tasks that involve natural language processing operate in a cross-lingual setting, where there is no need for perfectly fluent translations and the transfer of meaning can be modeled by employing MT technology. This thesis describes cumulative work in the field of cross-lingual natural language processing in a user-oriented setting. A common denominator of the presented approaches is their anchoring in an alignment between texts in two different languages to quantify the similarity of their content.

In particular, this thesis proposes an approach to detect cross-lingual textual entailment – a first step towards cross-lingual content synchronization. We model cross-lingual similarity both based on alignment scores and full translations and combine the models in a statistical learning framework. We then study another application of the system: reduction of noise created by automatic sentence-alignment of parallel text. We present an analysis of the kind of noise present in our data, a method to filter the noise and experiments on its influence on MT quality. A further application for cross-lingual similarity measures is translation retrieval. Translation memory systems support human translators by retrieving similar text that has been translated previously, so-called fuzzy matches. These can be used as a foundation for new translations. We propose a novel approach to retrieve fuzzy matches: instead of comparing source input and the source side of a bilingual sentence pair, we measure similarity between source and a target match candidate directly, enabling the use of monolingual resources as translation memories. We evaluate our method by integrating the fuzzy matches into a statistical MT system and show that cross-lingual retrieval can improve translation quality. Finally, we suggest an online learning approach for document-level translation, which exploits the interactive CAT process. We refine a phrase-based statistical MT system on human feedback by aligning source input and human post-edit and integrating the information into the MT engine immediately. We show that an adaptive system outperforms a non-adaptive baseline.

Zusammenfassung

Sprachübergreifender Zugang zu Informationen ist eine Schlüsseltechnologie in einem globalen Wirtschaftssystem. Auch wenn die Qualität automatisch generierter Übersetzungen nicht das Niveau menschlicher Übersetzer erreicht, existieren Anwendungen, bei denen maschinelle Übersetzung (MT) erfolgreich zum Einsatz kommt. Im Bereich computergestützte Übersetzung (CAT) werden Übersetzer durch ein MT-System unterstützt, das Übersetzungsvorschläge generiert. Durch das Feedback der Nutzer kann die MT-Komponente stetig angepasst und verbessert werden. Neben der eigentlichen Übersetzung existieren Aufgabenbereiche, bei denen Texte sprachübergreifend verarbeitet werden. Anstatt hierbei komplette Übersetzungen als Zwischenrepräsentationen zu verwenden kann die inhaltliche Beziehung zwischen Text in verschiedenen Sprachen auch direkt modelliert werden. Die vorliegende Dissertation beschreibt kumulative Arbeit in verschiedenen Bereichen der anwendungsorientierten, cross-lingualen Sprachverarbeitung. Alle vorgestellten Modelle und Ansätze eint die Idee von sprachübergreifender Semantik: Auf der Basis eines Alignments von Quell- und Zielsprachentext können inhaltliche Ähnlichkeiten quantifiziert werden.

Im Einzelnen werden beschrieben: Ein Ansatz zur Erkennung von sprachübergreifendem *Textual Entailment*, mit dem Ziel, Inhalte, die in verschiedenen Sprachen vorliegen, zu synchronisieren. Dazu werden Ähnlichkeitsmaße basierend auf Wort-Alignment mit maschinellem Lernen kombiniert. Das Modell wird darüber hinaus eingesetzt, ein satzaligniertes, paralleles Korpus aufzubereiten, das bedingt durch die Automatisierung des Alignments fehlerhafte Zuordnungen enthält. Art und Quelle der Fehler werden analysiert und eine Methode vorgestellt, fehlerhafte Alignments aus den Daten herauszufiltern. Eine weitere Anwendung für cross-linguale semantischer Ähnlichkeitsmaße stellt *Translation Retrieval* dar. *Translation-Memory*-Systeme sind ein Werkzeug zur Unterstützung von Übersetzern. Diese gleichen zu übersetzende Sätze mit bereits übersetzten Segmenten ab; wenn ähnliche Sätze gefunden werden (sog. *Fuzzy Matches*), gibt das System deren Übersetzung aus, die dann zur Grundlage der neuen Übersetzung werden. Die vorliegende Arbeit stellt eine neuartige Methode vor, um *Fuzzy Matches* zu erhalten: Der Quellsatz wird mithilfe sprachübergreifender Ähnlichkeitsmaße direkt mit Sätzen in der Zielsprache verglichen. Dadurch können monolinguale Datensätze anstelle von bilingualen als *Translation Memory* eingesetzt werden. Wir führen eine Evaluierung dieser Methode durch, indem die *Fuzzy Matches* in ein MT-System integriert werden, und zeigen, dass diese die Übersetzungsqualität steigern. Zuletzt wird ein Ansatz vorgestellt, um ein MT-System *online*, d.h. satzweise während der Übersetzung eines Dokuments, anzupassen. Dabei wird der interaktive CAT-Übersetzungsprozess ausgenutzt und das von Übersetzern gegebene Feedback unmittelbar in die MT-Komponente integriert. Wir zeigen, dass ein solches selbstadaptierendes System einem statischen System überlegen ist.

Contents

CHAPTER 1

Introduction

The task of translating text from one language into another can be viewed as the search for a semantic equivalent of the source message in the target language (Newmark, 1988). *Machine translation* (MT) aims at performing this transfer task with minimal human intervention using computational resources. Early approaches to MT worked with manually crafted translation rules (Bar-Hillel, 1951), but the rapid increase in computational power during the 1990s and 2000s enabled the development of the purely empirical, data-driven *statistical machine translation* (SMT) paradigm (Brown et al., 1990, 1993). SMT became widespread after open-source implementations of decoders became available (Chiang et al., 2005; Koehn et al., 2007; Li et al., 2009; Dyer et al., 2010), which allowed researchers to quickly set up their own SMT system and experiments. Today, SMT is a large research area with numerous conferences, workshops and evaluation campaigns[1] dedicated to the topic and a rapidly evolving field. Still, the quality of machine translations is not such that the output is directly usable in real-world applications – apart from few domains with controlled language such as weather reports, for example – and the question remains, how MT can be advantageously employed (Kay, 1980, 1997b,a). A promising research direction is therefore the **convergence of SMT and human translation**. In today's global economy, localization is a large and growing industry and the demand for fast and high-quality translations, especially for technical documents, is on the rise (Sykes, 2009). Professional translators have been somewhat hesitant to adopt SMT into their workflow due to the high demands on translation quality. In recent years the

[1] For example WMT, EAMT, MT Summit, IWSLT, AMTA with IAMT and WPTP, NIST OpenMT, Translating and the Computer, MT Marathon and SSST as well as numerous workshops on special topics in MT at NLP conferences such as ACL, NAACL and Coling.

research community has shown a growing interest in advancing the integration of SMT and human translation. From EU projects to build open-source translation workbenches that include SMT engines (e.g. MateCat[2] and Casmacat[3]), real-time adaptive translation extensions for popular open-source decoders such as cdec[4] (Denkowski et al., 2014) and Moses[5] to start-ups successfully combining SMT, crowd sourcing and post-editing, such as Unbabel[6] (Chokkattu, 2014) or Lilt[7] (Abate, 2014; Green et al., 2014a,b), MT has started to focus on practical challenges. The majority of the mentioned systems have become publicly available only during the last two years.

A further research direction towards the real-world application of SMT technology is its deployment for other **cross-lingual natural language processing** (NLP) tasks, such as *cross-language information retrieval* (CLIR) (Nie, 2010), content synchronization (Negri et al., 2012) or detection of parallel documents, sentences or phrases (Resnik and Smith, 2003; Braune and Fraser, 2010; Munteanu and Marcu, 2006). In practice, many applications do not require perfectly fluent translations: Multilingual content synchronization can for example be applied to identify matching and mismatching content in a user-created multilingual encyclopedia (Monz et al., 2011). Different language versions of articles on the same topic in the encyclopedia evolve separately, since they are being edited by speakers of different languages. Even though divergences in content can partly be attributed to cultural shifts and a complete synchronization might not be sensible, identifying information that is missing in one language, could help extend the coverage of the encyclopedia in underrepresented languages. For the MediaWiki software that powers Wikipedia, for example, a translation workbench was recently introduced to encourage and support editors in translating articles into languages with sparse content (Laxström et al., 2015). An application for cross-lingual retrieval can be found in cross-language patent prior art search: In order to protect an invention in more than one country or region of the world, a separate patent has to be filed with several patent agencies in different languages. Before a patent application is drafted, applicants search for instances of so-called *prior art*, which consists of patents or documents that disclose intellectual property similar to the subject at hand. If central points of the invention have been laid out before, the patent application is likely to fail. Since drafting and applying for a patent is a costly process, making sure that all relevant prior art has been found is a crucial step for applicants. If a company applies for the protection of an invention, for which a patent has been granted in language L_1, in another country with language L_2, CLIR can be used to query the body of patents in language L_2 to search for prior art specific to this region using the original document in language L_1 as a query. The actual translation of the results, if relevant documents are found, can then

[2] http://www.matecat.com/
[3] http://www.casmacat.eu/
[4] http://www.cs.cmu.edu/~mdenkows/cdec-realtime.html
[5] http://statmt.org/moses/?n=Advanced.Incremental and http://statmt.org/moses/?n=Advanced.CacheBased
[6] https://unbabel.com/
[7] http://lilt.com/

be assigned to a human translator. Further applications are for example cross-lingual image search based on textual descriptions of the images (Etzioni et al., 2007) or cross-language plagiarism detection (Potthast et al., 2011). Usually, approaches to cross-lingual NLP tasks propose to model the transfer of meaning between different languages without employing the full translation pipeline (Resnik and Smith, 2003; Munteanu and Marcu, 2005; De Souza et al., 2013). Many approaches rely heavily on word alignments and lexical translation probabilities estimated on parallel data to quantify the pairwise similarity of text in two languages.

This thesis assembles contributions towards different tasks from the field of cross-lingual natural language processing and towards the integration of human translation and SMT. We describe approaches and experiments using different ways of quantifying and modeling cross-lingual similarity based on word alignments estimated on parallel data. We start by considering the notion of cross-lingual semantic relatedness and design a solution for the *cross-lingual textual entailment* (CLTE) task, suggested by Mehdad et al. (2010), as a first step towards tackling the problem of content synchronization across different languages. Our approach combines alignment and translation features and makes use of various steps of the SMT pipeline, ranging from word alignment to MT evaluation metrics. We describe our feature set in detail and present extensive experiments, where we identify the most useful features for different language pairs. Furthermore, we explore the problem from a machine learning perspective and compare different ways of formulating the problem for a learner.

We then apply the cross-lingual similarity features designed for detecting cross-lingual textual entailment to a related task from the area of SMT: detecting and removing noise from an automatically aligned noisy parallel corpus. Excluding non-parallel or only semi-parallel phrases from training can improve SMT quality (Cui et al., 2013). This task can be formulated as discarding pairs of text that are not semantically equivalent, which is a subtask of recognizing cross-lingual textual entailment. In the course of this work, we extend and improve the work described in Wäschle (2011) and extract a parallel corpus from patent documents for the language pairs French-English and German-French. We annotate a sample of 500 sentence pairs from the corpus with equivalence judgments that quantify how parallel a pair is, after conducting preliminary experiments to obtain a notion of parallelism. We report experiments with different levels of filtering and on learning a similarity threshold.

In the next step, we investigate cross-lingual NLP for a user-oriented task: *translation memories* (TM) are a widely used tool in the localization industry. TMs store previously translated segments of text and can be queried with the sentence currently under translation. If a similar sentence – similarity is based on a comparison of the source sides – has been processed before, its corresponding target is passed to the human translator for post-editing. These so-called *fuzzy matches* are in general not a perfect translation of the given sentence, but can help to reduce translation time and effort. Furthermore, they are fluent output so the post-editor is not forced to correct errors made by an SMT system. Usually, string similarity

is used to find good fuzzy matches. We strive to go beyond the surface and enable the use of monolingual resources for retrieving fuzzy matches at the same time. While previous work only searches for matches in the source language by comparing the input sentence to the source side of a bilingual sentence pair, we use a cross-language comparison to find fuzzy matches that are only available in the target language. We employ methods from the field of cross-language information retrieval to determine good target matches using SMT to produce target language representations of the query. In order to efficiently query large monolingual corpora in the order of tens of millions of sentences we propose employing *locality sensitive hashing* (LSH) as a pre-retrieval step to narrow down match candidates. We evaluate our retrieval by integrating the fuzzy matches into an SMT system. Our model re-ranks an n-best list of SMT hypotheses with regard to their similarity to a fuzzy match. We report results for cross-language translation memory retrieval that are comparable to source-side retrieval and show its application in a domain adaptation scenario, where additional monolingual data is available.

From the notion of cross-lingual similarity we turn to the question, whether and how machine translation can benefit from taking a larger context into account. Due to technical limitations, SMT systems usually translate segments of text in isolation without looking beyond sentence boundaries. This way, valuable global context information is lost and consistency is impossible to guarantee. Even if an SMT system is highly consistent in its translation choices (Carpuat and Simard, 2012), this consequently leads to them being also highly consistent in repeating errors when translating a text. This led us to consider a solution for MT consistency in an environment, where feedback on previous translations of sentences from the same text are available. This is the case in *computer-aided* or *computer-assisted translation* (CAT), where human translators post-edit hypotheses output by an MT system. We develop an online learning protocol using cache-based document-specific language and translation models, to show that SMT system and human user can mutually benefit from feedback. The user corrects errors made by the SMT system which in turn can help the user maintain translation consistency by suggesting recent translations of the same phrases. To extract information from user post-edits, a constrained search technique is used. We implement the protocol for the Moses SMT toolkit and report experiments on data, for which real-world user post-edits are available.

1.1 Scientific Contributions

This thesis cumulatively contributes the following approaches and data sets. All results, except for the noise filtering experiments, have been published in peer-reviewed research papers and the data sets have been made publicly available under a Creative Commons[8] license. In the

[8] http://creativecommons.org/licenses/by-nc-sa/3.0/

case of papers published with co-authors other than the author's supervisor, only the author's contributions towards the joint work is described in this thesis, except when explicitly noted.

Approaches

- A stand-alone system for recognizing cross-lingual textual entailment that took first place at the CLTE challenge at SemEval 2012[9] (Negri et al., 2012). An in-depth evaluation of the features and machine learning methods used by the final system (Wäschle and Fendrich, 2012).
- A study of noise in an automatically aligned corpus and experiments of the influence of the noise on SMT quality.
- A cross-lingual translation retrieval approach evaluated in a combined SMT and TM system, that can handle a large TM database with millions of sentences and yields significant improvements over the baseline using fuzzy matches to bias the SMT component. An application of the cross-lingual target fuzzy match retrieval technique to the task of domain adaptation with monolingual resources. An evaluation of different CLIR techniques for choosing good fuzzy matches given a query in the source language and a corpus in the target language (Wäschle and Riezler, 2015).
- An easy-to-implement technique for integrating feedback into the phrase-based Moses SMT decoder at run-time by adapting models online in a CAT environment that achieves improvements of up to 3 BLEU points over a static baseline on repetitive documents (Wäschle et al., 2013; Bertoldi et al., 2014).

Data

- A large amount of parallel data from the intellectual property (IP) domain for two language pairs, French-German (5.1 million sentence pairs) and English-French (18.8 million sentence pairs), accompanied by metadata (Wäschle and Riezler, 2012). The data has been integrated into the PatTR data set[10] and has to date been used for example in the Ninth Workshop on Statistical Machine Translation (WMT 2014)[11] (Bojar et al., 2014), for online adaptation in the MateCat EU project (Bertoldi et al., 2014), and for multilingual information retrieval in the medical domain in the Khreshmoi[12] EU project (Pecina et al., 2014).

[9] See http://www.cs.york.ac.uk/semeval-2012/task8/.
[10] Available from http://dx.doi.org/10.11588/data/10002.
[11] See http://www.statmt.org/wmt14/medical-task/.
[12] http://www.khresmoi.eu/

- A hand-annotated data set, created by two non-expert annotators and one expert juror, of 500 French-English sentence pairs from the data set described above, with two annotation levels quantifying cross-lingual semantic similarity/level of parallelism.

The thesis is structured as follows. First, we provide some background on statistical machine translation (Chapter 2), since all introduced models make use of (parts of) the SMT pipeline, and related topics from computer-assisted translation. Chapter 3 describes a system for recognizing cross-lingual textual entailment entered at SemEval 12, followed by the application of the features designed for this task to filtering a parallel corpus for SMT training (Chapter 4). Chapter 5 introduces cross-lingual fuzzy match search and describes an efficient implementation as well as an evaluation of the impact of integrating TM matches into SMT. We investigate taking SMT beyond the sentence level in Chapter 6 and present an online learning approach for translation consistency in a computer-assisted translation framework. Related work is discussed preceding each individual chapter; experiments and results are presented separately for each approach. Chapter 7 sums up findings and contributions.

CHAPTER 2

Background: Statistical Machine Translation and Computer-assisted Translation

In this chapter we aim to offer a short introduction to statistical machine translation and give an overview of the state-of-the-art techniques and tools, which are used in this thesis. Furthermore, we provide a brief overview over the field of computer-assisted translation and MT post-editing, which promises practical use for SMT systems beyond narrow domains. We establish basic terminology for both fields in the beginning.

2.1 Terminology

The foundation of parameter estimation for statistical translation models is *parallel text* or *bitext* and refers to pair-wise aligned sentences that are mutual translations of each other. Parallel corpora might contain a fraction of sentence pairs that are not actually translations of each other; depending on the amount of non-parallel data, corpora are characterized as *noisy parallel* or *comparable*. The language we translate into is called the *target* language, the language we translate from the *source* language. For historical reasons, a sentence in the target language is denoted by $\mathbf{e}$ and a sentence in the source language by $\mathbf{f}$[1]. We refer to the translation generated by an MT system as the *hypothesis* or *system output* and to the human translation used to evaluate as the *reference (translation)*. Finding the optimal parameters for the log-linear model (see Section 2.2.6) with discriminative training is referred to as *tuning*.

[1] The first SMT systems were built for translation from French into English.

In the terminology of computer-assisted translation, a translation unit is called a *segment*. A segment might correspond to a sentence or a headline, but also sub-sentential fragments[2] or whole paragraphs are possible, depending on the application. Editing a translation output by an MT system or a translation memory is referred to as *post-editing*. Post-editing can be performed both by a human (translator) and by automated systems (Simard et al., 2007).

2.2 Statistical Machine Translation

The task of statistical machine translation has been defined as the search for the best derivation $\hat{\mathbf{e}}$ in a hypothesis space[3] $\mathbf{H}$ given a source string $\mathbf{f}$, according to the conditional probability distribution $p(\mathbf{e}|\mathbf{f})$ (Brown et al., 1993).

$$\hat{\mathbf{e}} = \underset{\mathbf{e}\in\mathbf{H}}{\operatorname{argmax}}\, p(\mathbf{e}|\mathbf{f})$$

By applying Bayes' rule this probability can be decomposed.

$$\hat{\mathbf{e}} = \underset{\mathbf{e}\in\mathbf{H}}{\operatorname{argmax}}\, p(\mathbf{f}|\mathbf{e}) \times p(\mathbf{e})$$

An interpretation of this model is as an instance of a noisy channel, where the intuition is that the original message $\mathbf{e}$ has been corrupted during a transmission, resulting in the observed message $\mathbf{f}$. The task is to find the most likely original message given the observation. The term $p(\mathbf{e})$ can be seen as a target *language model* while the conditional term $p(\mathbf{f}|\mathbf{e})$ corresponds to a *translation model*. The parameters of the language model can be estimated on monolingual data, while the parameters of the translation model are learned from bitext. The search for the best $\hat{\mathbf{e}}$ given these parameters is called *decoding*. In practice, decoders use heuristics to limit the hypothesis space and make the search feasible.

2.2.1 Sentence Alignment

Texts translated by humans are the training examples, from which statistical machine translation systems can learn. Documents are usually translated as a whole, resulting in parallel data at the document, paragraph or section level. To make the data processable statistical machine translation systems, it is necessary to break it down into smaller units and align text at the sentence level, which is an unsupervised task. The first models for sentence alignment

[2] This can be the case when translating long sentences with several subordinate clauses as appear, for example, in patent claims, which can feature a high two figure number of embedded clauses.

[3] In theory, this is the space of all possible sentences in the target language.

were based on sentence length ratio (Brown et al., 1991; Gale and Church, 1993); later approaches added lexical models estimated on a first length-based pass over the data (Moore, 2002). Correspondences between sentences are not necessarily one-to-one, so state-of-the-art models allow one-to-many and many-to-one alignments (Braune and Fraser, 2010).

2.2.2 Word Alignment

Translation model parameters are estimated on the word level. Deriving word alignments (for an example see Figure 2.1) between parallel sentences is an unsupervised task. The most widely used models for word alignment are generative models that seek to maximize the likelihood of the data, most notably the IBM models (Brown et al., 1990), named after the company where they were developed. The baseline model is IBM Model 1, which introduces the idea of alignment variables $a : j \rightarrow i$ to model the conditional probability[4] of a sentence $\mathbf{e}$ (a sequence of tokens $e_j, j \in 1, ..., |\mathbf{e}|$) given sentence $\mathbf{f}$ (a sequence of tokens $f_i, i \in 1, ..., |\mathbf{f}|$).

$$p(\mathbf{e}, a|\mathbf{f}) = \prod_{j=1}^{|\mathbf{e}|} t(e_j|f_{a(j)})$$

The parameters of the model are the word translation probabilities $t(e|f)$, i.e. the probability that token e is the translation of token f. To estimate these parameters we need the word alignments $a : j \rightarrow i$; however, the alignments are what we want to derive in the first place. To solve this problem, the expectation-maximization (EM) algorithm (Dempster et al., 1977) can be applied in the following way. Assume an initial (uniform) distribution of word translation probabilities, (1) estimate probabilities for all alignments based on this distribution, (2) collect new counts for the word translation probabilities from the data and repeat steps (1) and (2) in an alternating fashion. After a few iterations, the probabilities will converge due to the latent co-occurrence information, i.e. which words in the source language co-occur frequently with which words in the target language.

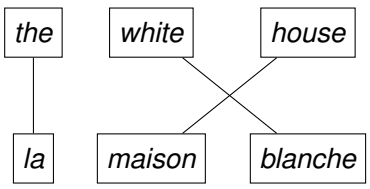

Figure 2.1: Word alignment.

IBM Model 2 adds a *reordering* parameter corresponding to a locality bias, model 3 adds *fer-*

[4] Formally, a normalization constant has to be added in order to assure that $p(\mathbf{e}, a|\mathbf{f})$ is a probability distribution, which we left out for reasons of comprehensibility.

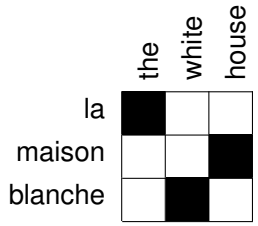

Figure 2.2: Word alignment matrix.

tility, i.e. a model for how many words in the target a source word can generate and model 4 relative reordering. Subsequent models are initialized with the probabilities of the previous, simpler models. The estimation procedure results in a lexicon of word translation probabilities and a set of word alignments that link a source word to one (or more) target tokens. It is possible for words to remain unaligned. Word alignments can be visualized as points in an alignment matrix (Figure 2.2). Alignments and probabilities depend on the translation direction. In practice, to derive a stable word alignment for a sentence pair, EM is therefore run in both translation directions and the resulting alignments are symmetrized. Different strategies are possible for this process,e. g. the *intersection* or the *union* of the sets of alignment links, or heuristically expanding the alignment (Koehn et al., 2003). A variety of tools exist for word alignment that implement the IBM models. Some of the most widely used are GIZA++ (Och and Ney, 2003), the BerkelyAligner (Liang et al., 2006b) and fast_align (Dyer et al., 2013).

2.2.3 Translation Models and Decoding

Several strategies have been proposed for decoding. Although the estimation of translation models is word-based, *phrase-based translation* (Koehn et al., 2003) emerged as a way to include a larger context in the translation process and to account for the fact that correspondences between words are often not one-to-one. Parallel phrases can be extracted from word alignments. Phrase pairs containing at least one alignment point and within which no word is aligned to a word outside of the phrase, are called *consistent* phrase pairs. The phrase pairs that are consistent with the word alignment shown in Figure 2.2 are listed in Figure 2.3.

$m = 1$	*la* / *the*
	maison / *house*
	blanche / *white*
$m = 2$	*maison blanche* / *white house*
$m = 3$	*la maison blanche* / *the white house*

Figure 2.3: Phrase pairs of length m that are consistent with the word alignment in Figure 2.2.

X → *la maison X* | *the X house* X → *white* | *blanche*

Figure 2.4: Two hierarchical phrases extracted from the word alignment matrix in Figure 2.2. In addition to the initial phrase pairs pictured in Figure 2.3, which correspond to lexical rules such as X → *white* | *blanche*, rules with gaps can be generated by replacing sub-phrases with the non-terminal symbol X. In practice, the number of resulting rules has to be reduced by applying heuristics – e.g. limiting to the number of non-terminal symbols per rule – to reduce the complexity of the model (Chiang, 2007).

During decoding, target hypotheses are constructed from left to right using phrase pairs extracted from the training data. Since the hypothesis space becomes unfeasibly large, pruning techniques such as beam search (Tillmann and Ney, 2003) are applied to reduce the search space. A popular implementation of the phrase-based approach to statistical translation is the Moses decoder (Koehn et al., 2007).

A *hierarchical* extension of the phrase-based paradigm (Chiang, 2007) allows discontinuous phrases that contain gaps by modeling correspondences as translation rules in the form of a context-free grammar (CFG). The resulting grammar is *synchronous* (SCFG), meaning that rules contain both source and target side in the form $X \rightarrow house|maison$ (see also Figure 2.4). Only one generic non-terminal symbol X is used, so grammar rules do not necessarily conform to syntactical components of a sentence. Decoding with an SCFG corresponds to parsing the source side, which automatically generates a target side parse forest. The best derivation is selected according to features that operate on the grammar rules. To reduce the decoding complexity, especially after adding language model scores, cube pruning is applied (Chiang, 2007). In this thesis we employ the cdec decoder (Dyer et al., 2010) for experiments involving hierarchical phrase-based translation models.

2.2.4 Additional Generative Models: Language and Reordering Model

One view on the language model that is used in statistical machine translation is the noisy-channel intuition described above. Other important factors for including a language model are context and disambiguation. While phrase-based translation models take a larger context into account than basic word-based models, the context is still fairly limited for reasons of generalization and translations are assembled using a number of separate phrases. Adding a high n-gram language model promotes fluency of the output and considers a larger context that spans several phrases. A language model approximates the joint probability of a sequence of target tokens $\mathbf{e} = (e_1, ..., e_m)$.

$$p(e_1, ..., e_m) = \prod_{i=1}^{m} p(e_i|p(e_1), ..., p(e_{i-1}))$$

To make the model generalize, the context history, that is the number of previous words the model takes into account when choosing the next word, is restricted to n tokens.

$$p(e_1, ..., e_m) \approx \prod_{i=1}^{m} p(e_i | p(e_{i-(n-1)}), ..., p(e_{i-1}))$$

The parameters of the model are estimated on a (preferably large) monolingual corpus in the target language with maximum likelihood estimation.

Reordering models add information about the different word order in source and target language. The basic model is distance-based with a cost linear to the distance of the reordering. More refined models are lexicalized (Tillmann, 2004). Hierarchical phrase-based models encode reordering information inherently in the grammar.

2.2.5 Evaluation Metrics

Automated evaluation of machine translation output is a research direction of its own, since deciding, if a translation is good or bad, is a task nearly as challenging as MT itself. Human evaluation is costly and therefore seldom done. The following evaluation metrics are used to measure the quality of a translation hypothesis given a reference translation.

BLEU (Papineni et al., 2002)

BLEU (*Bilingual Evaluation Understudy*) is the most widely used measure in MT due to its good correlation with human judgments and its inexpensive calculation. The score combines modified n-gram precision and a brevity penalty, since precision alone favors shorter translations.

$$\text{score} = BP \times \sum_{i=1}^{4} P_n$$

The modified n-gram precision P_n clips every n-gram count to the maximum number of times the n-gram appears in the reference. The n-gram counts are calculated over the entire corpus, matches are counted in each sentence. The brevity penalty BP is defined as

$$BP = \begin{cases} \exp(1 - \frac{|reference|}{|h|}) & \text{if } |h| \leq |reference| \\ 1 & \text{otherwise} \end{cases}$$

and is also computed over the whole corpus.

Meteor (Denkowski and Lavie, 2011)

Meteor (*Metric for Evaluation of Translation with Explicit ORdering*) is based on a word alignment between system output and reference translation. Words are matched at four different similarity levels, exact match, stem match, synonym and paraphrase. The harmonic mean of weighted alignment precision P and recall R is combined with a fragmentation penalty pn_{frag} into a weighted score.

$$\text{score} = (1 - \gamma \times \text{pn}_{\text{frag}}{}^{\beta}) \times \frac{P \times R}{\alpha \times P + (1 - \alpha) \times R}$$

The fragmentation penalty is defined as the number of contiguous sequences of matched words (*chunks*) divided by the total number of matches.

$$\text{pn}_{\text{frag}} = \frac{|chunks|}{|matches|}$$

Meteor comes with pre-trained weights α, β and γ and linguistic resources such as synonym and paraphrase tables for several languages. Meteor is more sensitive to the transfer in meaning than BLEU, but also more costly to compute. Besides, not all resources are available in every language. Unlike BLEU, the metric is sentence based.

TER (Snover et al., 2006)

Translation Edit Rate[5] (TER) was designed with post-editing effort in mind. It measures the minimum number of edits needed to generate the reference from the SMT output, normalized by the number of reference words.

$$\text{score} = \frac{|edits(h, reference)|}{|reference|}$$

All edit operations have equal cost: insertion, deletion and substitution of single tokens and shifts of sequences of tokens. Due to its foundations in edit distance, TER tends to favor shorter translations. TER is sentence based and is predominantly used in the field of post-editing and interactive translation.

2.2.6 Log-linear Model and Discriminative Parameter Training

In practice, the generative SMT models – translation, language and reordering model(s) – are combined into a global log-linear model (Och and Ney, 2002) as feature functions h_k with

[5] Sometimes referred to as *Translation Error Rate*, though this is misleading, since it is not an error rate.

corresponding weights λ_k. The best derivation $\hat{\mathbf{e}}$ under this model is the one that maximizes the combined model score.

$$\hat{\mathbf{e}} = \underset{\mathbf{e} \in \mathbf{H}}{\operatorname{argmax}} \exp\left(\sum_k \lambda_k h_k(\mathbf{e}, \mathbf{f})\right)$$

With $k = 2$, $h_1 = \log(p(\mathbf{e}))$, $h_2 = \log(p(\mathbf{f}|\mathbf{e}))$ and uniform weights we get the noisy-channel model described above. The advantage of this extended model is that its parameters can be trained in a discriminative fashion, balancing the contribution of the individual models. This tuning step is carried out on a small amount of development data, usually 1,000-2,000 sentence pairs and optimized with regard to one (or more) MT evaluation metrics. The model can be enriched with additional feature functions that model different aspects of the translation process. In the following, we discuss some popular methods for discriminative training.

Minimum Error-Rate Training (MERT) (Och, 2003)

MERT was the first approach that aimed to optimize the non-convex error metrics used in SMT directly. MERT operates on *n*-best lists, i.e. a ranking of the top *n* derivations output by the decoder. The weights of the log-linear model are optimized using line search: the model score of a translation hypothesis corresponds to a line, when only one weight is regarded at a time. Several iterations of the optimization are performed and the training set is repeatedly re-translated using the adjusted weights. Since the search is heuristic with no guaranteed optimum, several instances of the algorithm are started with different initial values and the resulting weights averaged to obtain the final weight vector. Kumar et al. (2009) extended MERT to work on the full search space generated by hierarchical MT systems.

Margin Infused Relaxed Algorithm (MIRA) (Watanabe et al., 2007)

Since MERT is able to optimize only a small number parameters (in double figures) reliably, alternatives have been proposed to handle a larger number of features. Enriching the translation model with sparse features has been done for example by Chiang et al. (2009). The Margin Infused Relaxed Algorithm (MIRA) learns in an online fashion, meaning that it updates parameters iteratively after each training example. MIRA is therefore able to operate on a large tuning set and optimize a huge number of features. Since sentence-wise metrics are required to do online updates, approximations of BLEU are used.

Pairwise Ranking Optimization (PRO) (Hopkins and May, 2011)

PRO approaches the SMT tuning problem as pairwise ranking by sampling pairs of candidate translation from an *n*-best list: a pair is labeled as a positive training example, if model score

and gold standard score agree in their ranking and negative, if not. This reduces optimization to a binary classification problem. As with the other methods, several iterations are performed and the *n*-best lists are regenerated after each iteration using the modified parameters.

2.3 Computer-assisted Translation

The growing demand for translation in a global economy has resulted in a need for speedy and efficient translation processes (Sykes, 2009). One way to increase productivity and make translation faster and cheaper is to automatize tasks, that can be reliably performed by a machine, have to be carried out repeatedly and might be tedious to human translators. *Computer-aided* or *computer-assisted translation* (CAT) is an instance of human-machine interaction that transforms translation into an interactive process, with the goal to reduce translation time and effort. The original idea is attributed to Kay (1980) (Hutchins, 1998). CAT tools offer a wide range of assistance to a human translator: Electronic dictionaries, spell and grammar checking, terminology management, translation memory databases, visualization of alignments and project management functionalities. Some tools also offer MT components, but in general, the localization industry has been hesitant to adopt machine translation due to the high demands on the quality of the output. For professional translators, time spent reading and discarding bad MT output is money lost, although studies suggests that this is never actually the case (Krings, 2001) – the field of MT *quality estimation* researches possible automated solutions for this problem (Ueffing et al., 2003; Blatz et al., 2004; Specia et al., 2009). The core component of the CAT user interface is an editor, in which the human translator composes his or her translation, assisted by one or more by the above-mentioned tools. Several aspects of CAT that are intertwined with machine translation are introduced in the following sections.

2.3.1 Translation Memory

A *translation memory* (TM) is a computational tool used by professional translators to speed up translation of repetitive texts (Christensen and Schjoldager, 2010). At its core is a database, in which source and target of previously translated segments of text are stored. Translation memories are capable of retrieving not only exact, but also partial matches, where only a certain percentage of source words overlap with the query, called *fuzzy matches*. The goodness of a match is generally measured with an edit-distance based *fuzzy match score* (Sikes, 2007). A CAT tool presents possible matches found in the database to a user, if the match is considered similar enough to the current source sentence according to some adjustable threshold. Even if the presented target sentence is not a perfect translation, a fuzzy match can be a helpful starting point for the translation of the current sentence and considerably reduce translation time and effort. Furthermore, a translation memory can help with translation

consistency and terminology control. In contrast to SMT, TM tools are widely used in the translation industry, since the results presented to the translator are fluent and a similarity threshold can be easily set to avoid spending time on reading translation suggestions that might not be usable. Translation memories are especially helpful for the translation of texts from repetitive domains, such as technical documentation or software localization, where vocabulary and expression are limited. Note, that the functionality of a translation memory combined with a human translator are similar to *example-based* machine translation (EBMT) (Nagao, 1984).

2.3.2 MT Post-Editing

With the availability of open-source CAT workbenches such as MateCat and the option to distribute translation tasks via crowd-sourcing (Jehl et al., 2012), the integration of human feedback into existing translation models has become a promising research direction in machine translation. In contrast to CAT, the goal is to improve MT performance with human-aided machine translation. This is made possible by keeping track of *post-edits*[6] of machine translations performed by human translators: Source sentences are presented to the user together with a translation hypothesis produced by an MT system. Users are asked to edit the MT output until it constitutes a fluent and correct translation of the input sentence. Performing post-editing instead of translating from scratch has been shown to reduce translation time and effort (Green et al., 2013). A further advantage is the fact that MT *post-edits* are very valuable for improving MT systems, since they are naturally closer to the MT system output than independently produced human references. Error metrics have been proposed based on of the number of edits a human translator performs on MT output (Snover et al., 2006); however, in practice, evaluation of MT using trained human translators is still costly and seldom done. Still, even learning from weak feedback, e.g. incomplete post-edits or post-edits done by non-native speakers might yet prove to be beneficial for MT. Post-edits can be recorded in varying detail, from only saving the final translation to recording a user's every keystroke.

2.3.3 Interactive Translation

Although MT post-editing with a subsequent integration of the feedback into an SMT system is an interactive translation process, the term *interactive machine translation* (IMT) refers in particular to systems that suggest partial translations to the user while the user is typing, based on the input so far (Nepveu et al., 2004; Ortiz-Martínez et al., 2010; López-Salcedo et al., 2012). The prediction is updated at every step the user takes while inputting the translation of a segment, with the models operating at word or even character level. This process has been shown to be beneficial in terms of translation effort (Barrachina et al., 2009).

[6] The prefix *post* references the fact that the editing takes place *after* automatic translation.

CHAPTER 3

Detecting Cross-lingual Textual Entailment

Cross-lingual textual entailment (CLTE) (Mehdad et al., 2010) has been proposed as an extension of textual entailment (TE) (Dagan and Glickman, 2004; Dagan et al., 2013). Recognizing entailment is defined as deciding whether the truth of a hypothesis H can be derived from a text T with semantic inference. CLTE adds a cross-lingual dimension to the problem by considering sentence pairs, where T and H are encoded in different languages, i.e. an English text E and a foreign hypothesis F or vice versa. A system that recognizes entailment across different languages might be applied to the task of content synchronization, for example the automatic synchronization of articles in a multilingual encyclopedia (most notably Wikipedia[1]). Articles on the same subject are written by several authors in different languages. As a result, information in one version of the article might not be present in other languages and can be used to extend them using MT or CAT. A first step towards achieving this goal is to identify overlapping parts of information on the sentence level as well as to classify the type of relation between non-matching parts. To this end, the Cross-lingual Textual Entailment for Content Synchronization task was introduced at SemEval-2012[2] (Negri et al., 2012). The models and systems described in this chapter were developed on the training data provided by the task organizers and our final system configuration was entered in the competition. The system ranked first, yielding the best overall score for three out of four language pairs.

We frame the task of recognizing CLTE as a statistical learning problem on a space of dense features modeling cross-lingual similarity. Statistical machine translation methods and tools

[1] http://www.wikipedia.org/

[2] http://www.cs.york.ac.uk/semeval-2012/task8/

are employed to derive features. The final system combines both cross-lingual and monolingual features: Cross-lingual similarity is based on word alignments. Monolingual similarity is assessed on a pivot translation of one text into the language of the other using established features used to detect monolingual textual entailment, such as distance and bag-of-words overlap. We rely on the learner to combine and identify the important features for solving this task. We pursued several goals during development.

- Address the task without deep linguistic processing or resources to make it easily portable across languages using only parallel data.
- Identify important features to draw conclusions about the role of SMT and evaluate how methods from SMT can be exploited to solve a cross-lingual semantic task.
- Develop methods to quantify cross-lingual similarity directly and compare them to a pivot-based approach, where the task is reduced to a monolingual scenario using the full SMT pipeline.

We argue that a system for recognizing cross-lingual entailment can benefit from integrating SMT methods, such as measures based on word alignments between E and F, as opposed to only employing SMT as a black box to gain a pivot translation, which is then passed to a monolingual system for recognizing entailment. The large amounts of bilingual parallel data used to train an SMT system naturally model synonymy and paraphrasing across languages – similar resources for monolingual TE are magnitudes smaller and hard to come by – and suggest a considerable advantage by adding cross-lingual information. We also found that the problem of cross-lingual textual entailment is closely related to the problem of finding parallel sentences for extending SMT resources, for which many techniques exist. The success of our system in the competition shows, that SMT is a good fit for solving a task from the field of cross-lingual semantics, supporting the notion of SMT as a semantic task. A detailed analysis shows, that we achieve good results using both cross-lingual or pivot-based monolingual features exclusively, but the best system is a combination of both. A further key aspect of our approach is the assessment of different learning strategies.

3.1 CLTE Task

The CLTE task emerged from the monolingual Recognizing Textual Entailment (RTE) task (Dagan et al., 2006a); however, the view on entailment is actually quite different. In classic RTE, the text T is typically longer than the hypothesis H (for an example see Figure 3.1). The entailment problem is binary: TRUE (H is entailed by T) and FALSE. To establish, whether the truth of H follows from T, inference and therefore a deep semantic analysis of both text and

hypothesis is necessary. In Figure 3.2, the hypothesis is completely contained in the text as a substring, so a superficial analysis, such as edit distance or *n*-gram overlap, would likely produce the result that *H* is entailed by *T*. In order to recognize that this does not correspond to the actual structure and subsequently meaning of the sentence, a deeper analysis is inevitable. Subsequent instantiations of the RTE challenge went on to add CONTRADICTION as entailment option, furthering the need to use methods for semantic inference to solve the task.

text	*CBS newsman Harry Reasoner is returning to his Iowa hometown to get married Saturday.*
hypothesis	*CBS newsman Harry Reasoner was born in Iowa.*
entailment	TRUE

Figure 3.1: An example for textual entailment from the first Pascal RTE challenge (Dagan et al., 2006b). To identify the entailment, a system needs to derive that *Harry Reasoner was born in Iowa* is entailed by *his Iowa hometown*.

text	*The National Institute for Psychobiology in Israel was established in May 1971 as the Israel Center for Psychobiology by Prof. Joel.*
hypothesis	*Israel was established in May 1971.*
entailment	FALSE

Figure 3.2: An example for no entailment from the the first Pascal RTE challenge (Dagan et al., 2006b). Although the text contains the complete hypothesis as a string, the hypothesis is not entailed by the text.

In contrast, CLTE considers sentence pairs *E* and *F* in different languages, which are of approximately the same length. The entailment is predicted in both directions, i.e whether *E* entails *F* and *F* entails *E*. Entailment is understood in terms of information that has been added or is missing from one or both sentences. The data set features no sentences that do not share at least part of their content, nor does it contain contradicting statements. Mehdad et al. (2010) define four entailment relations.

BIDIRECTIONAL entailment both sentences have similar content, i.e. *E* and *F* are semantically equivalent

FORWARD entailment the source language sentence *E* contains additional information that is not expressed in the target language sentence *F*, i.e. *E entails F*

BACKWARD entailment the target language sentence *F* contains additional information that is not expressed in the source language sentence *E*, i.e. *F entails E*

NO ENTAILMENT both source language sentence E and target language sentence F contain information that is not expressed in the respective other language

The SemEval-2012 CLTE task featured a training set of entailment instances in four language pairs[3]. Negri et al. (2011) describe the process of constructing the training and test corpus in great detail. Starting from a parallel corpus of English, Spanish, French, German and Italian text, monolingual entailment pairs were created by paraphrasing an English sentence E and deleting or adding information to construct a modified sentence E'. The modified sentence was then paired with the translation of the original sentence in different languages[4] thus creating a bilingual entailment pair. Two examples are given in Figures 3.3 and 3.4. Due to their origin, we concluded that out model should be able to capture the following key aspects between sentences E and F.

- Paraphrases and synonymy to identify semantic equivalence.
- Phrases that have no corresponding phrase in the other sentence, indicating missing (respectively, additional) information.

We hypothesize that – differing from the monolingual RTE challenges – the CLTE task does not require deep semantic processing, i.e. inference to be solved, but can instead be viewed as an alignment problem. As a solution we propose mostly alignment-based features that identify matching and mis-matching parts in the foreign language, leaving parts, where information is missing or has been added, unaligned. For this reason our methods draw from the related problem of identifying parallel data for SMT training, as laid out in Sections 3.2 and 3.3, by quantifying cross-lingual similarity.

F	*Ban Ki-Moon ist der achte und derzeitige Generalsekretär der Vereinigten Nationen,* ***seitdem er im Jahr 2007 das Amt von Kofi Annan übernahm****.*
E	*Ban Ki-moon became the 8th and current Secretary General of the United Nations.*
entailment	FORWARD

Figure 3.3: Examples for FORWARD entailment from the SemEval-2012 CLTE data set. The main clauses of both sentences are parallel, while text 1 contains additional information stated in a subordinate clause.

[3] Spanish-English (**es-en**), Italian-English (**it-en**), French-English (**fr-en**) and German-English (**de-en**).
[4] We generally refer to the non-English language sentence as F.

F	*Die Deepwater-Horizon-Ölpest war eine Ölpest **im Golf von Mexiko**, die im Jahr 2010 drei Monate lang anhielt.*
E	*In Deepwater Horizon accident, the oil poured for almost three months of the year 2010, **followed by an explosion**.*
entailment	NO ENTAILMENT

Figure 3.4: Example for NO ENTAILMENT relation from the SemEval-2012 CLTE data set: both texts contain additional information that cannot be found in the other text, i.e. a location (*im Golf von Mexiko*) in text 1 and details on the spill (*followed by an explosion*) in text 2.

3.2 Related Work

The majority of research on entailment has been conducted on monolingual TE in the context of the yearly Recognizing Textual Entailment (RTE) challenge (Dagan et al., 2006a; Bar Haim et al., 2006; Giampiccolo et al., 2007, 2008; Bentivogli et al., 2009, 2011). We include established monolingual features in our approach, such as alignment scores (MacCartney et al., 2008), edit distance and bag-of-words lexical overlap measures (Kouylekov and Negri, 2010) from this body of work. Mehdad et al. (2010) first transferred the task to the cross-lingual setting, however reducing the task to monolingual entailment again by using machine translation to create pivot representations; Mehdad et al. (2011) present the first actually cross-lingual approach by exploiting parallel corpora to generate phrase alignment features, which are then passed to a classifier that decides, if entailment is present. Our approach combines ideas from both, mostly resembling Mehdad et al. (2011). There are, however, several significant differences: We use word translation probabilities instead of phrase tables for generating alignment features and model monolingual and cross-lingual alignment separately. Furthermore, we do not use linguistic processing for generating cross-lingual features. Regarding the view on entailment, MacCartney and Manning (2007) proposed the decomposition of top-level entailment, such as EQUIVALENCE (which corresponds to the CLTE BIDIRECTIONAL class), into atomic forward and backward entailment predictions, which is mirrored in our multi-label approach with two binary classifiers.

In addition to the TE task itself, there are several fields in which similar problems are considered. The field of MT evaluation aims to find measures which quantify monolingual semantic similarity of two sentences,. Our approach derives features from the MT evaluation metric Meteor (Denkowski and Lavie, 2011), similar to its use in Volokh and Neumann (2011) for the monolingual TE task[5] or Finch et al. (2005). Inversely, Padó et al. (2009) showed that textual entailment features can be used to measure MT quality, indicating a strong relatedness

[5]We employ Meteor a differently, though: we compute separate alignment features instead of using only the final Meteor score and add our own features computed on the Meteor alignment, such as null alignments and longest unaligned sequence.

of the two problems. In fact, the CLTE task is also closely related to the problem of identifying parallel sentence pairs in a non-parallel corpus, e.g. the web (Resnik and Smith, 2003). Cross-lingual similarity is usually quantified based on an alignment between source and target features. For example, Smith (2002) proposed the *tsim* metric, which is based on the number of links between source and target, and is a cross-lingual variant of Jaccard similarity or *resemblance* (Broder, 1997), which is used to calculate the similarity of documents in the same language. Similar to Munteanu and Marcu (2005) we combine several cross-lingual measures by modeling them as features and use machine learning to train a classifier, which judges if two sentences are sufficiently parallel. Since we identified the need to detect aligned parts of the sentence, we employ features similar to those used to detect parallel sub-sentential fragments in non-parallel corpora (Munteanu and Marcu, 2006).

3.2.1 The 2012 and 2013 SemEval CLTE for Content Synchronization Task

The first CLTE for Content Synchronization Task was held at SemEval-12 (Negri et al., 2012); 10 teams participated. Apart from our combined approach, systems were either based on purely cross-lingual features or on monolingual features using pivot translations. Since a comparison of our system and the other submissions is provided in Section 3.5.5, we will not discuss approaches in detail at this point. The CLTE task was organized again the following year at SemEval-13 with a new test collection (Negri et al., 2013). Six teams competed, among them a team formed by the organizers (Turchi and Negri, 2013). Three teams (Vilarino et al., 2013; Kouylekov, 2013; Jimenez et al., 2013) returned. Again, approaches were both either cross-lingual (Turchi and Negri, 2013; Kouylekov, 2013; Graham et al., 2013) or based on translation (Vilarino et al., 2013; Zhao et al., 2013; Jimenez et al., 2013). Only one team combined both types of features, similar to our approach, but used less features in total (Graham et al., 2013). In particular, Vilarino et al. (2013) explored *n*-gram based features at word an part-of-speech level and experimented with different classification setups. Kouylekov (2013) employ standard RTE distance metrics in a cross-lingual setup using a dictionary and multilingual term weights from Wikipedia. Zhao et al. (2013) combined different features extracted from monolingual sentence pairs obtained by automatic translation, including syntactic information. Jimenez et al. (2013) based features on (monolingual) representations of the input sentences as sets of tokens or scores and computed intersection and unions based on the cardinality of theses sets.

3.3 Modeling CLTE with SMT Features

We define a number of dense features on pairs (E, F); most features can be characterized as similarity scores. The different measures are combined in a linear model of the form

$$Y_i = w \cdot X_i$$

where a label Y_i is predicted based on observations X_i, encoded as a feature vector. The parameter vector w can be estimated with statistical learning on a set of training examples with gold-standard labels. Since such a framework can handle a large amount of features, we combine both cross- and monolingual similarity scores to generate feature representations. The approach can therefore be considered *hybrid* in the context of the task. Two monolingual variants of the input sentence pairs are obtained by translating the English sentence E into the language of F, yielding the pair $(Tr(E), F)$ and vice versa F into English, yielding $(E, Tr(F))$. Google Translate[6] is employed as a translation engine in order to have a robust translation baseline that is able to provide similar translation quality for all language pairs, independent of domain – properties that would be hard to achieve if we chose to train our own system. Since our approach combines diverse features, we rely on the learner to identify useful features by assigning them weights. In Section 3.5 we perform a feature analysis to determine the contribution of individual features.

3.3.1 Length-based Features

The length ratio of a sentence pair, i.e. the fraction of the number of tokens, is a purely structural indicator for added or missing information. We define three length-based measures, one on the cross-lingual pair and two on its monolingual variants using full translations as pivot, to capture this information.

- English-to-Foreign length ratio, $\text{lr}_{\text{E2F}} = \frac{|E|}{|F|}$
- English-to-English-Translation length ratio, $\text{lr}_{\text{E2Tr(F)}} = \frac{|E|}{|Tr(F)|}$
- Foreign-to-Foreign-Translation length ratio, $\text{lr}_{\text{Tr(E)2F}} = \frac{|Tr(E)|}{|F|}$

Using several similar, overlapping measures adds robustness to our approach. Obviously, purely length-based features are completely uninformed about the content of the text and will only work in the narrow setting of the task, where each sentence is by default semantically related to its counterpart. This feature group therefore serves mainly as a baseline to assess the contribution of other, more informed features.

[6] http://translate.google.com/

3.3.2 Cross-lingual Alignment Features

We define a number of cross-lingual, content-sensitive similarity features based on word alignments between tokens $e_i \in E$ and $f_j \in F$. We obtain a lexical translation table by estimating word translation probabilities (Brown et al., 1993) with GIZA++ (Och and Ney, 2003) on a data set concatenated from Europarl-v6[7] (Koehn, 2005) and a bilingual dictionary obtained from dict.cc[8][9] added to extend coverage. We chose these resources, because approximately the same amount of data is available for all four language pairs. Instead of working with the bidirectional alignment produced by GIZA++, we define out own alignment model: since we can reasonably assume that both sentences are well-formed in their respective language, we treat them as bags-of-words and discount positional information. In fact, we observed that the paraphrasing operations used to construct the training and test data resulted in translations that are semantically equivalent but phrased very differently than the original sentence. In order to capture these correspondences, we found that a basic greedy one-to-many alignment model without any positional bias worked best as the foundation for cross-lingual features. We define alignment functions $a_{E2F} : i \rightarrow j$ and $a_{F2E} : j \rightarrow i$ that align each token e_i in E with the token f_j in F with the highest associated translation probability $t(e_i|f_j)$ based on the respective lexical translation table and vice versa.

$$a_{E2F}(i) = \underset{j=1,\ldots,|F|}{\operatorname{argmax}} \; t_{E2F}(e_i|f_j) \quad \text{and} \quad a_{F2E}(j) = \underset{i=1,\ldots,|E|}{\operatorname{argmax}} \; t_{F2E}(f_j|e_i)$$

Alignment links are generated independently and there is no word order correlation bias. Words for which no possible translation is found in the lexical translation table are considered unaligned, that is aligned to the `NULL` token e_0 or f_0. No further restrictions, e.g. a one-to-one assumption, apply, so we can simply generate alignment links sequentially without having to resort to approximations such as competitive linking (Melamed, 2000). Resnik and Smith (2003) point out that when using lexical translation probabilities under such a model, frequent words[10] tend to collect a large amount of probability mass. Resnik and Smith (2003) therefore resort to an equiprobable translation lexicon in their model. In contrast, we did not find this phenomenon to be detrimental to our approach. We discuss this point in Section 3.6.1.

Since we are especially interested in the direction of the entailment, we found that using both alignment directions separately, i.e. both a_{E2F} and a_{F2E}, to compute features was superior to using a symmetrized set alignment gained with a heuristic such as *union*, *intersection* or *grow-diag-final*. Figure 3.5 visualizes a sample set of alignment links given by an alignment function

[7] http://www.statmt.org/europarl/

[8] http://www.dict.cc/

[9] The dictionary contains linguistic annotations, such as voice and numbers or tags indicating colloquial speech. We cleaned the dictionary by applying regular expressions, which removed all annotations.

[10] These usually correspond to *non-content* or *stop words*.

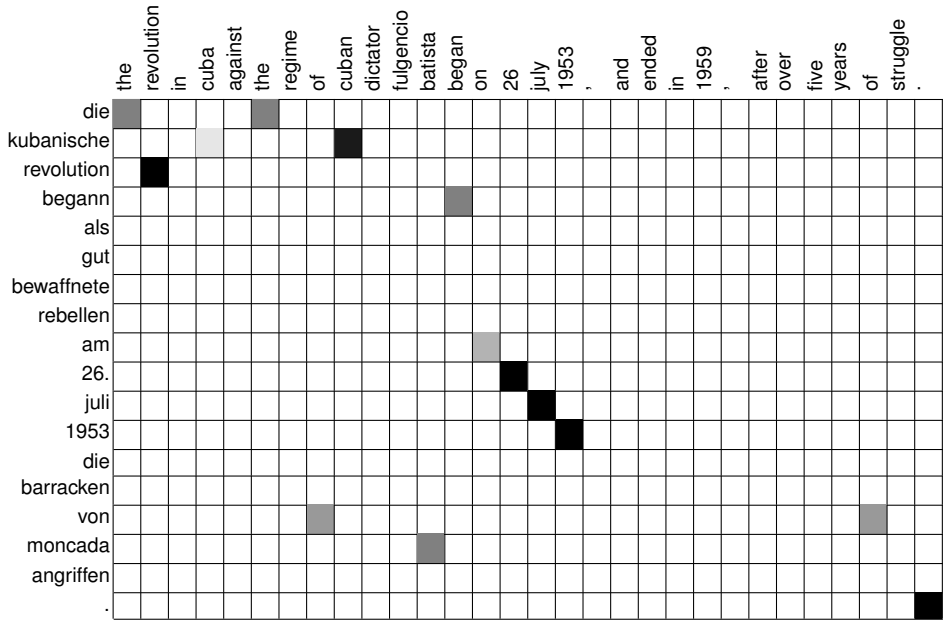

Figure 3.5: Alignment between German and English example sentence. The shade of the box indicates $t(e_i|f_j)$ – the darker the box the higher the word translation probability.

generated by our model. All content words are aligned correctly with a strong associative weight (black or dark grey boxes). Some non-content words are incorrectly aligned, however, with a weaker association (light grey boxes). Our alignment strategy yields two separate alignment functions a_{E2F} and a_{F2E}, that correspond to sets of alignment links with associated lexical translation probabilities t_{E2F} and t_{F2E}. From these, we derive the following features.

Alignment Precision. Two features that take the value of the number of aligned tokens in E and F, respectively, normalized by the total number of tokens.

$$\text{pr}_{\text{E2F}} = \frac{1}{|E|}\sum_{i=1}^{|E|}\min(a_{E2F}(i),1) \quad \text{and} \quad \text{pr}_{\text{F2E}} = \frac{1}{|F|}\sum_{j=1}^{|F|}\min(a_{F2E}(j),1)$$

A higher number of linked tokens corresponds to a higher alignment precision. Additional information in sentence F would for examples result in high $E2F$ alignment precision and low $F2E$ alignment precision.

Alignment Score. The alignment score corresponds to the alignment precision, but with each alignment weighted by its word translation probability according to the model.

$$\text{sc}_{\text{E2F}} = \frac{1}{|E|}\sum_{i=1}^{|E|} t(e_i|f_{a_{E2F}(i)}) \quad \text{and} \quad \text{sc}_{\text{F2E}} = \frac{1}{|F|}\sum_{j=1}^{|F|} t(f_j|e_{a_{F2E}(j)})$$

This extension to the alignment precision is meant to capture the strength of the association between two tokens. Non-content words tend to align with content words frequently, as shown in Figure 3.5, but with lower probability. The alignment score accounts for this phenomenon.

Normalized number of unaligned words. These features take the value of the absolute number of unaligned words in E and F, respectively, normalized by the length of the sentence.

$$\text{na}_{\text{E2F}} = \frac{1}{|E|}\sum_{i=1}^{|E|} \mathbb{1}_{\{0\}}(a_{E2F}(i)) \quad \text{and} \quad \text{na}_{\text{F2E}} = \frac{1}{|F|}\sum_{i=1}^{|F|} \mathbb{1}_{\{0\}}(a_{F2E}(j))$$

where $\mathbb{1}_{\{0\}}$ is an indicator functions that returns 1, when an alignment is 0.

Longest unaligned sequence. Two features us_{E2F} and us_{F2E} that take the absolute value of the length of the longest contiguous sequence of unaligned words in E and F, respectively. While single unaligned words may be due to the word alignment model being out-of-vocabulary, a contiguous sequence of unaligned words is an indicator for added information in one language.

3.3.3 Monolingual Similarity Metrics

Lexical overlap metrics computed on bag-of-words representations of text are frequently used similarity measures in monolingual TE. The most popular SMT evaluation metric BLEU uses n-gram precision to quantify the similarity of a translation hypothesis and a reference Papineni et al. (2002). By using the two pivot variants $(E, Tr(F))$ and $(F, Tr(E))$ we can compute similar features on the CLTE pairs.

N-gram Overlap. The overlap coefficient of two sets A and B is defined as the cardinality of their intersection divided by the cardinality of their union. It is related to Jaccard similarity, where the cardinality of the intersection of two sets is divided by the size of the larger set.

$$\text{OC}(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)}$$

The lexical overlap is computed for unigram, bigram and trigram representations of the two

sentences for both translation pairs, yielding $2 \times 3 = 6$ features[11]. Similar to BLEU, the n-gram precision is measured at different levels of n. We do not compute a brevity penalty, since this information is already captured by the length-based features.

Edit Distance Alignment. In addition to the overlap features, we include edit distance via a purely string-based monolingual alignment, similar to the cross-lingual alignment described above. Instead of their joint translation probability, we use string edit distance ED as a similarity measure. Each token $e_i \in E$ is sequentially aligned to its most similar token $e'_j \in Tr(F)$, i.e. the one it has the smallest edit distance to,

$$a_{E2Tr(F)}(i) = \underset{j=1,\ldots,|Tr(F)|}{\operatorname{argmin}} \mathrm{ED}(e_i, e'_j) \quad \text{and} \quad a_{Tr(E)2F}(j) = \underset{i=1,\ldots,|Tr(E)|}{\operatorname{argmin}} \mathrm{ED}(f_j, f'_i)$$

for each token $f_j \in F$ and the translation $f'_i \in Tr(E)$. With this procedure, exact matches will always align to each other. Furthermore, morphological variations and spelling variants are more likely to be aligned while providing an estimate of how close the two forms are. On both alignments we can then compute an alignment score by summing up over all distances in the alignment. We include the log function for scaling.

$$\mathrm{ed_E} = \log \sum_{e_i \in E} \min_{e'_j \in Tr(F)} \mathrm{ED}(e_i, e'_j) \quad \text{and} \quad \mathrm{ed_F} = \log \sum_{f_j \in F} \min_{f'_i \in Tr(E)} \mathrm{ED}(f_j, f'_i)$$

3.3.4 Meteor Features

The Meteor evaluation tool (Denkowski and Lavie, 2011) relies on linguistic information to generate an alignment between two sentences in the same language. The semantic similarity of the two sentences can then be compared based on this alignment. Meteor is usually used for scoring the output of statistical machine translation systems given a reference. In particular, the tool employs stemming, paraphrase tables and synonym collections to align words between the two sentences (for an examples, see Figure 3.6). Meteor scores the resulting alignment by combining alignment precision $\mathrm{pr_M}$, recall $\mathrm{rc_M}$ and a fragmentation penalty, where *chunks* is the number of contiguous sequences of matched words,

$$\mathrm{pn_{frag}} = \frac{|chunks|}{|matches|}$$

into a weighted score with adjustable weight parameters α, β and γ.

[11] Denoted as $\mathrm{oc}_{E_{1gr}}$, $\mathrm{oc}_{E_{2gr}}$, $\mathrm{oc}_{E_{3gr}}$ and $\mathrm{oc}_{F_{1gr}}$, $\mathrm{oc}_{F_{2gr}}$, $\mathrm{oc}_{F_{3gr}}$.

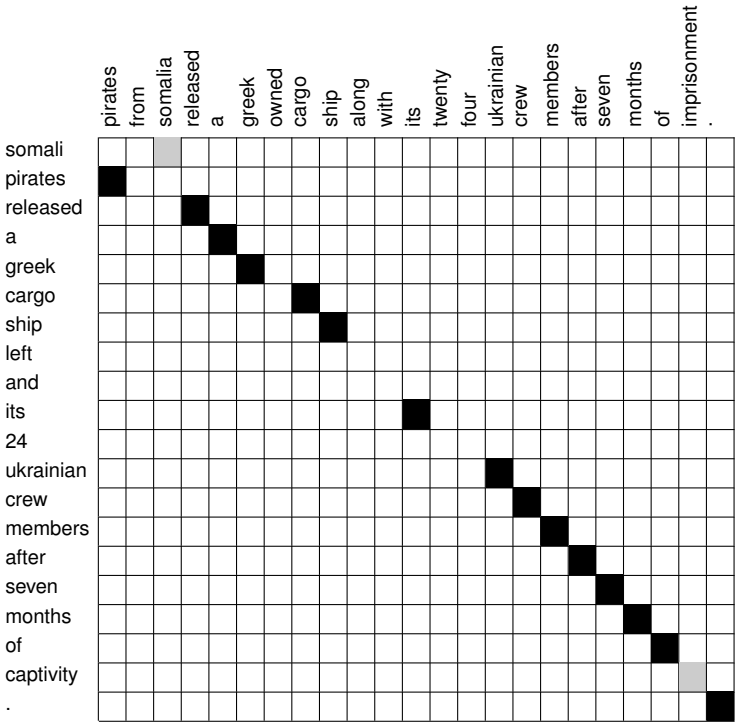

Figure 3.6: Meteor alignment matrix between two English sentences. Black boxes indicate exact matches, grey boxes matches at stem or synonym level, e.g. (*somali, somalia*) and (*captivity, imprisonment*).

$$\mathrm{sc_M} = (1 - \gamma \times \mathrm{pn_{frag}}^{\beta}) \frac{\mathrm{pr_M} \times \mathrm{rc_M}}{\alpha \times \mathrm{pr_M} + (1-\alpha) \times \mathrm{rc_M}}$$

We use both the overall score[12] and separate precision, recall and fragmentation scores as features in our model. Meteor provides linguistic resources for several languages, but the synonymy module is only available for English. We therefore compute scores on $(E, Tr(F))$.

3.3.5 Additional Meteor Alignment Features

We can use the alignments output by the Meteor scorer for $(E, Tr(F))$ to calculate – in addition to the separate Meteor scores mentioned above – the following features, analogous to the features computed on the cross-lingual alignments[13].

- normalized number of unaligned words, $\mathrm{na_M}$

[12] The score was computed with the pre-tuned weights shipped with Meteor-1.3.

[13] Alignment precision is already included in the model via the separate Meteor scores.

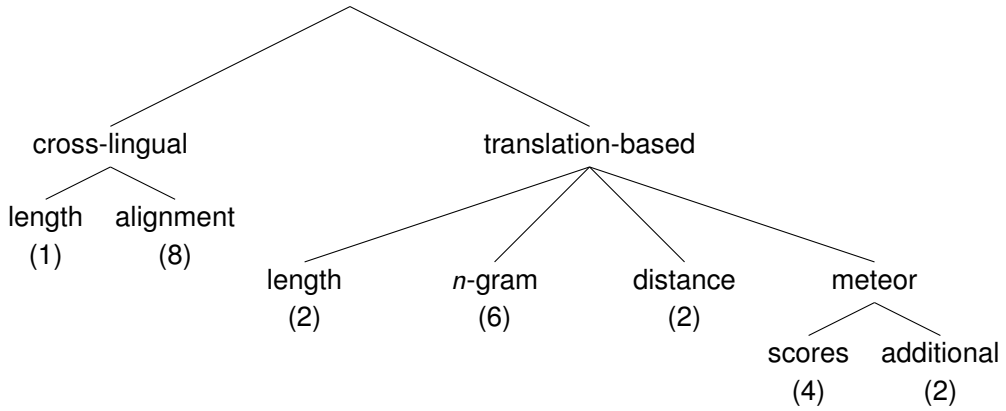

Figure 3.7: Features grouped by type.

- length of the longest unaligned subsequence, us_M

In total, we define 25 dense features: 9 cross-lingual features, i.e. computed directly on E and F, and 16 monolingual features, i.e. based on translations of E or F paired with the original other sentence. Figure 3.7 shows a taxonomy of the features.

3.4 Classification

We treat the entailment task as a classification problem and encode all previously described metrics as features in a statistical learning framework. As dense features like the similarity scores and key figures described above take values in different data ranges, we normalize all feature value distributions to the normal distribution $\mathcal{N}(0, \frac{1}{3})$, so that 99% of the feature values lie in the interval [-1, 1]. We employed two toolkits to learn a model for predicting entailment. Both toolkits implement *support vector machines* (SVM) (Vapnik and Vapnik, 1998).

SVM^{light}[14] is a widely used, light-weight toolkit for learning SVMs. It implements binary classification, structured prediction and ranking SVM using varying optimization strategies. Multi-class learning is framed as an instance of structured prediction (Joachims, 1999).

Sol[15] was developed at ICL Heidelberg and was employed for this task in order to evaluate its performance compared to SVM^{light}. It uses stochastic gradient descent and implements binary, multi-class and multi-label classification as well as regression and structured prediction (Fendrich, 2012).

An SVM is a supervised model, which learns a hyperplane that separates the projections of the training examples in the feature space according to the output classes. It is an instance of

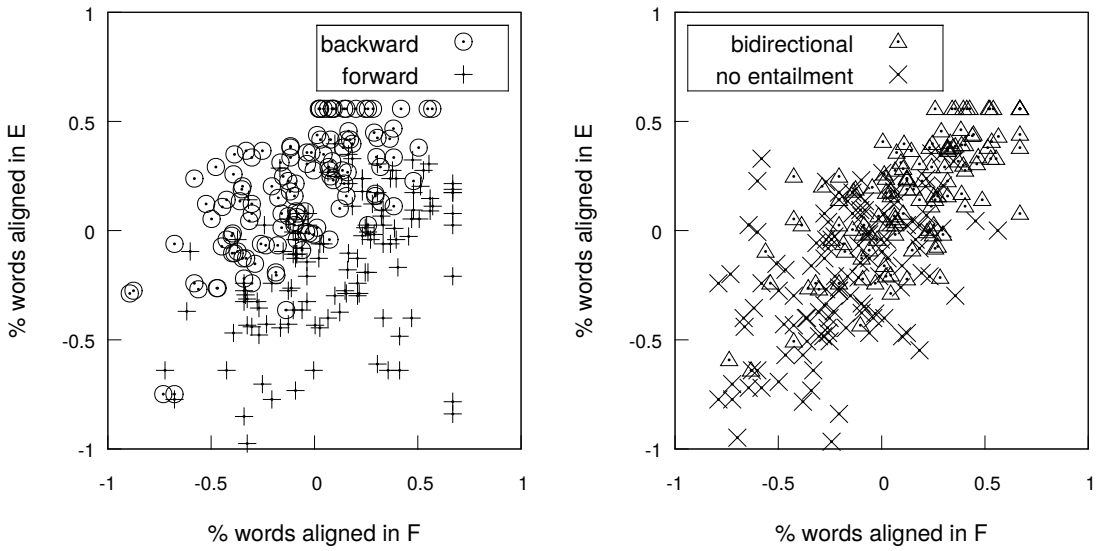

Figure 3.8: Example space for two features.

a large margin classifier, i.e. it aims to find the best separating hyperplane which maximizes the distance to the training data points by minimizing the squared norm. Our goal is therefore to identify features that make the training data separable. Figure 3.8 depicts an example feature space with the values of the two cross-lingual alignment ratio features. With the two features shown in Figure 3.8, the examples are not separable, but a tendency for the entailment classes to reside in different areas of the feature space[16] is visible. There are different learning strategies in order to predict the output for the CLTE task.

Binary Classification. A binary classifier has the ability to output exactly two labels $y \in -1, 1$, corresponding to *yes* and *no*. In our task, such a classifier can predict the presence $(+1)$ or absence (-1) of directional entailment.

Multi-class Classification. Multi-class learning aims to discriminate between $Y = 1, ..., k$ target classes, four in our cases. The final class is directly output by the classifier. Possible strategies for multi-class classification are *structured prediction*, where the classifier output is a vector of probabilities where each entry corresponds to a target class, or the *one-vs-all* scenario, where a binary classifier is learned that discriminates class j from all other classes $\neq j$ for each class $j \in 1, ..., k$. Both Sol and SVM^{light} implement the *structured prediction* strategy for multi-class learning.

[16] Depicted in two separate figures for better readability.

Multi-label Classification. Instead of one single output label, multi-label classification assigns every instance a subset of non-exclusive labels, $Y \subseteq 1, ..., k$. Multi-label classification can be implemented by learning a binary classifier for each label – either independently or using a single objective function. (Fendrich, 2012).

Since the CLTE task defines four target entailment classes, multi-class classification seems like the obvious choice to address the problem. However, the four entailment relations can be further broken down into two atomic relations, namely directional entailment from E to F and from F to E (see Table 3.1). This makes is possible to learn a separate binary classifier for each atomic entailment relation and combine the output to obtain the final entailment class following Table 3.1. This approach is actually an instance of multi-label learning, with the presence or absence of an atomic entailment relation as the values of two labels: $E \rightarrow F$ and $F \rightarrow E$. Sol (but not SVM^{light}) features an implementation of multi-label learning with a single objective function. We can compare this to two separately learned binary classifiers.

$E \rightarrow F$	$F \rightarrow E$	**entailment**
1	1	*bidirectional*
1	0	*forward*
0	1	*backward*
0	0	*no entailment*

Table 3.1: Combination of atomic entailment relations.

3.5 Experimental Results

To train supervised systems, the CLTE task organizers provided 500 sentence pairs for development annotated with the four entailment relations defined in Section 3.1, as well as 500 pairs (without annotation) for testing. The classes are equally distributed over both data sets and the test data mirrors the training data distribution. All reported results on development data are calculated with 2-fold cross-validation, i.e. by splitting the development set into two parts of 250 sentence pairs each, learning the model on one part and evaluating on the other and vice versa. We report the mean classifier accuracy[17] over both runs. We employed random permutation (Noreen, 1989) to assess significance between two system runs: class labels of both runs are randomly shuffled and the evaluation metric – accuracy in our case – recomputed. We then count, how frequently the difference in score on the shuffled data exceeds the original difference in score. The more frequently the shuffled data produces a similar or larger difference, the more likely it is that the difference we observed between the

[17] *Accuracy* is defined as the number of correctly classified examples, both positive and negative, divided by the total number of examples.

two runs is random. In practice, we shuffled the labels a 1,000 times and considered results at a level of $p < 0.05$ significant, meaning that only in less than 5% of the random assignments the observed difference in accuracy turned out larger than the original difference.

3.5.1 Baselines

	es-en	**it-en**	**fr-en**	**de-en**
random	25.0%	25.0%	25.0%	25.0%
length	40.8%	39.8%	41.2%	39.4%

Table 3.2: Baselines on development set.

We calculated two baselines to assess the quality of our model (Table 3.2). As a random baseline, we assigned the same entailment relation to all examples, which results in an accuracy of 25.0%, since the four classes are equally distributed across the data set. The length-based baseline consisted of a system using only the first feature described in Section 3.3, the token ratio measuring the fraction of the numbers of tokens in text E and F. This baseline is actually quite competitive, but this is, as pointed our before, a direct result of the limited scope of the task. In a real-world application, sentences with no content overlap will be assessed, so a purely length-based criterion would fail.

3.5.2 Choosing a Classifier Setup

	es-en	**it-en**	**fr-en**	**de-en**
multi-class	47.0%	45.6%	46.6%	45.8%
multi-label	58.6%	52.6%	56.8%	52.2%
2 $\times$ binary	64.6%	61.4%	62.8%	58.8%

Table 3.3: Accuracy employing two different classifiers to estimate the parameters of the linear model (2-fold cross-validation on development set).

Table 3.3 shows results obtained with the full feature set using three different classification strategies. Using two binary classifiers to detect the two entailment directions separately significantly outperforms both other options on all language pairs. Although multi-label learning makes use of the same information by learning the separate entailment directions as labels, it consistently performs 4-9 percentage points worse. The multi-class classification with the *one-vs-the-rest* strategy for the four entailment classes places a distant third. The results suggest that the view of the problem as detecting entailment only in one direction at a time makes

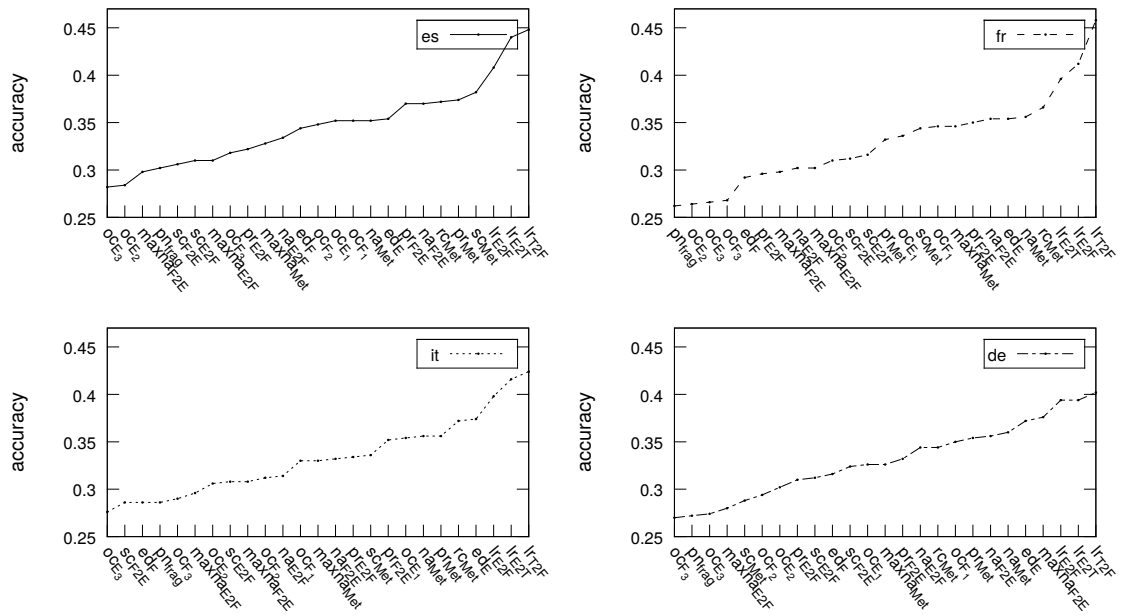

Figure 3.9: Performance of individual features.

it easier to model for a statistical learner and that joint modeling of the two atomic entailment relations is actually disadvantageous.

3.5.3 Individual Feature Results on Development Data

To assess the contribution of single features, we performed three studies of their individual behavior. Due to the number of features, it was not feasible to test all possible combinations of features and determine the best. In the first study, we evaluated each feature by learning a classifier only on the single feature and measured the performance of the resulting classifier on the development data. Figure 3.9 shows results for individual features, ranked by performance. When only one feature is available to the classifier, the purely length-based features (lr_{E2F}, $lr_{E2Tr(F)}$ and $lr_{Tr(E)2F}$) perform significantly better than other features, while overlap features, especially with higher order *n*-grams ($oc_{E_{2gr}}$, $oc_{E_{3gr}}$, $oc_{F_{2gr}}$, $oc_{F_{3gr}}$), perform worse. The Meteor scores, sc_M, pr_M and rc_M, are good individual predictors for all languages pairs except **de-en**. This can be explained by the fact that automatic translation from and into German is more challenging than into the other languages due to the richer morphology.

The power of the features lies in their combination, as can be observed in Figure 3.10. For this study, we incrementally added one feature at a time, starting from a baseline classifier with only one length-based feature and ending with a classifier trained on the full set of features.

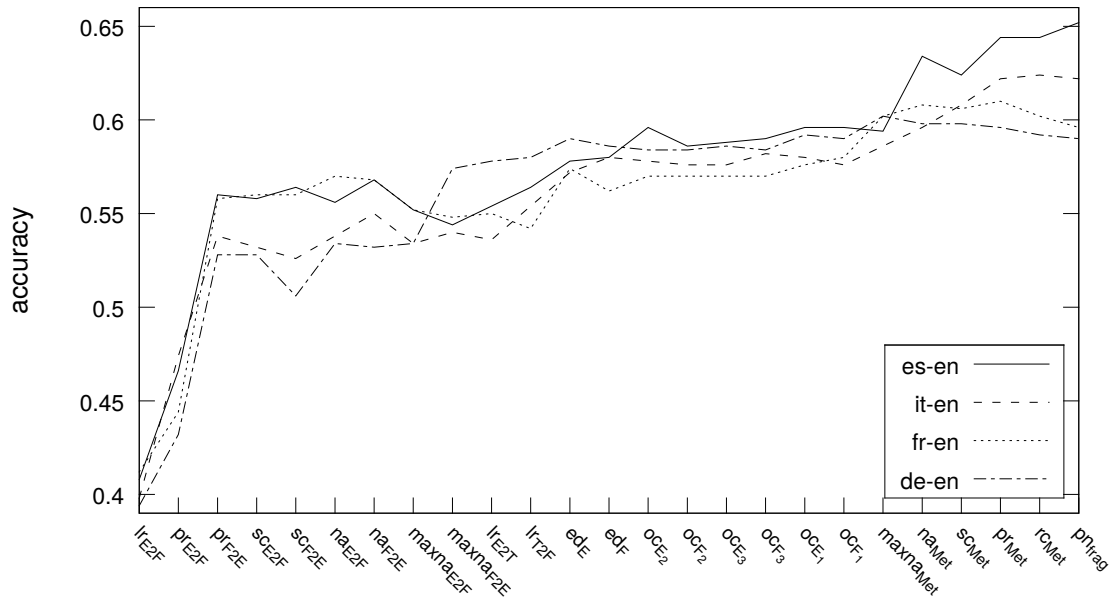

Figure 3.10: Incremental feature activation.

Performance climbs steeply with the first two added features, $lr_{E2Tr(F)}$ and $lr_{Tr(E)2F}$ alignment ratios, and continues to climb steadily. This shows that later features do not add much more information, but still contribute to and stabilize the overall result. Observing the individual lines, adding some features results in small drops in score on **es-en**; the score later recovers and finally receives a significant boost from the Meteor alignment features. On the **de-en** task, the largest boost comes from the features capturing the length of the longest unaligned subsequence. This observation is confirmed by the third study, where, starting from a full system, we evaluate the change in accuracy from switching off, i.e. removing a single feature from the full set. The longest unaligned sequence features play a crucial role on all subtasks. The performance of the system for the **es-en** subtask drops significantly, when the Meteor precision feature is switched off, but apart from that, the study confirms the robustness of the feature set, since the removal of a single feature has no significant influence on the score.

To get more conclusive results, we performed ablation tests for groups of features. Sets of similar features were switched off, i.e. not used to train the classifier, and the impact on accuracy measured (Table 3.4). A significant impact is indicated in **bold** font. A single feature group significantly impacts the score only in two cases, namely the Meteor score features for **es-en** and the cross-lingual alignment features for **de-en**. However, no feature group hurts the score either, since negative variations in score are not significant. To ensure that the different feature groups actually express diverse information, we also evaluated our system using only one group of features at a time (see Table 3.5). The results confirm the most

Figure 3.11: Ablation test, i.e. switching off one feature and evaluating the performance drop, on development set.

significant feature type for each language pair, but even the best-scoring feature group for each pair always yields scores 3-6 percentage points lower than the system with all feature groups combined. We conclude that the combination of diverse features is one key aspect of our system.

3.5.4 Results on SemEval12 CLTE Test Set

We submitted two runs to SemEval12, using the full feature set and a combination of two binary classifiers as a setup. The runs differed only in the choice of learner. Accuracy scores are given in Table 3.6. Sol outperforms SVM^{light} on three of the four language pairs, although the differences are not significant. Table 3.7 shows detailed results for the best run for each entailment class separately. Precision is highest for the BACKWARD entailment class for three out of four language pairs; for the **fr-en** subtask, the classifier achieves the highest precision on the NO ENTAILMENT class, but at the cost of very low recall. Recall is equally good on the BACKWARD class, making this the entailment relation that is most reliably identified by our classifier for all language pairs, followed by the FORWARD relation. BIDIRECTIONAL and NO ENTAILMENT are less well delimited, with NO ENTAILMENT getting least reliably identified.

feature group (#)	**es-en**	**it-en**	**fr-en**	**de-en**
Meteor scores(4)	**61.6**%	60.0%	61.8%	59.0%
monolingual (8)	64.4%	60.8%	62.0%	59.6%
token ratio (3)	65.2%	60.6%	62.0%	58.8%
cross-lingual align (8)	63.8%	59.2%	62.0%	**52.6**%
monolingual align (2)	64.8%	62.4%	59.0%	59.6%
all (25)	64.6%	61.4%	62.8%	58.8%

Table 3.4: Ablation tests on development set. Significant impact on accuracy by removing the specified feature group is denoted in **bold** font.

feature group (#)	**es-en**	**it-en**	**fr-en**	**de-en**
Meteor scores(4)	61.6%	55.2%	56.8%	52.2%
monolingual (8)	54.4%	51.6%	55.8%	50.2%
token ratio (3)	45.0%	42.2%	44.6%	41.2%
cross-lingual align (8)	50.2%	50.8%	51.6%	55.8%
monolingual align (2)	58.0%	55.4%	55.0%	47.8%
all	64.6%	61.4%	62.8%	58.8%

Table 3.5: Classifier accuracy of individual feature groups on development set.

3.5.5 System Comparison

Our system performed best out of ten systems for the language pairs **es-en** and **de-en** and tied in first place for **fr-en**. For **it-en**, our system came in second. Table 3.8 shows an overview of the systems designed for SemEval12. The majority use machine learning (ML) to assign entailment classes to the given sentence pairs. When it comes to features, all system except ours restrict themselves to exclusively using either cross-lingual (**cl**) or translation based (**tb**) features, relying on the automatic translation of one of the two sentences. The official results on the test set suggest that the richer feature space is an advantage for our model. In general, approaches that used heuristics performed worse than systems using ML.

	es-en	**it-en**	**fr-en**	**de-en**
SVM^{light}	63.0%	55.%	56.4%	55.8%
Sol	63.2%	56.2%	57.0%	55.2%

Table 3.6: Results on test set.

	FORWARD			BACKWARD		
	P	R	F1	P	R	F1
de-en	56.7%	54.4%	55.5%	59.6%	67.2%	63.2%
fr-en	56.4%	67.2%	61.3%	58.2%	73.6%	65.0%
it-en	56.4%	60.0%	58.1%	62.8%	64.8%	63.8%
es-en	60.7%	65.6%	63.1%	67.7%	70.4%	69.0%

	NO ENTAILMENT			BIDIRECTIONAL		
	P	R	F1	P	R	F1
de-en	53.6%	48.0%	50.6%	50.4%	51.2%	50.8%
fr-en	67.6%	38.4%	49.0%	50.0%	48.8%	49.4%
it-en	55.1%	52.0%	53.5%	50.0%	48.0%	49.0%
es-en	60.2%	59.2%	59.7%	64.3%	57.6%	60.8%

Table 3.7: Precision, recall and F-measure for all entailment classes.

3.6 Discussion

We have shown that features obtained by using methods from statistical machine translation can be successfully applied to a cross-lingual variation of a semantic task. We found that combining similarity scores computed on both cross-lingual and monolingual alignment features resulted in the highest accuracy. Comparing cross- and monolingual features yielded mixed results depending on the language pair. Apart from using scores obtained with the Meteor tool, which employs paraphrase and synonym tables, no linguistic processing was used, which makes the approach easily portable, as long as parallel data is available. Key to our approach is furthermore the view of the four-class entailment problem as a bidirectional binary problem. We conclude that our results can be improved most by building on better alignments, i.e. using more data for estimating cross-lingual alignments and larger paraphrase tables. Experiments with more informed features, as briefly discussed in Section 3.6.1, did not yield better classification results.

3.6.1 Discarded Features

While developing the model, several extensions and additional features were designed and tried out on the development data. For the sake of completeness and as a hopefully helpful guide for future work in the field of CLTE, we sum up some of the findings below.

Enlarging the training data. We created additional training pairs by reversing E and F and

system	tb	cl	assignment
Wäschle and Fendrich (2012)	○	○	classifier
Esplà-Gomis et al. (2012)		○	ML
Jimenez et al. (2012)	○		classifier
Mehdad et al. (2012)		○	classifier
Meng et al. (2012)	○		classifier
Vilariño et al. (2012)	○		heuristic
Castillo and Cardenas (2012)	○		classifier
Perini (2012)		○	heuristic
Kouylekov et al. (2012)		○	classifier
Neogi et al. (2012)	○		heuristic

Table 3.8: Classifying the systems submitted for the SemEval 12 CLTE task along the dimension of features (translation-based or cross-lingual) and method for assigning examples an entailment class based on those features.

adjusting the label accordingly: FORWARD and BACKWARD label where switched to reflect the reverse relation, while BIDIRECTIONAL and NO ENTAILMENT relations remained fixed. However, adding the additional data had no influence on the overall result, suggesting that the size of the given data set was sufficient to learn the model.

Stop word filtering. When estimating word alignments, non-content words such as articles or prepositions tend to receive translation probability mass with many different content words. In a greedy alignment procedure, as used in our model, this makes them align to content words that actually have no counterpart in the other language. Due to this, important indicators for added information might be lost. To prevent greedy alignments of content and non-content words, we added a customized stop word filtering step for each language before calculating alignments. Contrary to our intuition, this did not improve system performance significantly.

IDF weights. Instead of radically discarding non-content words by filtering with a stop word list, we tried weighting tokens by their *inverse document frequency* (IDF) when computing alignment precision. IDF of a term t is defined as $log(1+N/n_t)$, where N is the total numbers of documents in the corpus and n_t the number of documents that contain t. Alignment scores that included IDF weights performed comparably to their unweighted version.

Stemming. Since the step most prone to errors, on which many features rely, is the alignment between E and F, we analyzed the most common alignment mistakes. Even with the recall-oriented alignment, incorrectly NULL-aligned tokens were a big problem, i.e. the system was frequently out-of-vocabulary. We tried stemming tokens using the

Porter stemmer (Porter, 1980) before computing the cross-lingual alignment and before computing *n*-gram overlap in the monolingual setting, but did not gain consistent improvements from this measure.

Compound splitting. When translating German text into English, compound splitting on the German side can be beneficial to correctly translate German compound words, which can be arbitrarily long and are not likely to be found in a dictionary or training corpus (Wäschle and Riezler, 2012). We hypothesized that compound splitting on the German sentences could help with the alignment of German and English compound words (for an example see Figure 3.6.1[18]), and trained a compound splitter on the German portion of the training corpus for the word alignments, but we found no improvements on the **de-en** subtask. This might simply be due to the fact, that only a small fraction of compound words was contained in the given examples, which did not influence performance much.

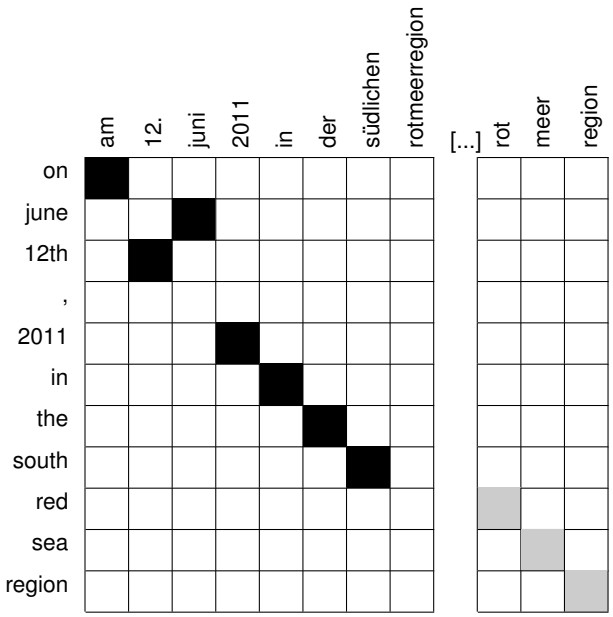

Figure 3.12: Word alignment between German and English text, without (left) and with compound splitting (right) on the German side.

Our developmental results suggest, that additional processing of the tokens might not be the direction in which to extend our system. The presented model relies crucially on parallel data to compute alignments and yield good automatic translations, suggesting that the precision of these steps might have the most influence on overall performance. However, De Souza et al. (2013) improved performance of a CLTE system by using shallow linguistic processing. While

[18]The shown phrases are an excerpt of example 443 of the SemEval12 CLTE training set, which was simplified for demonstration.

adding an IDF feature that computed a variant of alignment precision, where each token was represented by its associated IDF score, gave only small improvements, features taking into account part-of-speech information when computing alignment scores resulted in a significant boost in performance over purely quantitative features as used by our model.

CHAPTER 4

Quantifying Cross-lingual Similarity in a Noisy-Parallel Corpus

In the section on related work in Chapter 3 we discussed the link between cross-lingual textual entailment and the task of finding parallel phrases in a comparable corpus. Filtering out non-parallel sentences from a largely parallel corpus is a further complementary task. Usually, sentence-parallel corpora are automatically constructed from parallel documents, i.e. documents for which one or more translated versions are available. Since human translations are rarely literal, texts often exhibit discrepancies – to which extent depends on the translation process. To obtain parallel text from documents in the intellectual property (IP) domain, translations of patent text sections can be acquired through various channels (Wäschle and Riezler, 2012). Translations of full documents including the largest part, the description, are produced, when an invention is filed in different countries. However, several years can pass between the filing and different standards apply for document formatting in different countries. This can lead to discrepancies between different language versions of a document. Since pre-processing and sentence alignment are fully automated to deal with the large amounts of parallel data[1], non-parallel pairs can find their way into the parallel data. To follow up on open questions left by previous work (Wäschle and Riezler, 2012), we investigate the quality of sentence-parallel data automatically extracted from patents more thoroughly. We conduct a survey with human raters to estimate the amount of non-parallel or noisy parallel sentence pairs in the final data and to detect their origin. We propose to introduce an additional filtering step in the sentence alignment pipeline for data extracted from comparable documents and evaluate the influence of such a filter on SMT quality. The filtering system is based on the

[1] In the order of tens of millions of sentences.

classifier we employed to detect cross-lingual entailment described in the previous chapter. We find that the SMT pipeline is robust enough to deal with a significant amount of non-parallel data in the corpus. However, filtering reduces the amount of data necessary to train the SMT system without impacting translation quality, thus saving training time.

Discovering parallel sentences in parallel or comparable documents is a well researched problem. Gale and Church (1993) were the first to propose a sentence alignment algorithm based on sentence length ratio using dynamic programming (DP). Moore (2002) and Braune and Fraser (2010) extended the approach with a second pass using IBM Model 1 lexical probabilities obtained during the first, length-based pass, enabling them to produce high-precision alignments even for noisy parallel corpora. Since the amount of naturally occurring parallel data is limited and employing human translators to create new translations from scratch generally too expensive, the problem of finding parallel sentences in comparable data, e.g. from the web (Resnik and Smith, 2003), has become a field of research. Munteanu and Marcu (2005) proposed a two-step approach, using cross-language information retrieval (CLIR) to find topically related documents in different languages and a *maximum entropy classifier* to detect parallel sentences. This approach has since been extended, e.g. by making the search more efficient (Tillmann, 2009), mining parallel sentences from off-topic documents using a bootstrapping strategy (Fung and Cheung, 2004) or reducing the search space with a trainable translation similarity measure (Ştefănescu et al., 2012). The problem has also been expanded to the detection of sub-sentential parallel fragments (Munteanu and Marcu, 2006; Quirk et al., 2007). Our task looks at the problem from the opposite direction: instead of identifying parallel sentences in a large amount of non-parallel data, we try to detect a small rate of noisy parallel and non-parallel data in large amounts of bitext. A first step towards this goal is therefore to quantify the levels of parallelism in the data.

Filtering on top of a DP sentence aligner has been done in several ways. Mújdricza-Maydt et al. (2013) use bootstrapping with a conditional random field sequence classifier to improve the precision of a sentence aligner. Utiyama and Isahara (2003) propose an additional filtering step after CLIR and DP alignment and define measures for alignment quality, which correlate with human judgments. The measures are used to rank sentences aligned with a DP method and a cutoff value is determined based on a manual inspection of a sample of data at several points in the ranking. Lu et al. (2009) automatically align Chinese-English patent text and find that the resulting corpus is noisy parallel. They annotate a sample of 1000 aligned sentences manually and score all sentence pairs according to three different similarity measures. The respective rankings are evaluated against the annotated data; furthermore, the influence on an SMT system is measured. However, it is also worth considering, if filtering noise is necessary at all. Simard (2014) question the best practice of cleaning data before SMT training. They investigate the case of MT contamination by injecting automatically translated bitext into the training data for an SMT system and measuring its impact on translation quality.

Our approach is most similar to the work by Utiyama and Isahara (2003) and Lu et al. (2009). We employ the Gargantua aligner (Braune and Fraser, 2010), an instance of DP alignment based on length and a second filtering step based on lexical translation probabilities estimated in the first step. Like Utiyama and Isahara (2003), we work on data from the intellectual property domain and use human annotations to produce a gold standard rating of a sample of data. But instead of ranking the sentence pairs according to a similarity score, we use a discriminative approach to detect non-parallel sentences by applying a variant of our system designed to detect textual entailment (see Chapter 3) adapted to the filtering task. Meta-parameters of the learning approach, such as the cutoff value, are set on development data. Although the field is already well-researched, we think the study at hand is beneficial. Our concrete research goals are:

- How does the noisy parallel data influence translation quality on this particular corpus? Since the corpus is publicly available, this information can be helpful for researcher that consider using the corpus in their work. As a new data set, not many empirical studies are available so far to characterize the behavior of the data.
- What does the manual annotation of a sample from the data reveal about the problems of the parallel document extraction and automatic alignment? This information can be used to refine the alignment process for extracting patent bitext from comparable family patents for other language pairs.

We provide both intrinsic (on a hand-annotated set) and extrinsic (SMT task) evaluation to document our findings. The annotation serves two purposes:

Analysis. Determine the actual distribution of the levels of parallelism in the corpus.

Training Data. Provide the ability to learn on actual semi-parallel examples instead of artificially constructed semi- or non-parallel data.

We find that sentence pairs from patent abstract and claims sections contain very clean, parallel data, but that data extracted from descriptions contains about 15%-20% of noise. Even though the aligner uses a lexical filtering step, we believe that our filtering procedure will be able to identify these additional non-parallel sentence pairs, since information for the features, such as word translation probabilities, are estimated on established, clean parallel data extracted from patent titles, abstract and claims. We describe the fully automated data extraction and sentence alignment process in Section 4.1. The annotation procedure and an analysis of the results is described in Section 4.2. Section 4.3 details our adapted filtering model and experimental results are reported and discussed in Section 4.4.

4.1 Corpus Construction

Building on previous work on constructing a German-English parallel corpus of patent text (Wäschle, 2011; Wäschle and Riezler, 2012), we extracted bitext for two more language pairs, English-French and German-French[2] from the MAREC[3] patent document collection. MAREC contains more than 19 million patent applications and patents in a unified XML format. The patents have been collected from four major national and international patent agencies, EPO[4], WIPO[5], USPTO[6], and JPO[7], containing text in many European languages, most notably French and German, as well as English and Japanese. Some of the documents are multilingual and contain text sections in up to three languages. Following Wäschle (2011), we first extracted parallel sections[8] of text from multilingual patent documents using an XML parser. Since there are no parallel description sections in the same document, we aligned English patents from the monolingual USPTO corpus to patents with French content from the EPO and WIPO corpora. Since MAREC does not contain monolingual corpora of French nor German patents, this technique could not be exploited to derive description data for this language pair. The number of unique parallel sections in MAREC for both language pairs is shown in Table 4.1. We dealt with different *kinds*[9] of a patent document as detailed in Wäschle (2011), giving priority to document versions with a higher number of multilingual sections and newer publication date and disregarding other versions.

	title	abstract	claims	description
en-**fr**	2,504,772	1,172,121	617,736	57,236
fr-**de**	1,953,815	50,653	271,778	–

Table 4.1: Number of parallel sections.

Following the extraction step, we split and tokenized the raw text sections to prepare them for sentence alignment. We used the Gargantua aligner[10] (Braune and Fraser, 2010), a state-of-the-art sentence aligner (Abdul-Rauf et al., 2010) with reported precision values of more than 95% for different data sets. We employed Gargantua on batches of 10,000 parallel text

[2]The complete parallel corpus was released as PatTR under a Creative Commons license (by-nc-sa) and can be downloaded from http://www.cl.uni-heidelberg.de/statnlpgroup/pattr.

[3]http://www.ir-facility.org/prototypes/marec

[4]European Patent Office, https://www.epo.org/

[5]World Intellectual Property Organization, http://www.wipo.int/

[6]United States Patent and Trademark Office, http://www.uspto.gov/

[7]Japan Patent Office, http://www.jpo.go.jp/

[8]A patent document contains the text sections of title, abstract, claims and description.

[9]The kind, i.e. type of a patent document is encoded in a suffix to the document identifier, such as A0 or B1, and is used to distinguish between different versions of a document. These can be for example patent application and final granted patent or search reports and other augmentations.

[10]http://gargantua.sourceforge.net/

sections. The resulting alignment statistics for both language pairs can be found in Tables 4.2 and 4.3. For both languages, the claims turn out to be highly parallel with input-output-ratios of 95% and higher for both input languages. English and French abstracts turned out to be highly parallel as well, while French abstracts seemed to be considerably longer than their German counterparts. It is not surprising that French and English descriptions, which stem from different documents, exhibit a significantly lower input-output-ratio for both input languages. All statistics are in accordance with the numbers previously observed for English-German (Wäschle, 2011).

		en		**fr**	
	output	input	%	input	%
abstract	3,697,670	3,918,623	94.36%	4,034,051	91.66%
claims	6,966,851	6,966,851	98.10%	7,030,849	99.09%
description	5,594,745	5,594,745	79.15%	6,158,901	90.84%

Table 4.2: Alignment statistics for **en**-**fr**.

		fr		**de**	
	output	input	%	input	%
abstract	122,440	161,609	75.76%	125,686	97.42%
claims	3,034,007	3,176,191	95.52%	3,151,520	96.27%

Table 4.3: Alignment statistics for **fr**-**de**.

	en		**fr**	
	tokens	types	tokens	types
title	20,361,301	297,148	25,971,983	273,355
abstract	153,499,158	631,104	176,341,220	480,851
claims	509,447,509	1,043,258	578,720,048	918,638
description	229,420,848	865,687	241,304,626	941,973
total	912,728,816	2,072,167	1,022,337,877	1,904,081

Table 4.4: Types and tokens in final **en**-**fr** corpus.

We report the numbers of types and tokens in the final corpus in Tables 4.4 and 4.5. German text typically features a large number of types due to the compounding process that can result in arbitrarily long words. Especially patent titles, which mostly consist of descriptive nouns,

	fr		**de**	
	tokens	types	tokens	types
title	20,207,789	239,180	12,965,785	1,068,837
abstract	7,416,101	62,504	5,852,959	282,274
claims	255,947,775	553,298	207,106,263	2,538,687
total	283,838,214	704,355	226,187,490	3,329,068

Table 4.5: Types and tokens in final **fr**-**de** corpus

have a high fraction of types (Wäschle, 2011). Comparing the average number of tokens per sentence also reveals that claims feature especially long sentences. Figures 4.1 and 4.2 show some metadata statistics for the extracted sentence data. To help researchers split the corpus in a sensible way, we extracted selected meta information from the XML documents and annotated each sentence with the patent identifier of the original document, its family patent identifier – which groups together related documents filed in different countries –, date of publication and classification according to the International Patent Classification[11] (IPC). The majority of the data was published in the decade between 2000 and 2010 – proof of the growing importance of the IP domain and the need for technologies such as SMT and information retrieval to process the large amounts of data with high quality of the output. Due to the different nature of the four genres induced by the patent text sections, we keep the partition by section for for further processing, as detailed in the following section.

4.2 Annotation Procedure

Many approaches for detecting noise in corpora use pseudo-noisy parallel data, i.e. they take a corpus that is known to be parallel and add negative (non-parallel) instances by combining each source with each non-corresponding target sentence (possibly sampling from this set to create a more balanced data set with a larger percentage of positive (parallel) examples), to develop systems that distinguish between parallel and non-parallel data (Munteanu and Marcu, 2005). Others inject noise into a parallel corpus by incrementally adding automatically translated data to a corpus of human translations (Simard, 2014). Instead of artificially producing a noisy data set, our goal is to analyze the types of noise occurring in the corpus we produced using a fully automated pipeline. To this end, we chose the more time-consuming route of producing manual annotations for a sample of the sentence-aligned data. We proceeded in a two-step approach, starting with a preliminary annotation of a small amount of text from all automatically aligned text sections, to gain a first insight, and, after careful analysis, a

[11] http://www.wipo.int/classifications/ipc/en/

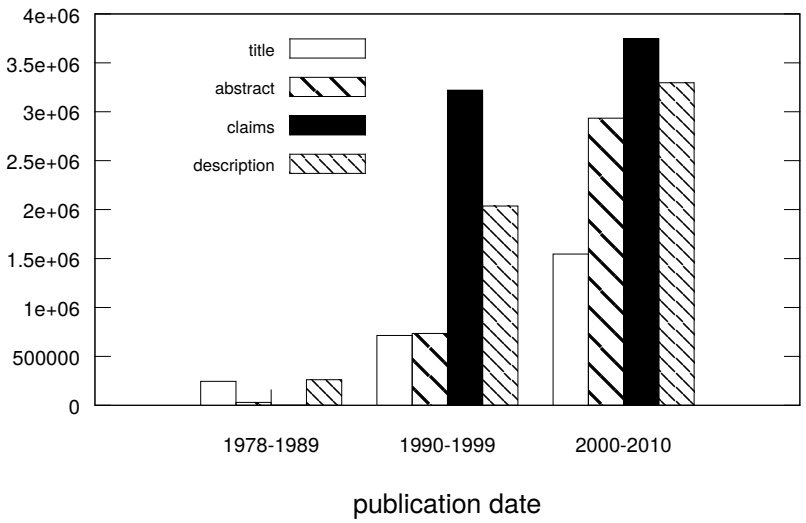

Figure 4.1: Number of **en**-**fr** sentence pairs by publication date.

second annotation of a larger sample of text to produce a data set for learning and evaluating a filtering system.

4.2.1 Preliminary Experiment

We performed a preliminary investigation on the automatically aligned data to get an impression of the amount and type of noise in the data. We were especially interested in non-parallel phenomena that appeared regularly, indicating structural differences between the documents in different languages. To this end, we sampled 85 sentences from three text sections, abstract (15), claims (15) and description (45), for each language pair, and had annotators mark anything notable, i.e. sentence pairs that they considered less than perfect translations. We conducted this procedure on German-English and French-English[12] data. Analyzing the annotations, we found no mentionable noise in abstracts and claims, which confirms our hypothesis that parallel sections extracted from the same document are near-perfect translations of each other. The description data, however, that was collected from separate documents, contained a significant amount of noise. We sum up our observations dividing them into two categories.

Structural noise

- We found multiple instances of sentence numbering, which was present on one side

[12]We did not study the third language pair in PatTR, German-French, because of the lack of sentence-aligned description data.

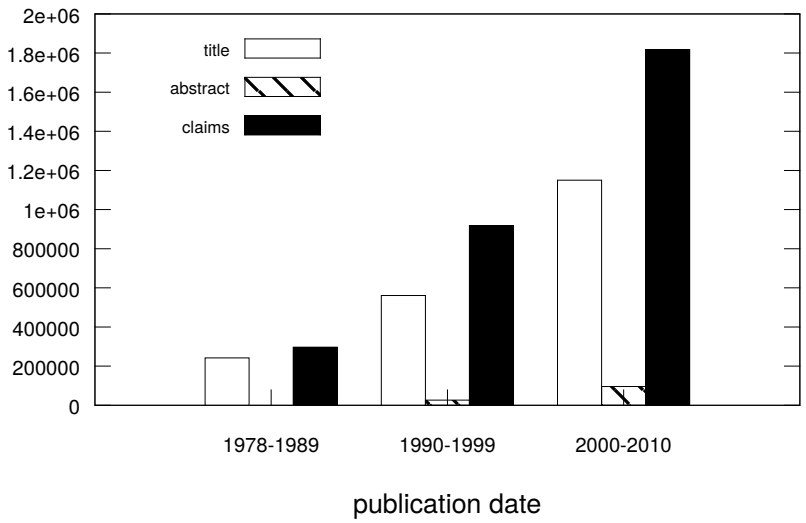

Figure 4.2: Number of **fr**-**de** sentence pairs by publication date.

of the sentence pair but not the other. This was not restricted to one language; German, English and French sentences contained numbering of the format `[XXXX]`, where $X \in 0, ..., 9$, with a respective counterpart missing in the aligned sentence. In some instances, numbering was present on both sides of a sentence pair, but the numbers did not correspond to each other.

- English description sentences were frequently preceded by headlines in upper case font, such as `EXAMPLE`, `DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS` or `SUMMARY OF THE PRESENT INVENTION`, which was not mirrored by the German and French translation. The remainder of the sentence would usually be parallel.

We decided to address both sources of noise by removing all numbering in the format stated above and a fixed set of headlines in upper case letters using regular expressions.

Faulty alignments A different form of noise regards larger non-parallel units of text and can be attributed to the error rate of the tools used for processing the corpus.

- Some sentence pairs were parallel to a degree of only 20% of overlapping content due to faulty sentence splitting on one side of the data. Even though we used a sentence splitter equipped with a list of patent-specific abbreviations, occurrences of unknown abbreviations, e.g. names like `J. Clin.`, caused mid-sentence splits.
- We found that in the French-English data, introductory statements would often not be

en	**fr**
A biosensor for detecting introduction [...]	*L'invention porte sur un biodétecteur détectant l'introduction [...]*
Disclosed are clear, soap-gelled cosmetic [...]	*L'invention concerne des compositions [...]*
The degree of substitution (DS) [...]	*Der Substitutionsgrad (DS = degree of substitution) [...]*

Figure 4.3: Two instances of deviating introductory statements in French and English patent descriptions and an instance of an explanation of an English term in the German sentence.

en *It is, however, necessary to make sure that the gap 16 for the exit of the sugar is maintained.*
de *Eine derartige Verbindungsstelle 19 ist in der Figur schematisch dargestellt.*

Figure 4.4: An instance of aligned sentences with different content.

literal translations, even though a closer translation would have existed. Two examples are given in Figure 4.3.

- In addition to these sub-sentential phenomena, we also found sentence pairs that were of roughly the same length but had almost no content overlap. For an example, see Figure 4.4. These are due to errors made by the sentence aligner.

Table 4.5 shows, how often the phenomena described above appear in the data. Note, that the sample size for our study was very small, so numbers on the whole corpus will differ to some extent. However, the study leaves us with some pointers for how to proceed to improve the quality of the parallel data. In summary, we can state that there are multiple sources of noise in the sentence-parallel data derived from automatically aligning descriptions from two patent documents. We can easily remove structural noise, such as sentence numbering and headlines appearing only on one side of the data, and retain loose translations, that can still be informative. For highly non-parallel sentence pairs resulting from aligner error, we propose an additional filtering step based on a word alignment between source and target. We established that parallel data extracted from abstract, claims and title sections is highly parallel, which makes it a valuable resource. We can estimate lexical translation probabilities on this data and use these to align candidate sentences for filtering. On the alignment, we can compute similarity scores that are combined in a linear model, similar to the model used for recognizing cross-lingual entailment discussed in the previous chapter. To train and develop the model, we need a set of gold standard sentence pairs, which we produce in the next step.

	headlines	numbering	other
en-fr	6.67%	15.56%	15.56%
de-en	13.34%	28.89%	20%

Figure 4.5: Rates of noisy parallel sentence pairs by cause.

4.2.2 Main Annotation

To derive an annotated data set sufficiently large to train and develop a classifier on, 500 French-English[13] sentence pairs were sampled from the corpus. Regular expression filtering as described above was applied to the data before sampling. For testing purposes, the sampling was restricted to description sections from documents published in April 2008. The resulting data set underwent a manual annotation by two expert annotators. Since the sentence aligner produces many-to-one alignments, the data can be considered text rather than to sentences pairs. Annotators were asked to rate the pairs with regard to parallelism and semantic equivalence. They were given the following task description[14].

1. Assign an equivalence score to every pair, ranging from 0% to 100%. Completely differing texts correspond to a score of 0%, texts with identical content in both languages to 100%. We recommend using 20% steps in between or at most 10% steps, if the data indicates this. Small variations, such as *Figure* vs. *FIG.*, comma vs. semi-colon and loose translations are considered as equivalent. Focus on content discrepancy, e.g. an additional or incomplete sentence on one side – depending on the example, this would correspond an agreement with a score of of 40% or 60%.
2. In a second pass, rate the relation between the content of the two texts with one of the following four classes:

   EQUIV (can be derived from an equivalence score of 100%, no explicit annotation necessary)

   ADD$_{en}$ The French text is completely contained in the English text, but the English text features additional content.

   ADD$_{fr}$ The English text is completely contained in the French text, but the French text features additional content.

[13] Since we had to ration resources for annotation, we opted for annotating a large amount of data with two annotators, but only on one sentence pair, as opposed to covering both sentence pairs, but with fewer and less reliable resulting resources. For the CLTE task, De Souza et al. (2013) showed that parameters learned on one language pair transfer to other language pairs with only small impact on accuracy, especially when the languages are similar.

[14] Translated from German by the author.

EQUIV	445	89.20%
ADD$_{en}$	19	3.80%
ADD$_{fr}$	16	3.20%
NONE	19	3.80%
total	500	

Table 4.6: Distribution of classes in final annotation.

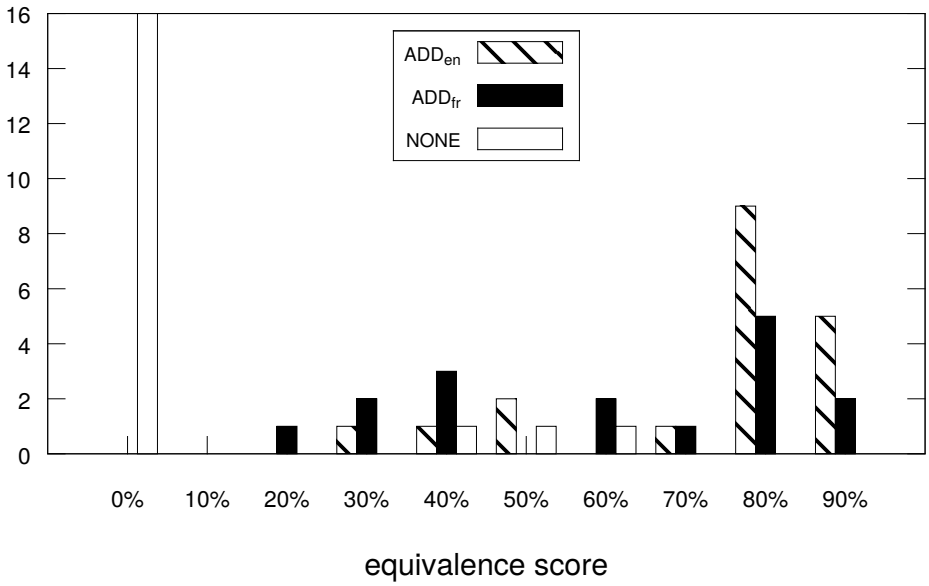

Figure 4.6: Distribution of scores given by the annotation.

NONE (can be derived implicitly from the other three classes, no annotation necessary)

Annotators agreed in 479 cases and disagreed in 21 with an inter-annotator agreement of $\kappa = 0.797$ for the class annotation. The disagreement cases were decided by a third juror[15] to obtain the final annotation. The majority of sentence pairs received a perfect score of 100% and an EQUIV rating (see Table 4.2.2 and Figure 4.6).

[15]The author.

4.3 Adapting a System for Recognizing CLTE for Filtering

To build a system for automatically detecting non-parallel sentences in our data, we employ the pipeline described in Chapter 3. However, we select a subset of the previously used features and make some changes to the individual components. In 3.5 we found that many of the features are redundant. To speed up the feature computation[16] we restrict the feature set to the cross-lingual features (see Figure 3.7), thus saving time, because we do not need to produce translations for all sentences in the training set. To make up for this change, we added some of the features that were included in the monolingual alignment group: features that transfer to the cross-lingual case but did not have a cross-lingual pendant in the CLTE system, such as the Meteor fragmentation score (as chunk ratio). We tuned and developed the system on the 500 hand-annotated sentence pairs. We performed 2-fold cross validation to determine the best system setting in an intrinsic evaluation and applied this system to the filtering task for extrinsic evaluation.

4.3.1 Binarizing the Annotation

Since our annotation with equivalence scores ranging from 0% to 100% turned out to be extremely sparse, we opted not to learn a regression model, but instead decided to transform the problem into a binary classification task, where a classifier learns to distinguish between parallel and non-parallel sentences. It is a non-trivial problem to determining, which sentence pairs should still be considered as parallel, and depends strongly on the data and task at hand. Tillmann (2009) consider sentence pairs as parallel, where at least 75% of source and target words have a corresponding translation in the other sentence. Ştefănescu et al. (2012) mine for sentence pairs that improve the output of an SMT system and find that these can range from parallel and quasi-parallel to strongly comparable. Fung and Cheung (2004) define a lexical score computed on word pairs to quantify, how parallel or comparable a corpus is. They give examples for parallel, noisy parallel, comparable and non-parallel corpora. Utiyama and Isahara (2003) use four categories, *A*, *B*, *C*, and *D* that correspond to overlap ranges of $> 50 - 60\%$, $> 20 - 30\%$, $> 0\%$ and none, to manually assess their automatically obtained alignments. Utiyama and Isahara (2007) employ the same scheme but with slightly different ranges. Lu et al. (2009) asked annotators to classify sentences into three categories, *correct* (literal translation to 80% content overlap without phrasal reordering), *partially correct* (more than 50% content overlap) and *incorrect* (less than 50% content overlap).

We derive a definition for a parallel sentence pair by considering different cutoff values for transforming the equivalence score annotation into a binary annotation. A cutoff value of 0.7 corresponds to an annotation, where sentence pairs with an equivalence score of 70% and

[16] Our final data set consists of millions of instances, as opposed to hundreds in the CLTE task.

more are considered to be parallel. Examples with an equivalence score of less than 70% are subsequently positive examples, all others negative. Figure 4.7 plots the classifier accuracy[17] and F1 score[18] for the positive class (i.e. non-parallel sentence pairs) for all cutoff values between 0 and 1[19] in steps of 0.1. For both measures, the best cutoff value is 0.7, determined with grid search. The result can be read as the threshold, where it makes the most sense to distinguish between pairs above and below in terms of the features.

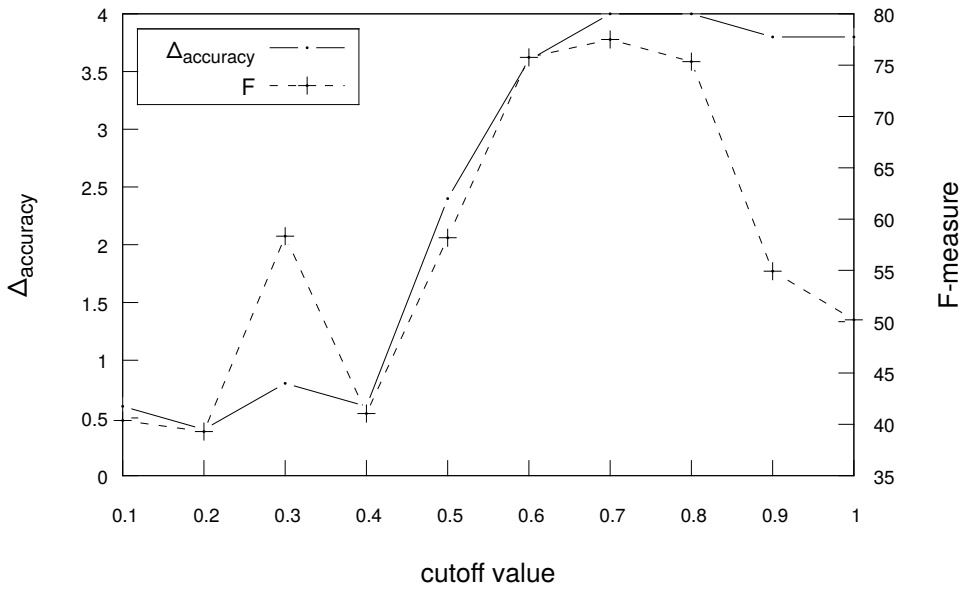

Figure 4.7: F-measure and gain in accuracy for different score cutoff values.

4.3.2 Feature Engineering

In total, we extract 14 features from a sentence pair ($\mathbf{e}$, $\mathbf{f}$), which can be divided into features operating on the level of tokens only,

- length ratio, $\frac{|\mathbf{e}|}{|\mathbf{f}|}$ and $\frac{|\mathbf{f}|}{|\mathbf{e}|}$
- length difference, $|\mathbf{e}| - |\mathbf{f}|$ and $|\mathbf{f}| - |\mathbf{e}|$
- absolute length of $\mathbf{e}$

[17] The number of correctly classified examples, positive and negative, divided by the total number of examples.

[18] We compute the F-measure as the harmonic mean of precision and recall, $F_1 = 2 \cdot \frac{P \cdot R}{P + R}$, where precision P is the number of true positives divided by the total number of elements labeled as belonging to the positive class and recall R is the number of true positives divided by the total number of elements that actually belong to the positive class.

[19] We perform 2-fold cross validation on the annotated training set to determine the best cutoff value.

- absolute length of **f**

computed on both sides of the sentence pair, and features operating on a cross-lingual alignment between source and target, adding lexical information. Alignment features are

- alignment ratio,
- percentage of null alignments,
- chunk ratio,
- and longest contiguous aligned span.

Each alignment feature is calculated twice, once for **e** and once for **f**. For a detailed description of the features see Chapter 3. We standardize feature vectors by subtracting the mean and dividing by the standard deviation for each feature value. Figure 4.8 reports scores for the two feature groups (different **alignments** and **token**-based) separately and in combination. Furthermore, we experimented with different heuristics for symmetrizing the underlying directional alignment. Since we are not interested in high recall but in high alignment precision, instead of using the asymmetrical, recall-driven alignment described in 3.3, we opted to base the cross-lingual alignment features on IBM Model 2 alignments and symmetrize the alignments. We compute the alignments using fast_align on a corpus of 6,202,442 sentence pairs from title and abstract sections, which we assume to be highly parallel on the basis of the preliminary annotation experiment described in Section 4.2.1. The titles could even be considered a domain-specific dictionary, since they contain mostly short nominal phrases. We therefore expect our alignments to be of high quality. We experiment with different symmetrization heuristics.

The *intersection* contains only alignment links common to both the **e**-to-**f** and **f**-to-**e** alignment.

The *union* considers the set of all alignment links in **e**-to-**f** and **f**-to-**e** alignment.

Grow-diag-final-and (gdfa) is a heuristic, which starts from the alignment *intersection* and *grows* this set by adding neighboring[20] alignment points from the alignment *union*. In the *final* step, alignment points between two unaligned tokens are added, even if they are not neighboring an established alignment point. (Koehn et al., 2005)

[20] Think of the alignment matrix as introduced in Chapter 2. Neighboring is defined as left, right, top, bottom or diagonal.

The different alignment strategies are indicated as A_x in Figure 4.8; *forward+backward* means that all alignment features are calculated on both **e**-to-**f** and **f**-to-**e** alignment separately, enlarging the feature space to 22. A_{all} denotes a system that includes alignment features for all different alignment strategies, resulting in 46 total features. The best alignment strategy turns out to be the *intersection* symmetrization heuristic. Combination with the token-based features results in an additive gain, indicating that they contain complementing information. Note that in this task, the *intersection* heuristic, which results in a high-precision alignment with few alignment links is preferable over more recall-oriented methods resulting in a larger set of alignment links, such as the *union*. We observed that the opposite was true for the CLTE task. This shows that choosing a fitting alignment strategy is a crucial step that depends on the targeted task. In the next section, we subject the system with the best parameters according to the presented experiments to an extrinsic evaluation on an SMT task.

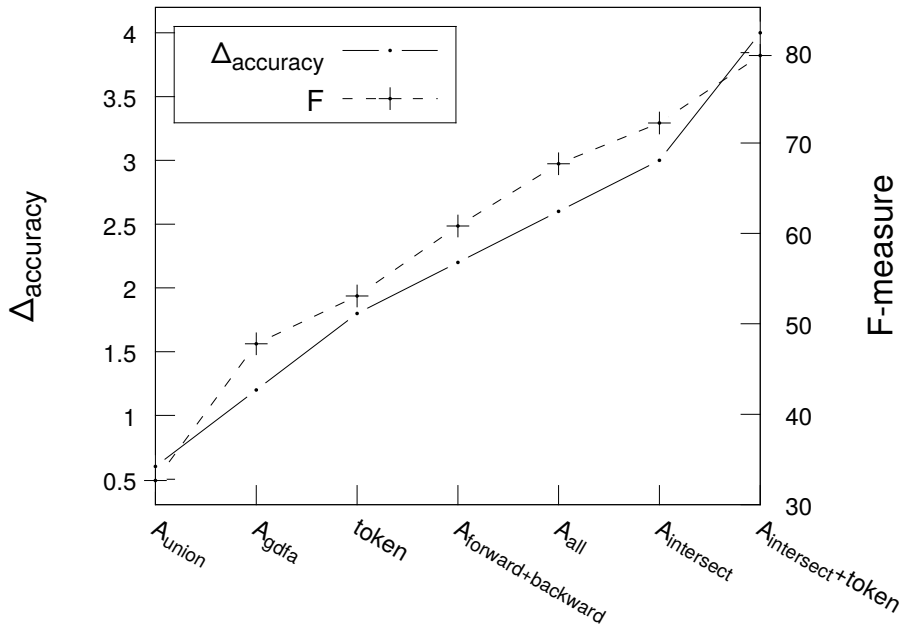

Figure 4.8: F1 score and gain in accuracy for different feature sets, sorted by worst to best performance.

4.4 Experimental Results

We employ the optimized, adapted system to remove the noise from the parallel data extracted from patent descriptions. We then evaluate the influence on SMT quality training translation

models on the filtered data. As test and development set, we chose data from patent abstracts in order to have reliable references for evaluation, while the development set for training the classifier, as noted previously, is comprised of description data. Word translation probabilities are trained on a separate corpus of data from abstracts[21] and titles, which feature no noise. For comparison, we train several baselines as detailed below.

4.4.1 SMT System and Data

The data is split by years of publication. Patent data from the years between 1978 and 2007 is reserved for SMT training, data from 2008 for meta-parameter tuning, testing and training of the filtering system. We build hierarchical phrase-based SMT models for the cdec[22] decoder (Dyer et al., 2010). We obtain word alignments with the fast_align[23] implementation of IBM Model 2 (Dyer et al., 2013). A 5-gram language model is learned with SRILM[24] (Stolcke, 2002) and the weights of the log-linear model are tuned with MERT (Och, 2003).

	patent section	date	sentences	**fr** tokens	**en** tokens
MT-train	description	1978-2007	5,145,292	184,920,614	176,187,963
MERT-dev	abstract	05/2008	2,000	96,384	82,220
MT-devtest	abstract	06/2008	2,000	98,335	84,355
MT-test	abstract	01/2008	2,000	100,323	87,323
filter-dev	description	04/2008	500	22,993	20,036
WTP-train	title, abstract	1978-2007	6,202,442	202,313,203	173,860,459

Table 4.7: Experimental data.

The full SMT training set contains more than 5 million French-English sentence pairs from description sections of patents issued between 1978 and 2007. The filtering methods operate on this data set and produce filtered variants of it, on which the translation model of the MT system is retrained[25]. The language model was trained on the English side of the parallel corpus and remains the same for all systems. Weights of the SMT baseline were tuned on a development set of 2,000 sentence pairs, sampled from abstracts of patents issued in May 2008. These weights are taken as fixed and used by all systems. An additional set of 2,000 sentences from abstracts was used to tune the parameter of the baseline filtering system. We test SMT systems on a set of 2,000 sentence pairs, sampled from abstracts of patents issued

[21] Of course excluding the test and development set
[22] http://http://cdec-decoder.org
[23] http://github.com/clab/fast_align
[24] http://www.speech.sri.com/projects/srilm/
[25] Retraining involves the full pipeline, i.e. word alignment and grammar extraction.

in January 2008, using BLEU to measure translation quality. An overview of the data is given in Table 4.7.

4.4.2 Baselines

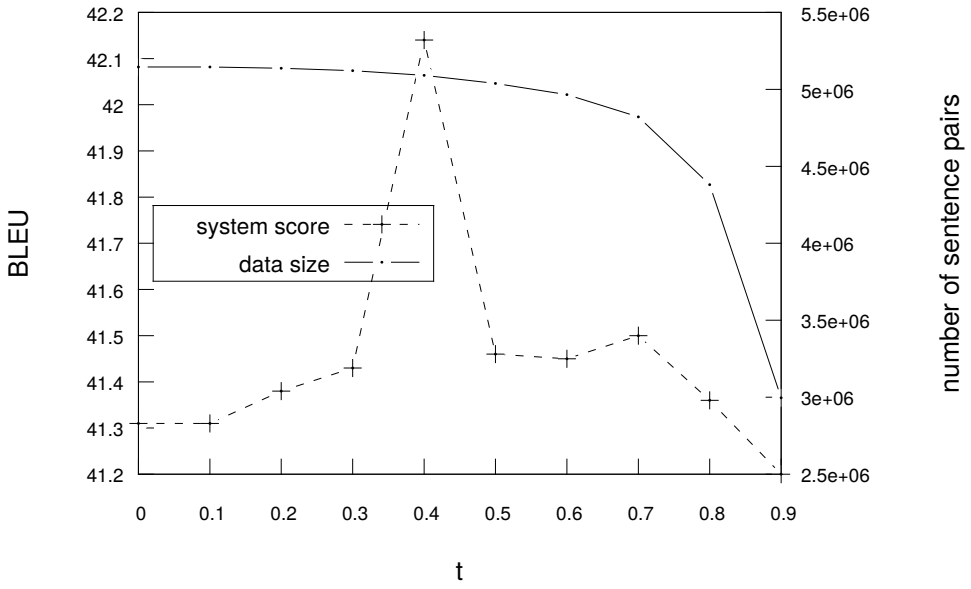

Figure 4.9: Training data size and resulting system BLEU score on development set for increasing length filtering threshold t.

We set up a baseline filter system trained on data, from which asymmetrical (with regard to the sentence length) sentence pairs were removed. The sentence length ratio of a sentence pair $(\mathbf{e}, \mathbf{f})$ is defined as follows:

$$\text{length\_ratio}(\mathbf{e}, \mathbf{f}) = \begin{cases} \dfrac{|\mathbf{e}|}{|\mathbf{f}|} & \text{when } |\mathbf{e}| \leq |\mathbf{f}| \\ \dfrac{|\mathbf{f}|}{|\mathbf{e}|} & \text{when } |\mathbf{e}| > |\mathbf{f}| \end{cases}$$

We perform grid search with steps of 0.1 to determine the best threshold value t, which specifies the minimum sentence length ratio for a pair to be kept in the training set. Table 4.9 shows data set size and BLEU score on devtest set for different values of t. While the data set is reduced to approximately 60% of its original size, only minor variations in BLEU score can be observed that are not significant in all cases but one. We chose the threshold value $t = 0.4$ as optimal setting for the **length filter** system.

In addition, we created a system, where a significant amount of data that was chosen randomly was removed from the training corpus. We trained two **random filter** systems; one on 4.5 million and one on 5 million sentences pairs from the original training set.

4.4.3 Results and Discussion

		devtest		test	
training data	size	BLEU	TER	BLEU	TER
unfiltered	5,145,292	41.31%	42.56	41.72%	42.75
random filter	4,500,000	41.35%	42.54	41.62%	42.85
random filter	5,000,000	41.39%	42.47	41.68%	42.80
length filter ($t = \mathbf{0.4}$)	5,090,663	42.14%	41.57	41.67%	42.72
length filter ($t = \mathbf{0.7}$)	4,821,560	41.50%	42.43	41.69%	42.79
trained filter	4,676,030	41.45%	42.43	41.67%	42.81

Table 4.8: System results on test set.

Table 4.8 shows results on development and test set. Differences between systems are small and not significant and cannot be distinguished from the effects of a random filter. Reduction in size using a trained filter is comparable to using a length-filtering baseline with a threshold of $t = 0.75$, which corresponds to a reduction of the training set size and accordingly a noise rate of about 10%. Our results confirm the findings of Goutte et al. (2012) that SMT systems can tolerate up to 30% of noisy data. In the case of our data this is not surprising, since (1) the amount of text is very large and (2) data from the IP domain can be comparatively repetitive – the latter property will be explored more thoroughly in Chapters 5 and 6. In conclusion, we can state that the originally aligned data, even though collected from comparable, not parallel documents, is of satisfactory quality for SMT training.

CHAPTER 5

Cross-lingual Fuzzy Match Retrieval for SMT

In Chapter 1 we phrased translation as the search for a semantic equivalent of a source sentence in the target language. In this light, the problem of machine translation can be framed as *translation retrieval*: instead of generating a translation from scratch, the given source is input into a retrieval system, which produces target translation candidates, ranked by relevance. Methods to find target language text using source queries are explored in the field of *cross-language information retrieval* (CLIR) (Nie, 2010). Viewing translation as retrieval is related to the concept of *translation memories* (TM), a computational tool used by professional translators to speed up the translation of repetitive texts. We introduced the architecture of translation memories in Chapter 2. Using the input source sentence as query, the TM delivers similar source sentences, which have been previously translated, and returns the corresponding translation. Most commonly, TM systems apply approximate string matching for retrieval, which makes them capable of retrieving not only exact matches, but also partial or fuzzy matches. These might not be perfect translations of the input, but a good starting point that saves translators time and effort. We propose to combine translation memory, cross-lingual information retrieval and statistical machine translation by using CLIR to retrieve fuzzy matches directly in the target language given the source query. SMT is employed to represent the query in the target language and to evaluate the retrieved matches. The idea of combining the strengths of TM and SMT tools has been successfully explored in recent years (Smith and Clark, 2009; Koehn and Senellart, 2010a; Zhechev and van Genabith, 2010; Ma et al., 2011). Instead of generating the whole translation for an input sentence from scratch, the SMT system is asked to translate only the non-matching parts, or translate the sentence with

the translation memory match as bias. In this chapter, we propose a light-weight hybrid TM-SMT system to integrate fuzzy matches found in monolingual corpora instead of bitext. While we use SMT to process the fuzzy matches found with cross-lingual retrieval, an even more promising practical use for cross-lingual fuzzy matching is the area of CAT and post-editing. Human translators have been using translation memories in their workflow for decades; cross-lingual fuzzy match retrieval opens up monolingual corpora as new resources with the opportunity to support translators by providing more and better matches.

In particular, we propose the following novel ideas: We directly perform source-target comparison instead of source-source comparison for fuzzy match retrieval. To make the search feasible on large monolingual corpora in the order of tens of millions of sentences, we use locality-sensitive hashing (LSH) as an efficient coarse retrieval technique to avoid a large number of pairwise comparisons. After generating a set of candidate translations, search is performed at a finer-grained level using cross-lingual similarity measures. Given a fuzzy match in the target language, our model re-ranks a list of the *n*-best translation hypotheses output by an SMT decoder using features that model the closeness of the hypothesis and the match. We are able to use target-language only matches, since our approach to integration does not rely on an alignment between source and target side of the match. We are able to show consistent and significant improvements on several technical domains (IT, legal, patents) for different language pairs (including Chinese, Japanese, English, French, and German). Results for source-target comparison are comparable to or better than using a target-language reference of source-side matches. Furthermore, we show that our approach can improve an SMT system in a setting, where additional monolingual data is available for domain adaptation.

5.1 A Model for Biasing an SMT System with TM Fuzzy Matches

Our integrated model for TM and SMT uses a pipeline approach (Figure 5.1) to combine several components: a TM **retrieval** module yields the best fuzzy match in the target language for a given sentence, and a **re-ranker** uses this information to re-score an *n*-best list of hypotheses output by an SMT **decoder**. For biasing the SMT system, we choose a light-weight re-ranking approach – as opposed to an integration of matches during decoding – for several reason: The pipeline model ensures portability across decoders. Furthermore, it makes integration of external features particularly easy. Thirdly, it does not rely on an alignment between TM source and target, e.g. to extract phrase pairs, which allows us to integrate a novel way to retrieve fuzzy matches. Using CLIR, we can retrieve fuzzy matches not only on bitext, but from monolingual data in the target language, which is usually available in considerably larger amounts than parallel data for a given domain. The hybrid TM-SMT model functions as an extrinsic evaluation of the retrieval approach. In order to be able to include corpora of language model training scale, we opt for two-step retrieval for performance reasons.

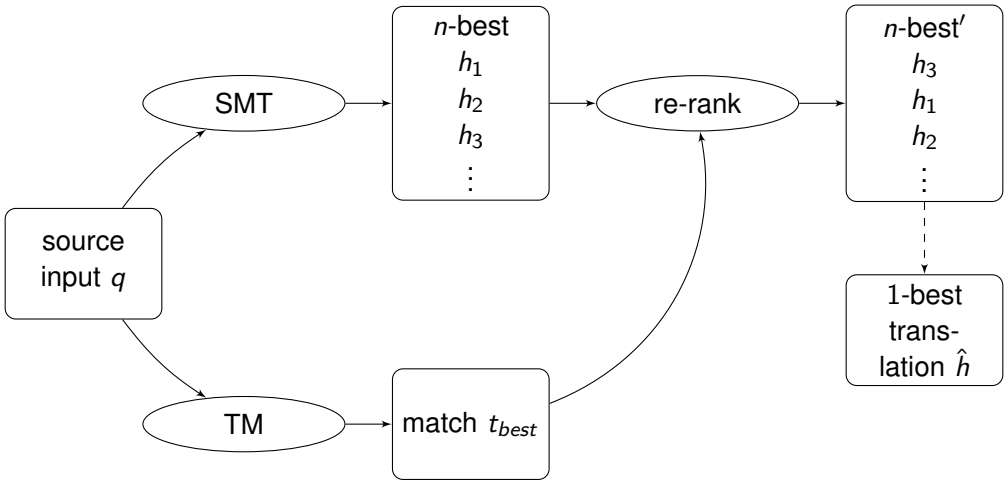

Figure 5.1: Integrated system pipeline. Note, that this overview disregards all connections between SMT and TM, e.g. shared data, translations produced by SMT for TM in the cross-lingual case, etc. The source q is input both into an SMT system and a TM retrieval component – the TM could rely on IR, CLIR or even both –, which returns a fuzzy match in the target language. The SMT system generates an n-best list of translations for the query, which are then re-ranked according to our model, which balances SMT model score and similarity of TM match t_{best} and a given translation option, possibly resulting in a new 1-best translation.

1. A first, coarse look-up step, which is fast, but largely insensitive to global word-order and uses hashing to avoid a pairwise comparison of all match candidates.
2. A refined ranking[1] step, which is slower, but considers word-order of the sentence and is performed only on candidates retrieved by step (1).

Integrating matches into SMT is one practical application for cross-lingual fuzzy match retrieval, but in particular for CAT applications, where translations and matches for new sentences need to be available in less than a second in order to be considered helpful, fast retrieval is key.

In the context of fuzzy match or translation retrieval, we refer to the input sentence as the *query*, denoted with q. The query is available only in the source language. The set of targeted documents is the translation memory. In the default scenario this is a collection of source sentences s with an associated target translation t (parallel data or bitext). We can apply **source-side** comparison or **IR** to assess the similarity of q and different s and return a target fuzzy match $\hat{t}$ that corresponds to the $\hat{s}$ with the highest similarity to q. We extend the TM component to enable **target-side** or **CLIR** retrieval, where the resource for finding possibly similar translations consists only of a collection of target sentences t, by finding the best

[1] Not to be confused with the final re-ranking of the SMT n-best list shown in Figure 5.1.

match $\hat{t}$ by direct comparison with q. We evaluate a number of CLIR techniques for the refined retrieval step; in general, we use SMT to produce a target language representations of the query and compute a cross-lingual similarity score. As a byproduct, we investigate a technique for integrating the matches during decoding with a hierarchical phrase-based SMT system, skipping the final re-ranking step.

5.1.1 Related Work

Work on integrating translation memory (TM) and machine translation (MT) can be divided into approaches at the sentence level, that decide whether to pass MT or TM output to the user (He et al., 2010a,b; Dara et al., 2013), and approaches that merge both techniques at a sub-sentential level (Smith and Clark, 2009; Koehn and Senellart, 2010a; Zhechev and van Genabith, 2010; Ma et al., 2011). Results are either measured in terms of retrieval precision or post-editing effort (CAT perspective), or using standard MT metrics (end-user perspective). The first class of approaches can be summarized as *translation recommender systems*, and are closely related to the task of MT quality estimation (Ueffing et al., 2003; Blatz et al., 2004; Specia et al., 2009), where post-editing effort is predicted. The overall goal is to improve a CAT pipeline, so the success of the approaches is judged from the translator's perspective. He et al. (2010a) propose a system that decides for a given source sentence, whether to pass a TM fuzzy match or the output of an SMT system to a post-editor. The problem is framed as binary classification; TER is used to approximate human judgments of recommendation quality. Features include SMT score, language model scores and fuzzy match score. In this approach, the SMT system is treated as a black box and translation alternatives are not considered. He et al. (2010b) extend their previous work by moving from binary prediction to ranking. The best matches from a TM according to fuzzy match score are merged with the n-best list output by an SMT model. To this end, features have to be independent from the respective systems, so the feature set is reduced to language model and IBM Model 1 scores. As for the previous model, the gold standard ranking is created using TER in absence of human judgments to approximate post-editing effort. Experimental results are presented in terms of IR measures, e.g. precision and recall.

A variety of models exist for combining TM and MT on a sub-sentential level. Most approaches focus on improving the SMT system using TM resources and methods, so success is measured in terms of MT quality using standard automatic metrics, such as BLEU or TER. In one line of research, the global SMT models remains static, while the model for the current sentence is biased locally with the help of fuzzy matches. Biçici and Dymetman (2008) developed Dynamic Translation Memory (DTM). They identify matching sub-sequences between the current sentence and a TM fuzzy match and use the combination of source and target with corresponding alignment to construct a non-contiguous biphrase, which is added to the

grammar for the current sentence with a strong weight. The decoder is then run as usual using the augmented grammar. Koehn and Senellart (2010a) employ a similar technique to extract translations for a whole sentence that contain gaps, where mismatches occur. Translations for matching parts are passed to the decoder (a) using XML-like mark-up of phrases in a phrase-based system and (b) by constructing large grammar rules for a hierarchical SMT system. Zhechev and van Genabith (2010) also use the Moses XML mark-up option to pass translations for matching phrases to decoder, but use sub-tree alignment between source sentence, TM source and TM target to extract translations for the matched parts. Ma et al. (2011) extend the work by Koehn and Senellart (2010a) and use discriminative learning to decide if a particular fuzzy match should be used to restrict the target translation. All approaches have in common that the SMT system is forced to translate only the non-matching segments, either by restricting translation via XML mark-up or by adding high feature weights to force the application of the rules or biphrases extracted from the TM match. Our approach is similar to these approaches in that the global MT system remains static; however, all presented approaches make use of the alignment between source and target of the fuzzy match, while our approach uses only the target side to bias translation, making it possible to use matches, for which only the target is available – think of a language model. Furthermore, by interpolating between model score and features measuring the similarity of a hypothesis and a match, the model controls the influence of the TM instead of forcing the application of the match.

A more adjustable integration of TM and SMT system is possible by adding one or more tunable features to the translation model in order to control the influence of TM matches during decoding. Such features can encode the quality of a match, e.g. by making use of the fuzzy match score. Simard and Isabelle (2009) extract phrase pairs from the fuzzy match and insert them into the phrase table, then add a similarity feature, which is used during re-scoring, to promote translation hypotheses that are close to the match. Since the calculation of the features on all hypotheses in the search space is not feasible, they resort to re-ranking the n-best list. Our re-ranking approach is similar, with the novelty of using not only matches found by querying the source side of the corpus but also the target. Wang et al. (2013) add several features to the translation model, which encode source and target matching status, handling multiple target translation options for a single source and reordering information. In an extension of this work, Wang et al. (2014) consider a real-life scenario, where TM and SMT training data diverge, for which the method proposed by Wang et al. (2013) is unsuited, since phrases found in the TM database might not be present in the SMT phrase table. To make up for this, they add segments to the phrase table in a fashion similar to Biçici and Dymetman (2008). Li et al. (2014) build on the work by Wang et al. (2013) and transform their features into feature functions. Furthermore, they extend the approach to handle multiple fuzzy matches by finding specific matches for each phrase in the sentence.

While fuzzy matching based on edit distance is the industry standard (Sikes, 2007), differ-

ent techniques for retrieving fuzzy matches from a translation memory have been proposed (Koehn and Senellart, 2010b; Simard and Fujita, 2012; Bloodgood and Strauss, 2014). Koehn and Senellart (2010b) use suffix arrays and *A** search to speed up the look-up process for fuzzy matches. Bloodgood and Strauss (2014) compare edit distance and weighted *n*-gram precision for fuzzy matching and evaluate their usefulness by collecting relevance judgments from human translators via Amazon Mechanical Turk. Simard and Fujita (2012) experiment with different MT evaluation metrics as similarity functions. Moving away from surface string matching, Vanallemeersch and Vandeghinste (2014) enrich the query with syntactic knowledge. Gupta and Constantin (2014) present an approach for incorporating paraphrases during the look-up. Similarly, our approach can be considered as a contribution towards a semantic translation memory. Cross-lingual match retrieval can make use of the whole search space that is constructed when an SMT system translates the source query and thus integrate multiple translation alternatives, i.e. paraphrases and synonyms in the target language. Methods for finding fuzzy matches in the target language can be derived from *cross-language information retrieval* (CLIR), in order to find text in the target language using a query in the source language (Nie, 2010). Locality-sensitive hashing for computing cross-lingual pairwise similarity was investigated by Ture et al. (2011), who find an effectiveness-efficiency trade-off, which has to be taken into account when using this technique in practical applications. A two-step approach with fine-grained cross-lingual matching based on translation lattices has been used by Liu et al. (2012) and Dong et al. (2014) for translation retrieval. Liu et al. (2012) propose to search a monolingual target-language corpus for translations of a given sentence, framing translation retrieval as a special instance of CLIR. They combine a phrase-based translation and a vector space retrieval model. Each *k*-best[2] list hypothesis output by the translation model is used to query the retrieval model; the resulting translation candidates are ranked according to the mixture (product) of translation score and retrieval score. The mixture weights are not tuned but manually set. They report precision for translation retrieval, but no SMT evaluation. Following up on this work, Dong et al. (2014) retrieve target side translation candidates using a lattice representation of possible translations of a source sentence instead of *k*-best list hypotheses. Similar to Liu et al. (2012) they perform a coarse retrieval step using all distinct words in the lattice as a query, followed by a fine-grained ranking of the retrieved candidates based on a score calculated on the lattice representation, using SMT and IR features derived from BLEU. They find that translation lattices significantly outperform retrieval based on *k*-best lists in terms of retrieval precision. The system is successfully applied to the task of identifying parallel sentences. Finally, the hypergraph-based models for CLIR used for fine-grained matching in this chapter are similar to the bag-of-words forced decoding approach by Hieber and Riezler (2015), but were developed independently.

[2]Values for *k* are 1 and 10.

5.2 Monolingual and Cross-lingual Fuzzy Match Retrieval

In the following, we will describe how we retrieve fuzzy matches from the translation memory resources using a combined technique of coarse and fine-grained retrieval. The overall protocol is the same for source and cross-lingual retrieval. In CAT practice, how well two sentences match is calculated using a so-called fuzzy match score (Sikes, 2007) on sequences u and v,

$$\text{FMS}(u, v) = 1 - \frac{\text{ED}(u, v)}{\max\limits_{|u|,|v|}}$$

where ED is the edit or Levenshtein distance (Levenshtein, 1966). Levenshtein distance is defined as the minimum number of operations[3] needed to transform the sequence $u = u_1, ..., u_i, ...u_m$ into the sequence $v = v_1, ..., v_j, ...v_n$. The minimum edit distance can be computed with dynamic programming using the Wagner-Fischer algorithm, which uses the following matrix recursion.

$$M_{i,j} = \begin{cases} \max(i, j), & \text{if } i = 0 \text{ or } j = 0 \\ \min \begin{cases} M_{i-1,j} + 1 & \text{del.} \\ M_{i,j-1} + 1 & \text{ins.} \\ M_{i-1,j-1} + cost(u_i, v_j) & \text{sub.} \end{cases}, & \text{otherwise} \end{cases} \tag{5.1}$$

where

$$cost(u_i, v_j) = \begin{cases} 1, & \text{if } u_i = v_j \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

The matrix M stores the distance between all possible prefixes of u and all possible prefixes of v and is filled from left to right and top to bottom. M then contains the minimum number of edit operations for the complete sequences in the last cell.

$$\text{ED}(u, v) = M_{|u|,|v|} \tag{5.3}$$

The complexity of computing the minimum edit distance with this procedure is $O(|u||v|)$. Note, that calculation of FMS is based on the sequence of tokens that forms a sentence rather than the sequence of characters. Although FMS is the industry standard, different methods have been proposed to select fuzzy matches. Bloodgood and Strauss (2014) compared percentage

[3] Allowed operations are deletion, insertion or substitution of characters.

match, edit distance and n-gram precision for translation memory retrieval. They found that a weighted n-gram precision measure actually correlates best with human judgments and most often retrieves the best rated match. We use FMS in the baseline, source-side retrieval scenario, but experiment with different similarity measures in the cross-lingual setting.

Computing edit distance or n-gram overlap against a corpus of tens of millions of sentences is too slow for real-time use in an interactive translation environment, even when using efficient data structures and lower bounds[4] to discard poor candidates early – especially considering long sentences with more than 100 tokens that appear for example in patent text. Methods for efficient approximate string matching have been proposed for extracting hierarchical translation rules from a large corpus on-the-fly, for example suffix arrays (Lopez, 2007) and most recently, GPU computing (He et al., 2015). In practice, TM or related retrieval systems often make use of two-step approaches with a coarse pre-retrieval that selects candidates for good fuzzy matches (Smith and Clark, 2009; Liu et al., 2012). The exact fuzzy match score or a related similarity measure can then be computed for a smaller candidate set in reasonable time. We adopt this approach and choose a fast method for approximating the pairwise similarity of items: locality-sensitive hashing (LSH). With LSH, similar items hash to the same bucket and a large number of comparisons, i.e. between query and every sentence in the translation memory, is reduced to repeated computation of hashes on the query and a lookup in a hash table. It is most commonly employed for tasks such as near-duplicate detection of websites, but fits our task very well. Ture et al. (2011), for example, used LSH for CLIR mate-finding. However, using LSH comes at the cost of getting an approximation with a certain error instead of exact results, but we believe, that in our application, finding a good fuzzy match might be just as helpful as finding the global best fuzzy match. We use MinHash, an instance of LSH, to obtain a set of fuzzy match candidates. The best match from this set is selected by maximizing a fine-grained similarity measure (CL)IR – named so because we define both a monolingual (IR) and cross-lingual (CLIR) variants – that quantifies the similarity of the query q_i and a match candidate.

5.2.1 Locality Sensitive Hashing for Coarse Translation Retrieval

MinHash (Broder, 1997) is an instance of LSH that estimates the similarity of two documents by reducing the dimensionality of the document signature using sampling. MinHash approximates the Jaccard similarity JC of two sets A and B,

$$\mathrm{JC}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

by generating fixed-length signatures of each set, from which the Jaccard similarity can be

[4] E.g. $\mathrm{ED}(u, v) \geq |u - v|$.

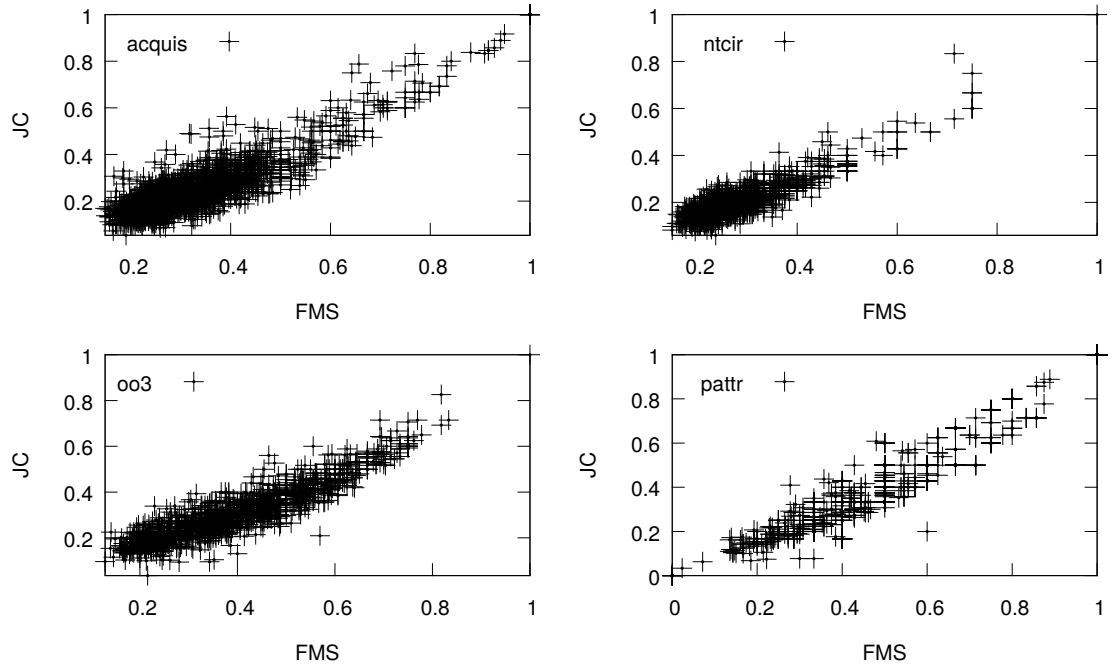

Figure 5.2: Correlation of fuzzy match score and Jaccard similarity on four data sets. The plot shows the fuzzy match score between each sentence in the test set and its best match (according to FMS) in the training set and their Jaccard similarity.

estimated. The signature is obtained by repeatedly hashing each member of the set and storing only the minimal resulting hash. By representing each sentence as a set of *n*-grams we can use this technique to efficiently approximate the *n*-gram overlap of two sentences, which is a good predictor of fuzzy match quality (Bloodgood and Strauss, 2014). Jaccard similarity and Levenshtein distance are related: $\text{ED} = 0 \Rightarrow \text{FMS} = 1 \Rightarrow \text{JC} = 1$, but the opposite direction is not true and for matches lower than 100%, only a broad correlation can be assumed (see plots in Figure 5.2). This means that we need to lower the threshold of the coarse retrieval to avoid low recall.

We implement the MinHash scheme following the description by Rajaraman and Ullman (2012, Chapter 3), but change it, where necessary, to fit the application and make the best use of the available computational resources. As a the first step, tokenized sentences are transformed into sets of *n*-grams[5]. The members of each set are repeatedly hashed using different parameters and the minimal resulting hash is stored for each hash function. This results in a signature vector of fixed length, which corresponds to the number of hash functions used.

[5]Rajaraman and Ullman (2012) call them shingles.

Hash functions We use hash functions of the form

$$hash(x) = (a \times x + b) \bmod v$$

where a and b are randomly chosen positive integers and v corresponds to the vocabulary size, the number of distinct n-grams. In order to avoid collisions, b and v have to be coprime, meaning their greatest common divisor is 1. The number of different hash functions used to compute a minimal hash is equivalent to the sample size. A larger sample size corresponds to a smaller error bar, but increases computational cost in both speed and memory: we have to compute more hashes and increase the size of the signature vectors and the resulting hash tables. The expected error for k hash functions is $O(1/\sqrt{k})$. In out experiments, we set $k = 200$ resulting in an expected error ≈ 0.07.

Signature computation We modified the algorithm given in Rajaraman and Ullman (2012, Chapter 3) to fit our needs. Since we want to be able to compute MinHash signatures continuously for a stream of incoming new sentences, we store the sampled hash function parameters and transform the signature matrix computation into an incremental procedure. Pseudocode for the signature computation for a single sentence is given in Algorithm 1. The sentence-wise computation makes parallelization straightforward. To handle the size of the signature matrix, we use memory-mapped files, which can be accessed by all workers in parallel. The signatures for a large set of documents can be computed in advance and signature computation for a query is performed in milliseconds.

Storing and Comparing Signatures To efficiently estimate the Jaccard similarity from the signatures without pairwise comparison, we apply the banding technique described in Rajaraman and Ullman (2012, Chapter 3). The basic idea is that similar items hash to the same bucket with higher probability than dissimilar items. This technique will result in some *false positives*, but greatly reduces the candidate space that has to be checked for similarity. The signatures are divided into smaller contiguous portions, called *bands*, which are then hashed to a large number of buckets. When two signatures hash to the same bucket, they are considered candidates for meeting the similarity criterion. This criterion is set in advance by adjusting a similarity threshold t through the number of bands and the number of items each band contains. Let S_A and S_B be two MinHash signatures of length k. Divide the signatures into p bands of r entries each so that $p \times r = k$. The probability that the two signatures agree in a single entry is their Jaccard similarity $\mathrm{JC}(S_A, S_B)$, so the probability that they agree in at least one band can be derived as follows.

Algorithm 1

procedure MINHASH(sentence $\langle s \rangle$, list of hash parameters $\langle h \rangle$, vocabulary size v)
  $\langle signature \rangle \leftarrow \langle\rangle$
  split $\langle s \rangle$ into n-grams
  $i \leftarrow 0$
  for all n-grams ng in $\langle s \rangle$ **do**
    $minHash \leftarrow \infty$
    for all hash function parameters (a, b) in $\langle h \rangle$ **do**
      $hash \leftarrow$ HASH(ng, a, b, v)
      if $hash < minHash$ **then**
        $minHash \leftarrow hash$
      end if
    end for
    $\langle signature \rangle[i] \leftarrow minHash$
    $i \leftarrow i + 1$
  end for
  return $\langle signature \rangle$
end procedure
procedure HASH(ng, a, b, v)
  return $(a \times ng + b) \bmod v$
end procedure

$$P(S_A \text{ and } S_B \text{ agree in every entry of band } i) = \text{JC}(S_A, S_B)^r \Rightarrow$$
$$P(S_A \text{ and } S_B \text{ disagree in at least one entry in band } i) = 1 - \text{JC}(S_A, S_B)^r \Rightarrow$$
$$P(S_A \text{ and } S_B \text{ disagree in all bands}) = (1 - \text{JC}(S_A, S_B)^r)^p \Rightarrow$$
$$P(S_A \text{ and } S_B \text{ agree in at least one band}) = 1 - (1 - \text{JC}(S_A, S_B)^r)^p$$

The function $1 - (1 - \text{JC}(S_A, S_B)^r)^p$ plots an S-curve. The placement of the steepest rise functions as a threshold that determines which pairs are likely to become candidates and which pairs are not, and depends on the parameters p and r. Figure 5.3 plots the function for several values of p and r. To position the rise near a chosen threshold t for $\text{JC}(S_A, S_B)$ we use the approximation $t = (1/p)^{1/r}$ (Rajaraman and Ullman, 2012, Chapter 3). To obtain the number of bands p we can transform the equation as follows.

$$t = \frac{1}{p}^{\frac{1}{r}} \Leftrightarrow \sqrt[\frac{1}{r}]{t} = \frac{1}{p} \Leftrightarrow t^r = \frac{1}{p} \Leftrightarrow \frac{1}{t^r} = p \Leftrightarrow t^{-r} = p$$

The algorithmic procedure to set parameters r and p to partition signatures of length k so that candidates with a Jaccard similarity $\text{JC}(S_A, S_B) \geq t$ are retrieved with high probability

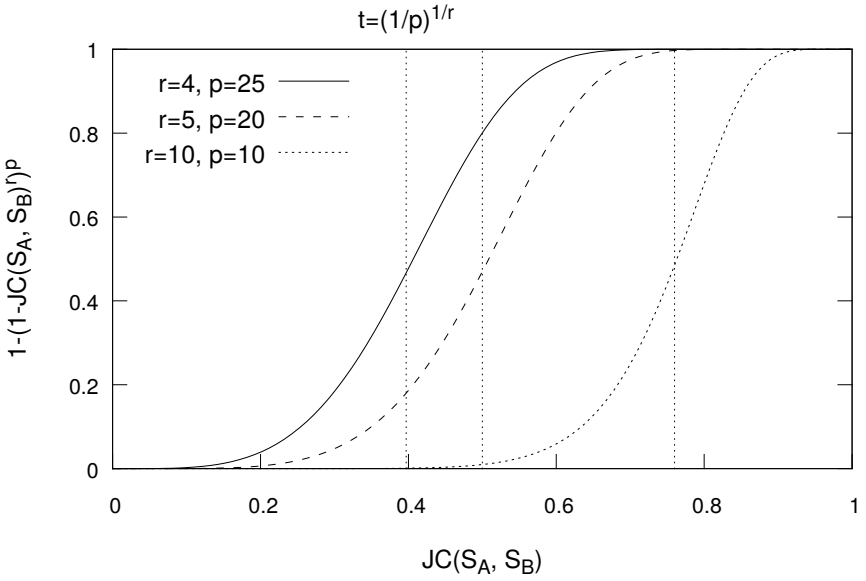

Figure 5.3: S-curves for different parameter settings.

is detailed in Algorithm 2. In setting t we are faced with an effectiveness-efficiency trade-off (Ture et al., 2011), where we find false positives, which slow down the second retrieval step, and also false negatives, which will cost overall performance. We set the threshold t for the Jaccard similarity for each data set on a held-out development set by choosing a setting in which a candidate match is returned for at least 90% of the sentences.

We use a data structure server to store the hashes for the translation memory, which can be efficiently queried when receiving new sentences to translate. We opted for the open-source software Redis[6], since it comes with a well-documented Python API. Signatures for all sentences in the translation memory are precomputed and the hashes for each band stored in separate databases. We compute signatures for queries on the fly and query the hash of each band against the respective database. If a sentence identifier is found in the database for this hash, the sentence is added to the candidate set of matches for the query. We then compute the actual Jaccard similarity for the set of match candidates returned for a similarity threshold setting t and rank them accordingly. We take the 100 best matches for each query and choose the best match in the fine-grained step described in the following.

[6] http://redis.io/

Algorithm 2

procedure FINDNUMBEROFBANDS(similarity threshold t, signature length k)
  $r \leftarrow 0$
  repeat
    $r \leftarrow r + 1$
  until $t^{-r} > k/r$ **or** $r = k$
  if $r > 1$ **and** $|k/(r-1) - t^{-(r-1)}| > |k/(r) - t^{-r}|$ **then**
    $r \leftarrow r - 1$
  end if
  return $\lceil k/r \rceil$
end procedure

5.2.2 Fine-grained Selection of the Best Match

The coarse LSH-based retrieval provides us with a set of up to 100 candidates ranked by their Jaccard similarity to the query. Even when using higher order n-grams, this step is largely ignorant of word order and especially global ordering. For choosing the actual best match from this subset we therefore employ methods more sensitive to word order, such as fuzzy match score. We experiment both with a baseline TM setup, where we look for matches in a sentence-aligned corpus computing similarity between the source input (query) and the source side of the sentence pairs in the TM, and our novel cross-lingual setup, where we compute similarity between the query in the source language and target candidates directly.

IR: Source-Source Matching

In the baseline or IR case, choosing the best TM match amounts to selecting the sentence pair $(s, t)_{i,best}$ from the coarse candidate set $\mathrm{LSH}(q_i)$ that achieves the highest fuzzy match score FMS of the (source) query q_i against the source side $s_{i,j}$ of the TM pair, and returning its target side $t_{i,j}$.

$$(s, t)_{i,best} = \underset{(s,t)_{i,j} \in \mathrm{LSH}(q_i)}{\operatorname{argmax}} \mathrm{FMS}(q_i, s_{i,j}).$$

Since computing edit distance is expensive, even when using dynamic programming to speed up the computation, we make use of a lower bound of edit distance: it is at least the difference of the length of the two sequences. While iterating over all candidates returned by the retrieval, we keep track of the current best candidate and score. If this number yields a smaller fuzzy match score for the current candidate than the current best candidate, we can skip the computation for the current candidate altogether. Also, if the current number of minimum edits exceeds the maximum number of edits needed to yield a larger fuzzy match score than the

current best candidate, the computation is stopped and the candidate discarded.

CLIR: Source-Target Matching

For the cross-lingual setup, however, this step is less straightforward. We would like to select a target sentence from a set of target-only candidates given a query in the source language. To generate a target candidate set with coarse retrieval we use the 1-best translation $Tr(q_i)$ by an SMT decoder trained on bilingual data to represent the query[7]. In the fine-grained matching step, we then compute a cross-lingual similarity score CLIR between the original source query q_i and all target candidates $t_{i,j}$ returned by LSH, and choose the candidate which maximizes this score as the best TM match.

$$t_{i,best} = \underset{t_{i,j} \in \mathrm{LSH}(Tr(q_i))}{\operatorname{argmax}} \mathrm{CLIR}(q_i, t_{i,j}).$$

We investigate several instances for CLIR, applying methods from cross-language information retrieval. We operate on different representations of the query q_i; in particular a direct translation, a lattice representation of the whole search space and a projection using word translation probabilities. Figure 5.4 aims to provide an overview over all dependencies of the model components in the cross-lingual setting.

1-best Fuzzy Match Score Similar to the source-side case, this model uses as a selection criterion the fuzzy match score of the candidate $t_{i,j}$ given the most likely translation hypothesis produced for the query q_i by an SMT model, $Tr(q_i)$.

$$\mathrm{CLIR}(q_i, t_{i,j}) = \mathrm{FMS}(Tr(q_i), t_{i,j})$$

This corresponds to a direct translation baseline in cross-language information retrieval.

Cross-lingual Fuzzy Match Score Instead of using SMT to translate the query, it is possible to directly compute an edit-distance-based similarity between query and match,

$$\mathrm{CLIR}(q_i, t_{i,j}) = \mathrm{FMS}_{cl}(q_i, t_{i,j})$$

[7] Alternative ways to construct queries are possible. We experimented with query expansion by adding translations found in the n-best, but found that using the 1-best translation prediction of the baseline system yielded superior results. The reason is likely that query and result tend to be of similar length, so when terms are added to the query the technique favors longer sentences. We can also conclude that the SMT baseline is quite strong, so a bias towards the best translation under this model is helpful.

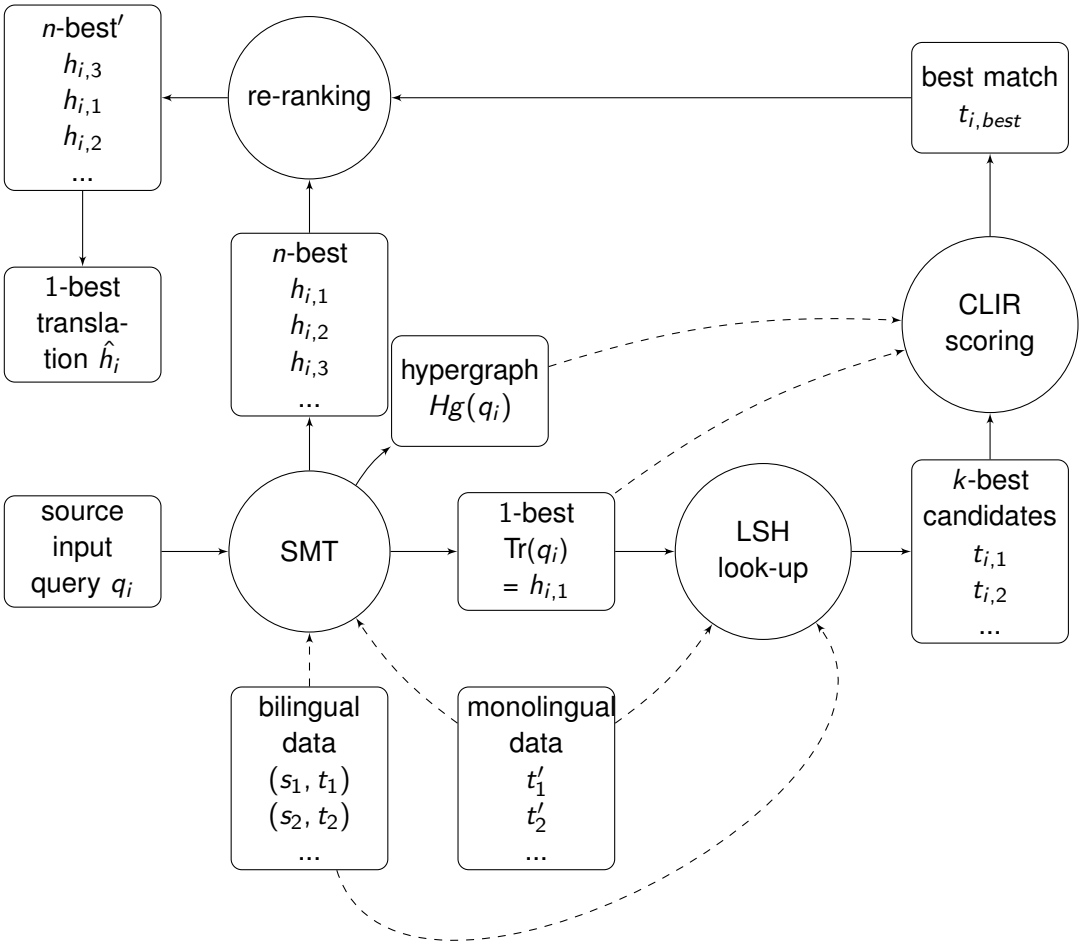

Figure 5.4: Schematics of full pipeline with target-side retrieval. Note, that it is also possible (but not pictured) to include additional data for the match retrieval that deviate from the SMT training data. CLIR summarizes several techniques that use different resources to represent the query, i.e. either hypergraph or 1-best translation.

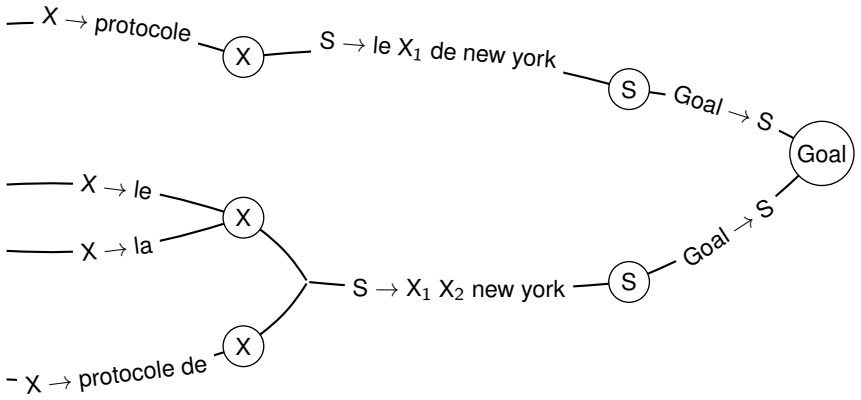

Figure 5.5: Hypergraph encoding two different derivations of the translation *le protocole de new york* as well as the derivation **la protocole de new york*. Note that the hypergraph edges actually correspond to synchronous rules, but only the target side of the rules is pictured.

We define a cross-lingual extension of the fuzzy match score, FMS_{cl}, based on IBM Model 1 lexical translation probabilities (Brown et al., 1993). We use a weighted variant of Levenshtein distance as defined in Equation 5.1, to incorporate the model: instead of using the indicator function given in Equation 5.2, which returns cost 0 for a match and cost 1 for a mismatch when considering a replacement operation, we adapt the cost function to reflect the strength of association between token u_i^s in the source and token v_j^t in the target language, by including the word translation probability $P(v_j^t|u_i^s)$ as a weight.

$$cost(u_i^s, v_j^t) = 1 - P(v_j^t|u_i^s)$$

The word translation probabilities can be estimated from a parallel corpus.

Hypergraph Score In addition to distance-based models featuring a static representation of the query, we explore two methods that operate on the full search space for the translated query (Dong et al., 2014). Different SMT derivations can be efficiently encoded in a translation lattice. We can enrich the translation lattice of the current input sentence, which corresponds to the query, with *n*-gram features that indicate the overlap status between the current derivation and a given TM match – in addition to the default SMT features. Dong et al. (2014) used this technique for translation retrieval using two different lattice representations produced by the Moses decoder. We adopt their approach for the `cdec` hypergraph. A hypergraph is a generalized graph in which a single edge may connect any number of nodes. A subset of hypergraphs[8] is isomorphic to a (synchronous) context free grammar, so the set of derivations

[8]Directed, acyclic and every edge has exactly one head, but might have zero or more tail nodes – a tree.

H_{q_i} for a given input q_i in hierarchical phrase-based SMT can be encoded by a hypergraph $Hg(q_i)$. Nodes correspond to terminal and non-terminal symbols and hyperedges to rules and associated feature values (for an example see Figure 5.5). A hyperedge e is scored by computing the dot product of the feature values $\phi(r_e)$ associated with its corresponding rule r_e and the global model weights w.

$$score(e) = w \cdot \phi(r_e)$$

All translation hypergraphs feature a goal node n_{goal}, which corresponds to the root of the parse tree. A translation hypothesis then corresponds to a path, which is a sequence of hyperedges, through the graph ending at the goal node. The path with the highest probability under the model, the Viterbi path, can be found with dynamic programming and semiring parsing (Goodman, 1999). A semiring is a tuple $\langle A, \bigoplus, \bigotimes, \overline{0}, \overline{1} \rangle$ consisting of an additive and a multiplicative operator, $\bigoplus$ and $\bigotimes$, and their respective neutral elements, $\overline{0}$ and $\overline{1}$. Instantiating these with $\langle \mathbb{R}_0^1, max, \times, 0, 1 \rangle$ and computing the inside score

$$\alpha(n_{goal}) = \bigoplus_{h \in Hg(q_i)} \bigotimes_{e \in h} score(e),$$

of the goal node with dynamic programming on the hypergraph yields the Viterbi score of the best derivation $\hat{h}$. Different instantiations of the semiring result in the Viterbi derivation or the inside probability of the whole hypergraph. We can add features to this model by extending the scoring function of a rule r given a match m. In addition to the default SMT features ϕ_{smt} and weights we add features ϕ_{match} which indicate the match status at each rule.

$$score(e, m) = w_{smt} \cdot \phi_{smt}(r_e) + w_{match} \cdot \phi_{match}(r_e, m)$$

We can use the Viterbi score as a cross-lingual similarity measure between the query hypergraph $Hg(q_i)$ and each match candidate $t_{i,j}$, i.e.

$$\text{CLIR}(q_i, t_{i,j}) = \max_{p \in Hg(q_i)} \sum_{e \in p} w_{\text{SMT}} \cdot \phi_{\text{SMT}}(r_e) + w_{match} \cdot \phi_{match}(r_e, t_{i,j}))$$

where p is a path through the hypergraph – corresponding to a derivation – and e the set of hyperedges on the path. We explore two different ways to incorporate match features $\phi_{match}(r_i, t_{i,j})$ in addition to the SMT feature set $\phi_{\text{SMT}}(r_i)$, using n-gram based measures.

Unigram Precision Features We based our match status feature on the MT evaluation metric BLEU, which combines n-gram precision and a brevity penalty. Since n-gram features are non-local, meaning they do not decompose over the hyperedges, the size of the hypergraph grows considerably when adding features for orders higher than $n = 1$ (Chiang, 2007). We therefore restrict our model to a unigram precision feature and a brevity penalty. The latter is only active on the goal node, i.e. on full derivations. The unigram precision feature indicates at every hyperedge how many nonterminals in the current rule are also present in the match, normalized by the length of the current derivation. With this model, we can assign a score to each match by decoding the input sentence with the extended model and each given match. The best match is then the one that results in the highest Viterbi score under the augmented model. The model was implemented using the cdec python interface (Chahuneau et al., 2012). To set the weights, we opted not to re-tune the SMT feature weights but to keep them fixed to the baseline configuration determined on a held-out set, while adjusting the additional two weights, as suggested by Hieber and Riezler (2015). We use pairwise ranking to optimize the additional feature weights, where pairs of competing matches are sampled from the ranking. The idea behind pairwise ranking is detailed in Section 5.3, so we omit further details at this point. The gold standard ranking of the TM candidates is defined by $\mathrm{FMS}(t_{i,j}, \hat{t}_i)$ with respect to the reference $\hat{t}_i$ for q_i. The learning goal is thus to adjust the weights of the n-gram features so as to rank the TM match highest that has the smallest distance to the reference.

Additional Language Model Features To work around the problem of non-decomposable higher order n-gram features spanning several hyperedges we tried adding match candidates as an additional language model when translating q_i. Again, the TM match candidate that produces the highest Viterbi score under this model is selected as the best match. This approach makes use of the fact that cdec handles the extension of the hypergraph to accommodate for the non-local higher order n-grams. Cube pruning (Chiang, 2007) makes the search feasible. Again, we keep the previously tuned SMT weights fixed and interpolate between baseline model score and the additional language model features, optimizing with pairwise ranking.

Both n-gram based approaches produce as a byproduct the Viterbi derivation for each match. The Viterbi derivation for the best match could be directly used as a translation. However, as we optimize $\mathrm{FMS}(t_{i,j}, \hat{t}_i)$ instead of $\mathrm{TER}(\mathit{derivation}(q_i, t_{i,j}), \hat{t}_i)$, where

$$\mathit{derivation}(q_i, t_{i,j}) = \operatorname*{argmax}_{p \in Hg(q_i)} \sum_{e \in p} w_{\mathrm{SMT}} \cdot \phi_{\mathrm{SMT}}(r_e) + w_{match} \cdot \phi_{match}(r_e, t_{i,j}),$$

the translation quality of the derivation is not taken into account. In practice, we find that this frequently results in non-fluent Viterbi derivations. It is possible though to use these derivations directly as an alternative to re-ranking, by optimizing translation quality with regard to a gold standard ranking induced by $\mathrm{TER}(\mathit{derivation}(q_i, t_{i,j}), \hat{t}_i)$ instead. We report additional

results on this variant, where we skip the re-ranking step that is detailed in the following.

5.3 N-best Re-ranking with TM Matches

To incorporate the retrieved fuzzy match into the SMT pipeline we use a simple re-ranking model on the n-best list output by the baseline SMT system and select the best derivation $\hat{h}$ under this model. We balance information from the SMT model and the match by computing a linear interpolation of SMT model score SMT and fuzzy match score FMS between hypothesis h and best TM target match $t_{i,best}$. We also add a weighted version of the FMS score using the retrieval score (CL)IR between TM match and original query q_i as confidence measure.

$$\hat{h} = \underset{h \in H(q_i)}{\operatorname{argmax}}\, w_1 \times \mathrm{SMT}(h) + w_2 \times \mathrm{FMS}(h, t_{i,best}) + w_3 \times ((\mathrm{CL})\mathrm{IR}(q_i, t_{i,best}) \times \mathrm{FMS}(h, t_{i,best})) \tag{5.4}$$

We learn weights for the different components of the score with pairwise ranking optimization (PRO) (Hopkins and May, 2011). The gold standard ranking is induced on the n-best list of SMT outputs, ranked by TER against the reference $\hat{t}$[9]. From the n-best list, pairs of translation alternatives h and h' are sampled. The sampling only includes pairs, that have a difference in score larger than a given threshold value α. We set this parameter as well as other sampling parameters as detailed by Hopkins and May (2011). The goal is for the model score to mirror the evaluation score. This constraint can be transformed as follows.

$$\begin{aligned}
\mathrm{TER}(h, \hat{t}) < \mathrm{TER}(h', \hat{t}) &\Leftrightarrow f_w(\phi(h)) < f_w(\phi(h')) \\
&\Leftrightarrow f_w(\phi(h)) - f_w(\phi(h')) < 0 \\
&\Leftrightarrow w \cdot \phi(h) - w \cdot \phi(h') < 0 \\
&\Leftrightarrow w \cdot (\phi(h) - \phi(h')) < 0
\end{aligned}$$

where $\phi(h)$ is the feature representation of a hypothesis h and f_w a scoring function of the form $f_w(h) = w \cdot \phi(h)$. For each pair of translation alternatives sampled from the n-best list we can subtract their feature values and their evaluation score. Translations that are ranked correctly result in positive examples, incorrectly ranked translations in negative examples. These examples form a training set, on which a supervised binary classifier can be trained, resulting in a weight vector w which we can use directly to compute the n-best list re-ranking model scores. It is easy to expand this model with more features, for example SMT evaluation met-

[9] It is customary in SMT to optimize BLEU score, but since we are working towards CAT applications, we use TER to approximate post-editing effort. (Denkowski and Lavie, 2012)

rics such as BLEU or TER or separate components thereof. We experimented with different feature configurations but found that in general, adding more features did not help. Often the model would over-fit and greatly reduce TER score, but at the cost of a negative impact on BLEU score. We therefore restricted the model to the three components in Equation 5.4.

5.4 Experimental Results

We experiment on a number of corpora of varying sizes, featuring different language pairs and domains. For each corpus, we divide the data into a training set for the SMT system, which doubles as translation memory. We leave three held-out sets: development set for SMT parameter estimation, devtest set for setting all parameters of the translation memory approach and a separate test set, on which the different systems are compared. Note, that in general SMT training data and TM resource are the same data, so the adapted system has no additional information. To compare source and different target retrieval methods in a fair setting, we used the bilingual data from training the SMT model as translation memory, which was restricted to the target side for target retrieval. To evaluate our target retrieval approach in more a realistic setting, we furthermore set up an experiment, where SMT translation model training data and TM resources deviate, because additional monolingual data is available.

5.4.1 Data Sets

Since translation memories are most effective on text that has a certain amount of repetition, we evaluate our approach on typical localization data from technical domains such as information technology (IT), legal and intellectual property (IP)[10]. All corpora are freely available for research purposes. Among the freely available corpora, only the JRC-Acquis corpus has been used previously in combinations of TM and SMT (Koehn and Senellart, 2010a; Li et al., 2014). Most work in this area reports results on TM data from industrial partners that are not publicly available. Usually, these data sets feature a large proportion of fuzzy matches in high ranges, e.g. between 80% and 100%, which makes it possible for the combined systems to achieve a large boost in score. Our reported results are in a smaller range, but achieved on data with a lower rate of high-percentage matches. We manage to gain improvements in performance from matches with an associated fuzzy match score between 10% and 80%, a range, in which many previously reported systems perform lower than the baseline (Koehn and Senellart, 2010a; Zhechev and van Genabith, 2010).

[10]Europarl has been used as a data set by Koehn and Senellart (2010a), but performance of the enriched SMT system actually dropped below the baseline, showing that less repetitive corpora are badly suited for the TM adaptation methods.

Information Technology From the IT domain, we prepared an English-Chinese corpus of manuals from the OPUS[11] corpus (Tiedemann, 2012), the OpenOffice 3 (**oo3**) data. We only kept sentence pairs that contained at least one Chinese character[12], resulting in roughly 50,000 sentence pairs. We tokenized the English side of the data with the tokenizer.perl script from the Moses toolkit and segmented the Chinese side using the Stanford Word Segmenter (Tseng et al., 2005) with the Chinese Penn Treebank standard. Dev, devtest and test sets were created by randomly sampling 1,000 sentence pairs each. The remaining sentence pairs were used for training and as TM.

Legislative For the legal domain, we downloaded and aligned English-French data from the JRC-Acquis corpus[13] (**jrc**), a collection of legislative text from the European Union (Steinberger et al., 2006). We sampled 1,000 sentences each for dev, devtest and test set from documents published in the year 2000 and kept documents published in other years to generate a training set of one million sentence pairs, after filtering out unbalanced sentences and sentences longer than 100 tokens.

Intellectual Property We used data from two patent corpora; English-German descriptions from the PatTR[14] corpus (Wäschle and Riezler, 2012) and Japanese-English data from the 10th NTCIR[15] challenge (Utiyama and Isahara, 2007). We used the official NTCIR-10 dev and test set[16] and produced test and held-out data for PatTR from documents from the year 2006 and collapsed the remaining data from documents published in 2005 and earlier into a training set. From the patent documents, we only took data from description sections, the largest free text section. To test our approach in a setting, where TM and SMT training data deviate, we set up an additional data configuration: we assume that we have parallel data from patent claims and the task is to translate text from a different genre, patent descriptions, for which only data in the target language available. In addition, a small amount of description bitext is available to tune parameters on – a typical *domain adaptation* scenario. The available monolingual data is used to extend both the language model as well as the target-language TM (Table 5.1).

Detailed statistics for experimental data sets are given in Table 5.2. To characterize the data sets, Figure 5.6 reports the distribution of fuzzy match intervals on the various corpora, including data for an actual translation memory from the industry (the numbers have been taken from Wang et al. (2013)). Data from **jrc** and **oo3** feature a large amount of 100% matches,

[11] http://datahub.io/de/dataset/opus

[12] We tested for Chinese characters by checking if they were in the Unicode range `[0x4E00, 0x9FFF]`.

[13] https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis

[14] http://www.cl.uni-heidelberg.de/statnlpgroup/pattr/

[15] http://research.nii.ac.jp/ntcir/ntcir-10/

[16] We merged, shuffled and again split up the data into three sets, to generate a devtest set.

	languages	genre	sentence pairs
SMT train	en-de	claims	6,089,006
dev/devtest/test	en-de	descriptions	each 1,000
LM train	de	claims+descriptions	16,201,061
TM	de	claims+descriptions	16,201,061

Table 5.1: Domain adaptation scenario: deviating SMT training and (monolingual) TM data.

while the rest of the matches is roughly uniformly distributed over the intervals from 20% to 90%. The industry TM has less perfect matches, but most matches rank high between 40% and 100%. The patent data however, has matches mostly in the 20% to 60% range. When we combine this information with the average sentence length for each corpus (Table 5.2)[17], it becomes clear that with longer sentences, high percentage or perfect matches become less frequent. For additional insight, we report repetition rate (Cettolo et al., 2014) for all data sets in Table 5.3[18] to characterize the data sets in term of their repetitiveness. Repetition rate is defined as the geometric mean of the rate of non-singleton n-grams for $n = 1, ..., 4$, computed over a sliding window of 1,000 sentences.

$$\text{RR} = \left(\prod_{n=1}^{4} \frac{\sum_s |\{g_n \in s | \text{freq}_s(g_n) > 1\}|}{\sum_s |\{g_n \in s\}|} \right)^{\frac{1}{4}}$$

where s is a subsample and g_n is an n-gram of length n. Repetition rate is very high on **jrc** and **ntcir** data, but low on **pattr**, even if it is from the same domain as **ntcir**. This is due to the fact that **ntcir** contains claims, which are highly repetitive, while the **pattr** subset that we used only contained descriptions. Given all the data, we can predict that the **jrc** data set might be suited best for our approach, since it contains a lot of matches in higher ranges and features very repetitive language.

5.4.2 Baseline SMT

We trained a hierarchical phrase-based baseline SMT model for the cdec decoder (Dyer et al., 2010) on each data set. A 6-gram language model was trained with SRILM (Stolcke, 2002) on the target side of the training data. The weights of the log-linear model were optimized with MIRA (Watanabe et al., 2007) on a held-out development set reserved for this purpose. We employed the baseline model to produce query translations and hypergraphs for the cross-

[17] This data was not available for the industry TM.

[18] On the two larger data sets, PatTR and ntcir, repetition rate was computed on a continuous subsample of 200,000 sentences.

corpus	data set	#pairs	vocabulary size	
			en	fr
acquis	train	1,093,784	121,237	139,590
	dev	1,000	2253	2760
	devtest	1,000	2455	3035
	test	1,000	2234	2793
			en	zh
oo3	train	49,863	5,957	7,671
	dev	1,000	1,368	1,469
	devtest	1,000	1,320	1,372
	test	1,000	1,330	1,437
			jp	en
ntcir	train	1,673,300	96,207	184,643
	dev	928	2,056	2,980
	devtest	944	2,457	3,380
	test	935	2,363	3,309
			en	de
pattr	train	8,369,562	728,205	678,877
	dev	1,000	5,590	6,166
	devtest	1,000	5,518	6,056
	test	1,000	5,754	6,308

Table 5.2: Statistics for experimental data.

lingual retrieval of target matches as well as to produce 500-best lists, which we re-ranked according to our model given the best match found after fine-grained retrieval. Retrieval and re-ranking parameters were optimized on an additional held-out devtest set. All presented results were obtained on a third test data set, where we measure BLEU (Papineni et al., 2002) and TER (Snover et al., 2006). Scikit-learn (Pedregosa et al., 2011) was employed to learn the parameters of the re-ranking models on the devtest set.

5.4.3 Retrieval Parameters

We used 3-grams to represent sentences in corpora with high average sentence length (legal, patent) and 1-grams for data sets featuring short sentences (IT) to compute MinHash signatures. For re-ranking, we settled on using 500-best lists after experimenting with lists of 100, 1000 and 5000 entries. We found that 500 distinct entries represented a large enough portion

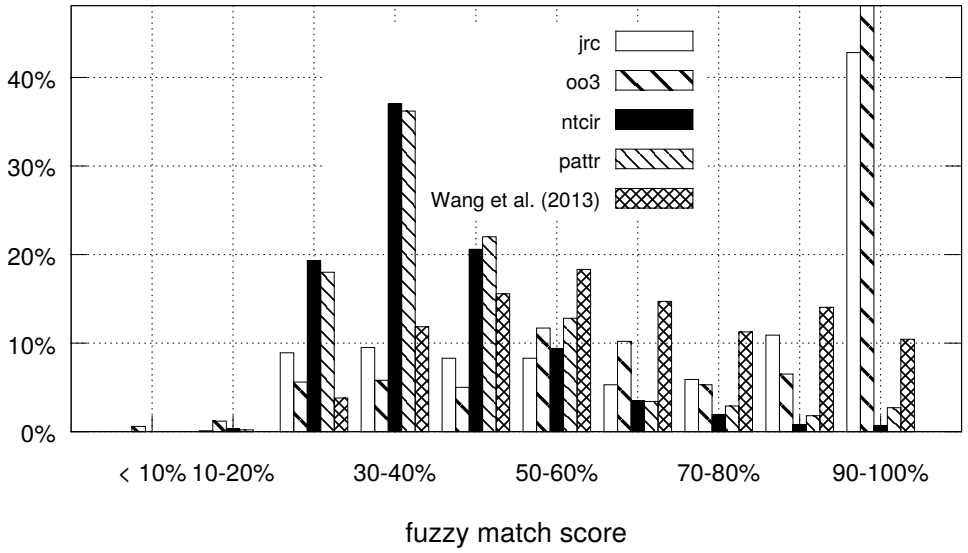

Figure 5.6: Distribution of fuzzy match scores for different corpora.

	acquis		**oo3**		**ntcir**		**pattr**	
	en	fr	en	zh	jp	en	en	de
repetition rate	16.85	16.75	5.98	5.68	16.43	16.90	5.85	3.24
sentence length	24.15	25.19	6.55	7.69	35.32	33.17	35.77	29.73

Table 5.3: Repetition rates and average sentence length in tokens for all target test sets. Numbers on source are comparable.

of the search space for the re-ranking to work at an acceptable time – recall, that edit distance has to be computed between all entries in the *n*-best list and the TM match and has a direct influence on overall time. We chose parameters for the banding approach so that at least one match candidate was returned for $> 90\%$ of the sentences in the devtest set.

5.4.4 Results

Since our experiments are subject to randomness introduced by hashing and parameter tuning, statistical significance was assessed following the method described by Clark et al. (2011) using the source code provided by the authors[19]. Experiments were repeated five times. We report results of the first run and note non-significant[20] differences to the baseline in *italics*.

[19] https://github.com/jhclark/multeval

[20] $p \geq 0.05$

	acquis				oo3			
	BLEU	Δ	TER	Δ	BLEU	Δ	TER	Δ
baseline	61.43		28.16		36.04		50.83	
+src-fms	62.62	+1.49	**26.63**	**-1.57**	**36.88**	**+0.84**	**48.94**	**-1.89**
+tgt-1-best-fms	**62.92**	**+1.48**	26.79	-1.37	36.26	+0.22	50.13	-0.70
+tgt-cl-fms	61.82	+0.38	27.83	-0.33	35.97	-0.07	50.44	*-0.39*
+tgt-unigram	62.23	+0.80	27.56	-0.60	36.16	*+0.12*	50.17	-0.66
+tgt-lm	62.29	+0.85	27.45	-0.71	36.09	*+0.05*	50.15	-0.67
	ntcir				**pattr**			
	BLEU	Δ	TER	Δ	BLEU	Δ	TER	Δ
baseline	24.52		66.52		26.89		57.51	
+src-fms	**25.51**	**+0.99**	62.75	-3.77	27.11	*+0.22*	57.04	-0.47
+tgt-1-best-fms	25.23	+0.71	63.59	-2.93	**27.31**	**+0.42**	**56.78**	**-0.73**
+tgt-cl-fms	**25.51**%	**+0.99**	62.81	-3.71	27.04	+0.14	57.26	-0.25
+tgt-unigram	25.42	+0.91	**62.63**	**-3.89**	27.03	+0.13	57.25	-0.26
+tgt-lm	25.38	+0.86	63.1	-3.42	26.98	+0.09	57.29	-0.21

Table 5.4: BLEU and TER difference to baseline for TM integration on by source-side matching and re-ranking (+src-rr) and variants of target-side matching and re-ranking (+tgt-*-rr). All improvements, except those in *italic* font, are significant with respect to the baseline at $p < 0.01$. Best results in **bold** face.

Source vs. Target Matches

Table 5.4 lists results on the test set with shared SMT and TM training data for all four data sets and different retrieval methods. The results show that adding the TM information always improves over the baseline, up to 1.57 BLEU and -3.89 TER, and our approach is successful across domains and language pairs. Improvements in TER – the metric that was optimized – are always significant at $p < 0.01$. Re-ranking using target-side only matches beats source-side retrieval in terms of TER on two data sets, pattr and ntcir. Except for ntcir, the *n*-gram based models for choosing the best target match always perform worse than the fuzzy-match-score based models. For all data sets except ntcir, monolingual fuzzy matching based on the 1-best translation outperforms the cross-lingual fuzzy match scoring approach and constitutes the best target retrieval strategy. Figure 5.7 shows detailed results on the different fuzzy match intervals, in particular the difference between **+tgt-1-best-fms** re-ranking model and the baseline. It is interesting to note that the highest gains are achieved in the 70-80% range, while previous research reports highest gains in the 95-100% range. This is apparently dependent on the data set, but it also suggests, that the baseline SMT system is already very good in

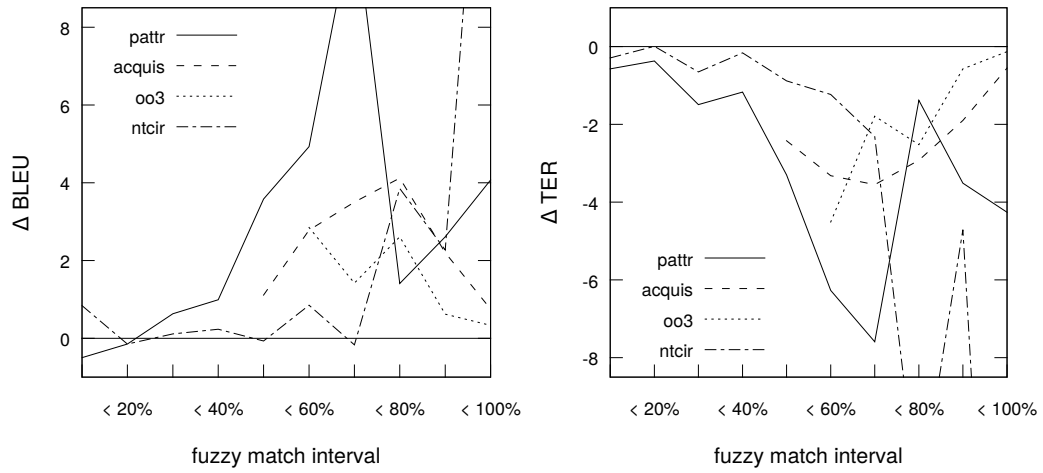

Figure 5.7: Δ BLEU and Δ TER between baseline and system output on different fuzzy match intervals

the high match range, at least for short sentences. For **ntcir** we achieve extremely high numbers in the 90-100% range and for **pattr** in the 60-70% range, but these scores are achieved on very few examples (7 and 14, respectively) and therefore cannot be expected to be stable. The difference between the data sets is probably due to the average sentence length – shorter sentences with a perfect match in the TM are easier to reproduce for the SMT system than longer ones, due to the smaller number of translation options. It is also remarkable, that for **ntcir** and **pattr** data sets even extremely low-range matches are beneficial. While there are some drops in terms of BLEU, TER always goes down, even on 0-10% matches.

Domain Adaptation with Additional Monolingual Training Data

To show the benefit of employing the CLIR method for retrieving target matches from monolingual corpora, we evaluated our approach on the domain adaptation task which we described in Section 5.4.1. We applied the best target retrieval method as determined on the **pattr** data, **+tgt-1-best-fms**. Results for the setup are given in Table 5.5. While using the monolingual in-domain data to train an enlarged language model already yields a huge improvement in performance over an unadapted baseline, using the monolingual data to retrieve TM matches and re-ranking with those on top of the adapted baseline is able to add another significant boost – without adding any bilingual data.

	BLEU	Δ	TER	Δ
baseline	15.72		68.40	
+in-domain lm	21.58	+5.86	62.54	-5.86
+tgt-1-best-fms	**21.81**	**+0.23**	**62.18**	**-0.36**

Table 5.5: Results for domain adaptation scenario.

	acquis		**oo3**		**ntcir**		**pattr**	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
baseline	61.43	28.16	36.04	50.83	24.52	66.52	26.89	57.51
+src-fms (+lsh)	62.62	**26.63**	36.88	**48.94**	25.51	62.75	**27.11**	57.04
+src-fms (-lsh)	**62.78**	26.96	**37.76**	50.07	**25.99**	**61.65**	27.04	**56.87**

Table 5.6: Re-ranking results for exact best fuzzy match, computed on the whole training corpus without coarse candidate filtering with LSH.

Influence of the LSH Approximation

The coarse first retrieval step makes use of two approximations: we try to predict match candidates with small edit distance using Jaccard similarity and we approximate Jaccard similarity by comparing fixed-length, smaller hash signatures instead of full document vectors. To evaluate the influence of these approximations, we performed re-ranking on actual best matches under the FMS measure, by computing FMS against the whole TM corpus instead of a set of candidates. We evaluated this in the source retrieval setting, since it does not depend on the choice of a query representation. The results can be considered an upper bound for our retrieval step, assuming, edit distance is the best measure to determine good fuzzy matches. Results for the comparison can be found in Table 5.6. For most language pairs, using the match with the global best fuzzy match score – without filtering with LSH – has no significant impact on score; where it does, on **oo3** data, the impact is twofold: BLEU goes up by one point, while TER worsens by also going up. TER is known to favor shorter translations – a property, which BLEU score counters with the brevity penalty –, so this result suggests, that using no filtering on a data set with short sentences results in longer matches being chosen as best match. Indeed, exact matches on **oo3** have an average length of 10.63 tokens, while matches retrieved after LSH filtering have an average length of 9.59. The impact on the baseline score is positive in both configurations. In summary, using the LSH approximation does not impact translation results negatively, which confirms that it is not important to find the global best match to improve SMT with fuzzy matches. Using an *n*-gram based overlap score, such as the Jaccard similarity in LSH, instead of the default fuzzy match score has an impact

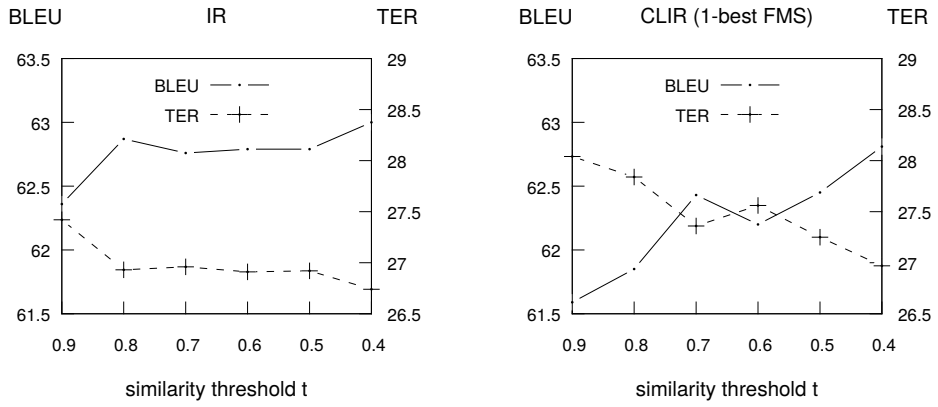

Figure 5.8: Influence of similarity threshold t during LSH retrieval on translation quality for bilingual (IR) and cross-lingual (CLIR) retrieval.

on the length of the retrieved matches, which influences translation outcome under different evaluation metrics.

In a further experiment, we investigated the influence of the similarity threshold value t, which determines, how similar two items have to be in terms of Jaccard similarity in order to be retrieved as match candidates during LSH with a high probability. We compared re-ranking with best matches found according to FMS both in the IR and the CLIR setting on the **acquis** data set. Evaluation results in BLEU and TER are plotted in Figure 5.8. Our findings agree with Ture et al. (2011): in a cross-lingual setting, LSH thresholds have to be set lower in order to gain the same effect as in a monolingual setting. In the IR case, a threshold value of 0.8 is enough to deliver good matches, upon which lowering the threshold cannot improve; in the CLIR setting, lowering the threshold constantly improves results, but at the cost of performing fine-grained matching on a growing candidate set.

Skipping the Re-ranking Step: Biased Decoding on the Whole Search Space

The CLIR scores that operate on the hypergraph, **+tgt-unigram** and **+tgt-lm**, correspond to the score of the Viterbi derivation on a hypergraph enriched with our match features. In the same pass, the Viterbi translation can be built as well. This can be understood as biased decoding with features capturing the closeness of the current derivation to the match. We can therefore use the translations resulting from this biased decoding as final output and skip the re-ranking step altogether, if we alter the objective function slightly to measure translation quality instead of retrieval quality, as discussed in Section 5.2.2. Results for such an evaluation for both hypergraph methods can be found in Table 5.7. Results are mixed; in general, biased

	acquis		**oo3**		**ntcir**		**pattr**	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
baseline	61.43	28.16	36.04	50.83	24.52	66.52	26.89	57.51
+tgt-lm rrank	62.29	27.45	36.09	50.15	**25.38**	**63.10**	26.98	57.29
+tgt-lm decode	62.65	27.44	**36.40**	50.73	24.47	66.67	27.04	57.45
+tgt-1-best-fms	**62.92**	**26.79**	36.26	**50.13**	25.23	63.59	**27.31**	**56.78**

Table 5.7: Re-ranking vs. biased decoding with fuzzy matches with target lm features.

decoding performs similar to re-ranking with best matches determined using the respective features, but in all cases but one is still outperformed by the best CLIR method combined with re-ranking, **+tgt-1-best-fms**. On the **ntcir** data, decoding performs significantly worse than re-ranking, while on **oo3** its is better in terms of BLEU, but worse in TER. Again, the length of the generated translations seems to play a role, which in turn depends on the features used in our model, some of them similar to BLEU (n-gram scores), which penalizes short translations, some similar to TER (edits distance in FMS), which favors them.

5.5 Analysis and Discussion

We have shown that biasing SMT towards matches found in a translation memory using both the same and deviating resources as the SMT training data yields significant improvements over the non-biased baseline. Tables 5.9 and 5.10 compare translation output between baseline and the **+tgt-1-best-fms** model. The examples show, that the system is able to correct syntactical errors and reordering problems, but also, that sometimes a change only means swapping translations for a phrase or a term, when both options would be correct choices. In this case, the translation both gains and loses from this phenomenon with regard to the reference. We assume that this holds for the whole test set: in some cases the system will randomly pick the right (meaning: used by the reference) translation; sometimes adding a match will change a correct translation. Since overall our system improves significantly over the baseline, this means meaningful changes are made frequently.

We conclude the following points from the experiments presented above.

- Matches retrieved by CLIR target translation retrieval models can be as helpful as matches retrieved on the source side. The target retrieval works better, if longer sentences are involved and the vocabulary is repetitive, even if the matches are in the lower fuzzy match ranges (as seen on **acquis** and **ntcir** data). We hypothesize that the target retrieval requires a certain amount of anchoring words in order for the retrieval to work.

source	in one particular embodiment , the aliphatic hydroxy carboxylic acids bear the hydroxyl group and the carboxyl group on the same carbon atom .
baseline	*in einer besonderen ausführungsform die aliphatischen hydroxycarbonsäuren , die die hydroxylgruppe und die carboxylgruppe an ein und demselben kohlenstoffatom tragen .*
tm match	*in einer besonderen ausführungsform des erfindungsgemäßen verfahrens tragen die aliphatischen hydroxycarbonsäuren die hydroxy - und carbonsäuregruppe am selben c - atom .*
+rrank	*in einer besonderen ausführungsform **tragen** die aliphatischen hydroxycarbonsäuren die hydroxygruppe und die **carbonsäuregruppe am gleichen c - atom** .*
reference	in einer besonderen ausführungsform tragen die aliphatischen hydroxycarbonsäuren die hydroxy - und carboxyl gruppe am gleichen c - atom .

Figure 5.9: Example system output on pattr test set: With the TM match (retrieved with a fuzzy match score of 68.18%), the syntax of the output has been corrected: the subordinate clause has been removed and the verb *tragen* placed correctly in the main clause. Furthermore, an article doubling has been removed (*die die*). Some vocabulary changes occurred, not all of them beneficial: *kohlenstoffatom* has been turned into *am gleichen c - atom*, which is both correct, but the latter is the term used in the reference; on the other hand, *carboxylgruppe* was correctly output by the baseline, but changed to *carbonsäuregruppe* – correct, but not the word used in the reference.

source	the present invention is based on the object of improving a power trowel in such a way that corner areas can be better smoothed .
baseline	*der erfindung liegt die aufgabe zugrunde , eine leistung traufel in einer solchen weise , daß eckbereichen besser geglättet werden kann .*
tm match	*der erfindung liegt die aufgabe zugrunde , das gattungsgemäße verfahren so zu verbessern , daß die halbleiterbauelemente billiger hergestellt werden können .*
+rrank	*der erfindung liegt die aufgabe zugrunde , einen spachtel leistung derart zu verbessern , daß eckbereichen besser geglättet werden können .*
reference	der erfindung liegt die aufgabe zugrunde , einen flügelglätter dahingehend zu verbessern , dass eckenbereiche besser geglättet werden können .

Figure 5.10: Again, re-ranking with regard to a TM match (with a match score of 59.1%) improved the translation by adding a missing verb to the subordinate clause, *verbessern*, and correcting the form auf the auxiliary verb *können*.

- LSH is an efficient way to retrieve candidates for good fuzzy matches. Using an approximation as a filtering step instead of exact retrieval does not come at the cost of translation quality.
- On long sentences in a less repetitive domain, gains are rather small (**pattr**). Similarly, on a domain with short sentences and a high number of perfect matches (**oo3**), re-ranking gains are barely significant, presumably because the SMT baseline is already very strong.
- In a domain adaptation scenario, additional monolingual in-domain data used as a TM can give small, but significant improvements over a very competitive language model baseline, proving the relevance of the CLIR methods.
- Advanced methods for choosing the best target TM match do not perform better than basic fuzzy-match-score-based ranking against the 1-best translation. This suggests that a strong SMT model bias – relying on the best translation as suggested by the SMT decoder – is beneficial; at least, when the SMT baseline system is good. Defining more informed match features on the hypergraph might be able to change this outcome.

In summary, we can conclude that cross-lingual retrieval is a promising way to exploit monolingual resources for translation. While our light-weight re-ranking approach functions similar to a language model and we can assume that the improvements we saw will become smaller, when using higher-order language models, the main contribution lies in the extension of the retrieval. Even though we used it for the main evaluation, the targeted application of our approach is not SMT, but computer-assisted translation: Human translators are hesitant to integrate SMT into their workflow, since the generated translations lack fluency and might contain errors. The cross-lingual retrieval can be used to mask the SMT by providing (fluent) fuzzy matches in the target language that are similar to the SMT hypothesis. These matches are not accessible with standard, source-source TM retrieval. Our method has the potential to massively enlarge the search space for fuzzy match search in order to provide human translators with more and better matches. In this light, one main goal of future work is an evaluation in a post-editing setting aimed at finding a reduction in post-editing effort. We expect to see gains in correspondence with those seen in our SMT experiments, when adding fuzzy matches retrieved with cross-lingual search.

CHAPTER 6

Online SMT Adaptation with User Feedback

State-of-the-art SMT systems translate text sentence by sentence. Each sentence is processed in isolation and context beyond the sentence level is not considered when choosing translation options. This simplification makes it difficult for a system to resolve ambiguities and choose the correct translation in the global context. Furthermore, it gives rise to the problem of translation consistency, since the system has no memory of previously produced translations on the same document. This is especially impairing when considering a real-world translation scenario, such as computer-assisted translation (CAT), where a user and an MT system interact to create translations. The MT component, a translation memory or a combination of both suggest translations, which are then revised and refined by a human translator. In this context, a static MT system will reproduce mistakes that have been previously corrected by the user. This leads to an unnecessarily inefficient translation process and a possibly frustrating experience for the user, since the system is unable to learn from its mistakes – for an example see Figure 6.1. Since translation in a CAT scenario is an interactive and time-sensitive task, a complete retraining of the translation models, as performed for batch adaptation, is not feasible. We therefore propose an online learning approach that extracts information from user feedback and immediately integrates the information into the MT engine to reduce time and effort spent on correcting MT hypotheses. Context-specific translations are learned from already edited segments of a text by aligning source sentences and user-approved translations after they have become available. We augment the system with new phrases learned from these alignments and bias it towards already seen translations that have been selected by the user, encouraging translation consistency with regard to the human translation. To

SENT. 7	source	*Annex to the **Technical Offer***
	MT output	*Annex all'**Tecnica Offri***
	user translation	*Allegato all'**Offerta Tecnica***
SENT. 8	source	*Sistemi Informativi **SpA***
	MT output	*Sistemi Informativi **SpA***
	user translation	*Sistemi Informativi **S.p.A.***
SENT. 21	source	*This document is Sistemi Informativi **SpA**'s **Technical Offer***
	MT output	*Questo documento è Sistemi Informativi **SpA** di **Tecnica Offri***
	user translation	*Il presente documento rappresenta l'**Offerta Tecnica** di Sistemi Informativi **S.p.A.***

Figure 6.1: Information extracted from user feedback on the translations of an MT system should be used to update the system as soon as it becomes available. The phrase *Technical Offer* first appears in sentence 7; when it appears again in sentence 21, the MT system should be able to reproduce the correct translation supplied by the user earlier, *Offerta Tecnica*. In the same way, the spelling variant *S.p.A.* could be learned.

achieve this, we construct local language and translation models based on the growing cache of available human translations and combine them with the static global model. We evaluate our approach on two domains and language pairs. Since an authentic CAT scenario is difficult and expensive to create, we measure translation quality on the reference translations of the test set. Although the evaluation is not interactive, part of the test sets were created by CAT users in a field test. In this way, we gain multiple references for each sentence and a test set, which is similar to the target scenario. Our adapted models achieve improvements of up to 3 BLEU point over the static baseline.

The work described in this chapter was mainly carried out during an internship with the Human Language Technology unit at Fondazione Bruno Kessler, Trento, Italy[1] for the MateCat project[2], under the supervision of Marcello Federico and Nicola Bertoldi. Results have been published in Wäschle et al. (2013) and Bertoldi et al. (2014) with a comparison to a discriminative adaptation approach developed by Patrick Simianer and additional experimental work conducted by Nicola Bertoldi and Mauro Cettolo. The chapter is organized as follows: Section 6.1 discusses related work on online adaptation and interactive SMT. Section 6.2 describes the adaptation of language and translation model in detail. Experiments and results are presented in Section 6.3.

[1] http://hlt.fbk.eu/

[2] Machine Translation Enhanced Computer Assisted Translation, http://www.matecat.com/

6.1 Related Work

The problem of updating a system incrementally emerges in many areas of research on machine translation. Online learning methods have been applied to the task of tuning system parameters, e.g. in the context of stochastic methods for discriminative training (Liang et al., 2006a; Chiang et al., 2008) or streaming (Levenberg et al., 2010, 2011). In these scenarios, the model is updated after translating each example of the training set. After training is completed, however, the model remains static when translating test sets. Liang and Klein (2009) employ online versions of the EM algorithm to the problem of generating word alignments. Incremental adaptations of the whole system have been presented for larger batches of data (Bertoldi et al., 2012; Cettolo et al., 2013b). In terms of granularity, our scenario is most similar to the work by Hardt and Elming (2010), where the Moses training procedure is employed to update the phrase table immediately after a reference becomes available. Our work, however, focuses on adapting both language and translation model with techniques that leave the global models unchanged. This is important in a CAT environment, where several users might use the same global model but individual local models. The CAT scenario is also related to interactive machine translation (IMT) (Nepveu et al., 2004; Ortiz-Martínez et al., 2010; López-Salcedo et al., 2012), where the human translator receives suggestions for translation options on the word or phrase level while inputting his or her translation, and to cache-based SMT (Tiedemann, 2010). In contrast to this work, our approach operates strictly on the level of full sentence translations that are revealed after a prediction for an input is made. In the field of domain adaptation, combining a local model learned specifically for the test domain and a global model learned on general data, is common approach. Possible combination approaches are for example a log-linear framework (Koehn and Schroeder, 2007), mixture models (linear or log-linear) (Foster and Kuhn, 2007) or fill-up methods (Bisazza et al., 2011).

Other work specifically on sentence-level online adaptation for SMT in a CAT scenario has been presented. Martínez-Gómez et al. (2012) use different well-known online algorithms, such as perceptron, online passive-aggressive, and discriminative ridge regression, to adapt model parameters; in particular the weights of the log-linear model and the phrase table weights. Their adaptation of the translation model parameters is similar to our approach. However, we also adapt language model parameters and are able to augment the translation model with new phrases in addition to re-weighting phrases already present in the model. Denkowski et al. (2014) adapt three components of a hierarchical phrase-based SMT system to user post-edits: translation grammar, language model and model parameters using the suffix array technique, hierarchical Pitman-Yor process language models (HPYPLM) and MIRA. Simard and Foster (2013) adopt a different approach: instead of adapting the model components, they incrementally learn an automatic post-editing (APE) system that translates MT output into user-corrected output and employ it on top of the SMT pipeline, applying automatic post-edits before the output is passed to the user.

A number of related papers have been published on adaptation experiments for the MateCat tool. Blain et al. (2012) align source and user translation using the MT hypothesis as a pivot followed by full re-training of the phrase table, which is the interpolated using the Moses backoff mode. They compare the results of their alignment procedure to incremental GIZA++ alignments. Bertoldi et al. (2013) was published as a companion paper to Wäschle et al. (2013), experimenting with a repetitiveness measure as a criterion to predict the impact of the adaptation method and presenting additional results using a context-reward strategy. Mathur et al. (2013) perform sentence-wise adaptation by introducing an additional feature which represents a goodness score of each phrase pair in the test set as well as learning weights for the log-linear combination with the MIRA algorithm.

A related area is the analysis of translation consistency (Carpuat, 2009; Ma et al., 2011; Ture et al., 2012). Carpuat and Simard (2012) show that increased consistency of an SMT system does not correlate with better translation quality – however, translation errors are indicated by inconsistencies. Our work can be viewed as an approach to improve translation quality by enforcing local consistency with regard to human translations via online learning.

6.2 Online Adaptation for CAT

Online learning strategies are a good fit for tasks, which can be learned incrementally, or where the training data is too large to be processed in one batch. A model improves over time with incremental updates applied after receiving an example from the training set. This view on learning fits well in a CAT workflow, where an SMT system produces translation suggestion and a human translator post-edits or rewrites the hypotheses. New training examples become steadily available, while working on a document. A hypothetical CAT workflow involving a translator and an MT system provides the opportunity to integrate two online learning steps (in *italic* font).

For each sentence in the document:

1. MT system produces a translation (hypothesis)
2. user corrects MT output
3. *collect feedback from user correction*
4. *update MT engine with collected feedback*

In particular, our online adaptation approach for the SMT component of a CAT system is fitted to the MateCat workbench. Figure 6.3 shows the architecture of the tool: a translation server

processes requests by several users at once and handles the translation resources, TM[3] and the SMT system, distributing their output to the users. We can assume that the translators work on the same project and these resources are adapted to their common domain at hand, but each translator processes a different document. Our model steps in at the level of document adaptation. The main idea is to build a cache of document-specific translations for each user separately. After processing a document, the newly available parallel data might be added to the existing models, which we refer to as the global model, with a batch update (Bertoldi et al., 2012; Cettolo et al., 2013b). During translation, a local cached model is built specifically for each user and each document – when the user starts a new document, the cache that was collected on the previous document is emptied. Figure 6.2 formalizes this online learning protocol with a local cache.

train global model m_g
for all documents d of $|d|$ sentences **do**
  reset local model $m_d = \emptyset$
  for all sentences $s = 1, \ldots, |d|$ **do**
    combine m_g and m_d into m_{g+d}
    receive input sentence x_t
    output translation $\hat{y}_t$ from m_{g+d}
    receive user translation y_t
    refine M_d on pair (x_t, y_t)
  end for
end for

Figure 6.2: Online learning protocol.

The learning process starts from training a global model M_g on parallel data in the range of millions of sentence pairs. Then for each document d, consisting of a few hundred up to a thousand sentences, a local model M_d is created. For each example, first the static global model M_g and the current local model M_d are combined into a model M_{g+d}. Then the input x_t is translated into $\hat{y}_t$ using the model M_{g+d}. Finally the local model M_d is refined on feedback y_t that is received immediately after producing $\hat{y}_t$. In the evaluation of our experiments, the refinement step is simulated by updating on the reference, but the local prediction $\hat{y}_t$ for the sentence in question is not changed. This information is only available to the model for translating the following sentences, so the evaluation by computing scores over the whole set after online learning on the set is still fair.

We implement the adaptation of a local caching model for the phrase-based SMT system Moses. The main components of the SMT model are language (LM) and translation model (TM). Both components are adapted using similar techniques. While the adaptation of the language model only requires the information presented by the final translation, we need an

[3]Note, that in the remainder of this chapter, the abbreviation TM stands for *translation model*.

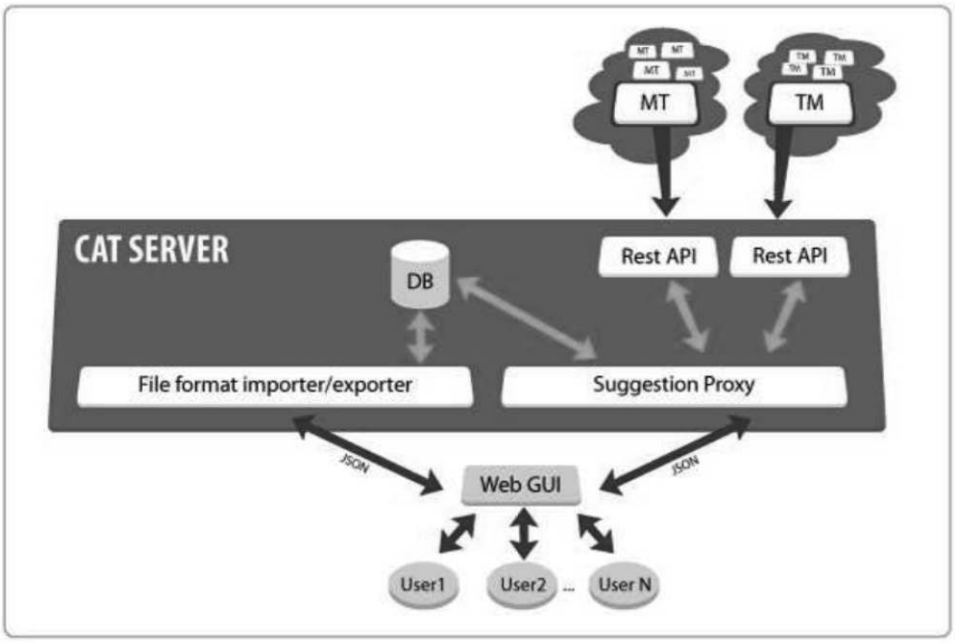

Figure 6.3: Architecture of the MateCat tool (Fondazione Bruno Kessler, 2012).

alignment to extract bilingual information to update the translation model. A strategy to obtain such an alignment is detailed in the next section.

6.2.1 Constrained Search Alignment for Feedback Exploitation

In order to extract information to refine translation models on user feedback, source and user translation need to be aligned. For our purposes, we choose a direct alignment at the phrase-level – as opposed to aligning at the word level and extracting phrases as done in the full SMT training pipeline. To this end, we employ a constrained search technique described in Cettolo et al. (2010). Given the set of all possible translation options[4] for all phrase segmentations of the source sentence, the constrained search seeks to optimize the coverage of both source and target sentences while minimizing distortion of the source. The search is implemented with dynamic programming and the alignment is built incrementally from left to right by growing the covered target span: at each target position i' that features a distinct set of alignment links to source spans, the contiguous alignment with the highest score considering the covered source and target span, $s_{[j\ldots j']}$ and $t_{[i\ldots i']}$, respectively, is recorded.

[4]The translation options can be output during decoding. Since every source sentence for which user feedback is available has already been run through the decoder, no additional decoding step is needed.

$$score(s_{[j...j']}, t_{[i...i']}) = (j' - j) + (i' - i) + distortion(s_{[j...j']})$$

From all alignments that fully cover the target text, the one that maximizes this score is selected, resulting in exactly one optimal phrase segmentation and alignment. In contrast to forced decoding, where translations that are not fully reachable with the given models cannot be produced, the constrained search allows gaps such that some source and target words or phrases may be uncovered by the alignment. This is achieved by adding dummy phrase pairs, which associate target words with an empty source span, for every target word. Note, that the phrase translation probabilities are not taken into account – the alignment only optimizes coverage while simultaneously trying to keep distortion small. Since the user post-edit is highly reliable, these gaps can be aligned to extract new phrase pairs that are not present in the global model. Since we have no information about the content of the gaps[5], this strategy can only be applied for unambiguous gaps, where exactly one OOV phrase is missing on the source and one on the target side. We assume that in this case we have a match and align them. From the resulting phrase alignment we collect three different types of phrase pairs.

New phrase pairs by aligning unambiguous gaps.

Known phrase pairs that are already present in the given model but should receive a high translation probability under the local model.

Full pairs consist of the complete source sentence and its user translation and are designed to emulate a translation memory.

Due to the highly ambiguous nature of non-content words and their behavior as garbage collectors during word alignment, only phrases that contain at least one content word are considered and entered into the cache. For the source and the user translation given in sentence 7 in Figure 6.1, the constrained search yields the alignment shown in Figure 6.4. We can extract the new phrase pair *Technical Offer* $\rightarrow$ *Offerta Tecnica*, the known phrase pairs *Annex* $\rightarrow$ *Allegato* and *to the* $\rightarrow$ *all'* and the full phrase *Annex to the Technical Offer* $\rightarrow$ *Allegato all' Offerta Tecnica* from the alignment.

6.2.2 Translation Model Adaptation: A Local Phrase Table

The growing collection of phrase pairs extracted from source sentences and corresponding user-approved translations enables the construction of a *local* phrase-based translation model. The goal of this local model is to reward MT translations that are consistent with previous user translations as well as to integrate new translations learned from user corrections,

[5] A more refined feedback model, which was not available in our case, could collect such information from the user.

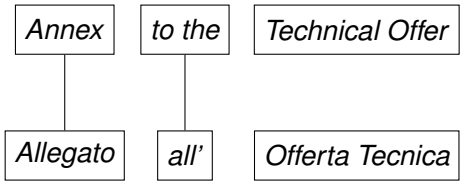

Figure 6.4: Phrase segmentation and alignment.

Annex to the `<p translation = "Offerta Tecnica || Proposta Tecnica" prob = "0.75 || 0.25">` *Technical Offer* `</p>`

Figure 6.5: Moses XML markup.

in order to better translate the remaining sentences. The local model can be seen as a bias that is backed up by the global translation model, since especially at the beginning of the document the local phrase table tends to be sparse. From each sentence pair, all phrase pairs extracted with the constrained search technique described in Section 6.2.1 are inserted into a cache. Phrase translation probabilities are estimated based on the relative frequency of the target phrase given the source phrase within the cache. Cache and model are updated on a per-sentence basis as soon as a user translation for a source sentence becomes available.

Decoder Integration Moses offers a convenient and fast way to integrate the constantly changing local model in the decoder at run-time without changing the global model in memory. With the Moses XML input option (Koehn and Haddow, 2009), phrases in the source sentence can be annotated with translation options and translation probabilities via XML-like markup, when they are sent to the decoder. The markup encodes a (miniature) translation model that is passed to the decoder. This model is used it for decoding the current sentence but immediately discarded afterwards. Multiple phrase translations and corresponding probabilities for a source phrase can be suggested (see Figure 6.5). Moses offers two options to interact with this local phrase table. In *inclusive* mode, the additional phrase translations compete with existing phrase table entries. The decoder is forced to choose only from the given translations in *exclusive* mode. During development, we found that the exclusive option is too strict given the phrases extracted with constrained search. Though most phrase pairs are correct and useful additions, for example spelling variants such as *S.p.A*→*SpA* or domain vocabulary such as *lease payment*→*canone*, some are restricted to a specific context, e.g. translation from singular to plural such as *service*→*servizi*, and some are actually incorrect. In the inclusive mode, the global translation and language model can reject unlikely translations and offers soft interpolation. From a technical point of view, the probability mass passed through the mark-up is split over the four phrase-table scores used in the log-linear model, $p(e|f)$, $p(f|e)$ and their lexical counterparts.

<dlt cblm= "Offerta Tecnica || Offerta || Tecnica"/>*Annex to the Technical Offer*

Figure 6.6: Cache-based language model markup.

We implemented the online learning as follows: The local cache is collected and controlled by a separate wrapper script around the decoder call. Before sending a sentence to the decoder, the source is checked for phrases present in the cache and, if phrase translations are found, annotated with the information. Following the construction of a translation by the decoder, the annotation operates in a greedy way from start to end of a sentence. First, the complete sentence is looked up in the cache to find for possible perfect matches. Then, starting from 5-grams[6] down to unigrams, the cache is checked for matches. In this way, translations for larger spans, which we assume to be more reliable, are preferred over word translations.

6.2.3 Language Model Adaptation: N-gram Cache Feature

Complementary to the local translation model, we employ a target language model based on a cache to reward target *n*-grams seen in previous user translations. Since the language model has considerable impact on the final output choice, the global LM might often discard new phrase translations produced by the local translation model, even if their associated phrase translation probability is high. We therefore add a local language model to support the translation options suggested by the local translation model in order to increase the effect of the adaptation and improve performance. We use an implementation of a cache-based language model, which adds an additional feature to the log-linear model (Bertoldi, 2014). The feature computes an additional score for each translation option based on a target *n*-gram cache. All *n*-grams g_n extracted from the user post-edit are associated with an age and a score, which is strictly positive and decays exponentially as the age increases.

$$score_{ng}(g_n) = exp(\frac{1}{age(g_n)})$$

New or reappearing *n*-grams are added to the cache with an age of 1. At each iteration of the online learning, i.e. after the processing of a single sentence, the age of the existing entries in the cache is increased by 1. The score of a translation option, i.e. a phrase $e_1, ..., e_m$ of length m, is computed by summing up over the cache scores of all *n*-grams contained in the option; *n*-grams that are not present in the cache receive a score of 0.

[6] We limited phrase size to 5 tokens.

$$score_{ph}(e_1, ..., e_m) = \sum_{i=1}^{m} \sum_{j=i}^{\max(n,m-i)} score(e_i, ..., e_j)$$

Using this technique, *n*-grams crossing over non-contiguous translation options are not taken into account. The feature is therefore strictly speaking not a language model, but can be seen as booster feature for specific *n*-grams. Since the model is implemented as a feature in Moses, its influence is controlled by an additional weight in the log-linear model. To optimize this weight, first all other weights are tuned with MERT and then taken as fixed, while the additional weight is optimized using grid search. The cache is filled dynamically using XML-like input similar to the XML input option for phrase translations (see Figure 6.6). For our purpose we update the cache with all user translation target *n*-grams that contain at least one content word after the translation becomes available, i.e. on the translation of the next sentence. We also experiment with the option of adding only the target side of phrase pairs included in the translation model cache to the language model cache.

6.3 Experimental Results

We use the open-source toolkit Moses (Koehn et al., 2007) to build the static global baseline model. Word translation probabilities were estimated with GIZA++ (Och and Ney, 2003). We learned a 5-gram language model using improved Kneser-Ney smoothing with the IRSTLM toolkit (Federico et al., 2008) on the target side of the training data. The weights of the log-linear interpolation were optimized with MERT (Och, 2003).

6.3.1 Data

As discussed in Chapter 5, CAT is especially relevant for technical domains. We develop and test our system on English-Italian data from the information technology (IT) domain. In addition, we show that the results transfer well to other domains and language pairs by evaluating a system trained on German-English patent text. Both domains are fairly technical in their topics and especially suited for the presented adaptation approach, since we expect many domain-specific terms to appear repeatedly throughout the document.

IT Data The training data for the global IT models is compiled from a translation memory and several OPUS[7] corpora (Tiedemann, 2012), totaling 1.2 million sentences and 19 million English running words. For development, six documents that each correspond to an IT project

[7] http://opus.lingfil.uu.se

are used, where the reference simulates a user-approved translation. We split the documents into two groups, dev1 and dev2. Weight optimization was performed on dev1 and the best overall system configuration was determined according to the scores computed on dev2. For testing, FBK provided a document from the IT domain, for which actual user corrections from three different translators (A-C) were available for each sentence, which were collected during a MateCat field test (Cettolo et al., 2013a). We report separate scores for all three translators, regarding each translator as a document. This choice has strong motivations in the online adaptation scenario. Each translator processed the sentences in his or her preferred order and provided different feedback. Consequently, the original baseline system evolves differently, and possibly achieves different performance. We aim to show that the proposed techniques give consistent improvement on different sets of user feedback, regardless of the overall performance of the baseline system.

Patent Data As evidence that the results transfer to different languages and domains, we train a system on German-English patent text sampled from title, abstract and description sections from the PatTRcorpus (Wäschle and Riezler, 2012). To depict a realistic scenario, data in the training corpus is sampled from documents from all top-level IPCsections, which can be viewed as technical subdomains, while domain-specific development and test documents are sampled from the same section, `E` (*Fixed Constructions*). We tune the weight for the cache-based language model feature on a domain-specific set dev1, consisting of three patents containing title, abstract and description sections, but take the best overall system configuration determined on the IT dev2 set. We report evaluation results on four more patent documents. Statistics for all data are reported in Table 6.1.

Extracting Whole Documents from PatTR

Since a huge number of small files is hard to handle for operating systems, PatTR has been stored in large files sorted by language pairs and patent text section, from which the parallel sentences were extracted. To reconstruct the original patent document, we can use two properties of the data. (1) Each sentence pair is annotated with a patent document identifier indicating the document it was extracted from and the IPC classes the document was assigned to. (2) Sentences appear in these files in the order in which they appear in the original documents – this is especially important for our task. We pick a document identifier in the targeted IPC class and extract all sentences that match this identifier from the title, abstract and description sub-corpora, in the order of their appearance. The sentences are concatenated in the following order: title first, followed by abstract and description.

	IT		patent	
	doc	sentences	doc	sentences
	train	1,167 K	train	4,199 K
dev1	prj1	420	pat1	318
	prj2	931	pat2	304
	prj3	375	pat3	222
dev2	prj4	289	pat6	300
	prj5	1,183	pat5	227
	prj6	864	pat6	239
test	prj7A	176	pat7	232
	prj7B	176	pat8	230
	prj7C	176	pat9	225
			pat10	231

Table 6.1: Statistics for training, dev and test data.

6.3.2 Evaluation

Since field tests with actual translators require considerable resources, we evaluate our approach automatically usingto BLEU (Papineni et al., 2002) on the given reference translations. While we adapt our system online on the test set, evaluation is carried out in batch using the whole translated test set and references, because the BLEU metric is not designed to be used on individual sentences. Usually, online adaptation approaches are evaluated in a second pass over the training set. However, we do not re-translate the test set and evaluate the first pass – this evaluation is still fair though, because only references of previously translated sentences are used in our adaptation of the system. Since we have collection of development and test documents, we report mean BLEU scores on all domains. The best results are highlighted in **bold** and the standard deviation is given in small font size.

6.3.3 Translation Model Adaptation

Table 6.2 shows mean BLEU scores and differences from the baseline on IT dev1 and IT dev2 sets for TM adaptation as described in Section 6.2.2. We see that each condition of TM adaptation, +**new**, +**known** and +**full**, yields individual improvements. Furthermore, improvements of individual conditions add up to an overall improvement of 2.90 (IT dev1) and 2.42 (IT dev2) BLEU points over the baseline for the combination of all conditions, namely +**new+known+full**. Standard deviation of the difference is quite high, but always the same or lower than the improvement in score. The **full** translations yield the highest individual im-

	IT dev1		IT dev2	
	BLEU	Δ [σ]	BLEU	Δ [σ]
baseline	22.59		21.49	
+**new**	23.11	+0.52 [0.57]	21.64	+0.15 [0.06]
+**known**	23.73	+1.14 [0.70]	22.24	+0.75 [0.15]
+**full**	24.22	+1.63 [1.73]	23.07	+1.58 [0.91]
+**new+known**	24.33	+1.75 [0.80]	22.42	+0.93 [0.19]
+**new+known+full**	**25.49**	+**2.90** [2.18]	**23.91**	+**2.42** [0.83]

Table 6.2: Translation model adaptation using **new**, **known** or **full** phrases (see Section 6.2.2) and combinations of the three on IT dev1 and IT dev2 on top of the baseline. Figures reported are mean BLEU scores and mean difference from the baseline (in small font size). Standard deviation of the difference is indicated in square brackets. Best results are highlighted in **bold**.

	tm	+1-**gram lm**	+4-**gram lm**	+**phrase lm**
IT dev1	25.49	27.32	**27.64**	26.36
		+1.83 [0.45]	**+2.15** [0.76]	+0.87 [0.74]
IT dev2	23.91	24.62	**25.25**	24.87
		+0.70 [1.17]	**+1.34** [1.26]	+0.95 [1.01]

Table 6.3: Different language model (lm) adaptation modes (see Section 6.2.3) on top of the best translation model (tm) adaptation. Figures reported are mean BLEU scores and mean differences and standard deviation from baseline (in small font size).

provement, indicating that the data is very repetitive, and the **new** translations the lowest. Closer examination revealed, that only a small number of new translations are added for each document.

6.3.4 Language Model Adaptation

Table 6.3 compares the effect of adding different language model adaptation conditions on top of the best translation model adaptation result, which corresponds to using all phrase types (see Table 6.2). All language model conditions improve over the translation model baseline. Using *n*-grams up to order 4 (+4-**gram lm**) yields the largest gains – 2.15 (IT dev1) and 1.34 (IT dev2) BLEU points – compared to an LM booster only for unigrams (+1-**gram lm**) or only the target side of the phrases used in the TM adaptation (+**phrase lm**). This highlights the importance of the language model bias feature for promoting the local translation model and the impact of document-specific adaptation.

6.3.5 Results on Test Sets

Results on the IT test set with three different post-edits are shown in Table 6.4. We find large improvements in BLEU point for the combination of TM and LM adaptation over a static baseline model for two translators, A and C. This corresponds to the findings on the IT dev sets. However, for translator B only a non-significant improvement is measured. A reason could be that the order in which the sentences were processed was not beneficial for the adaptation, or that the post-edits made by the translator were not consistent.Table 6.5 lists results on the four patent test sets[8]. The combined system achieves significant improvements over the baseline on all data sets.

	baseline	tm+4-gram lm
prj7A	41.10	**42.97** **+1.87**
prj7B	39.68	39.72 +0.04
prj7C	30.68	**33.76** **+3.08**

Table 6.4: Results for TM and LM adaptation on IT test set featuring user corrections by three translators each (A-C). Figures reported are BLEU scores with differences from baseline (in small font size).

	baseline	tm+4-gram lm
pat7	26.22	**27.28** **+1.06**
pat8	33.25	33.50 +0.25
pat9	36.57	**38.52** **+1.95**
pat10	31.72	**35.35** **+3.63**

Table 6.5: Results for TM and LM adaptation on patent test sets. Figures reported are BLEU scores with differences from baseline (in small font size).

6.4 Discussion

While our method can achieve large gains, on some of the data sets gains are barely noticeably and not significant. This indicates that the success of the method depends crucially on the data provided. The more repetitive and the more consistent the translation, the larger the improvement that can be gained by document-specific adaptation (Cettolo et al., 2014). With these prerequisites, our method is a good fit for predominantly technical domains. Even

[8]The results on patent data for the author's implementation of the system described above were kindly provided by Nicola Bertoldi, who coordinated the SMT experiments for the joint work in Bertoldi et al. (2014).

source	*A copy type is automatically assigned to a consistency group when the first relationship is added to the consistency group.*
baseline	*Una copia tipo viene automaticamente assegnato a un gruppo di coerenza quando la prima relazione viene aggiunto al gruppo di coerenza.*
adapt **tm**	*Una copia tipo viene automaticamente assegnato a un gruppo di **congruenza** quando la prima relazione viene aggiunto al gruppo di **congruenza**.*
adapt **tm+lm**	*Un **tipo di copia** viene automaticamente assegnato a un gruppo di **congruenza** quando la prima relazione viene aggiunto al gruppo di **congruenza**.*
reference	*Un tipo di copia viene assegnato automaticamente a un gruppo di congruenza quando la prima relazione viene aggiunta al gruppo di congruenza.*

Figure 6.7: Comparison of baseline and adapted system output on IT test. With an adapted translation model, the system is able to correct the translation of *consistency*, a phrase that frequently appears in the document. By adding language model adaptation, the system also produces a correct Italian compound word translation of *copy type*.

though the proposed method is simple, gains are substantial, which leads us to predict that a more refined feedback process could produce much larger gains. Our system has no information beyond the final post-edit, so the alignment of new phrase translation is limited to unambiguous cases. In an advanced system integrated in an editing workbench, we could ask the user to provide alignment for translation parts that the global background model cannot produce. Though our implementation is tailored to the Moses decoder, comparable document model caching strategies can be applied to other decoders as well. Denkowski et al. (2014), for example, proposed models for online adaptation for the cdec decoder.

6.4.1 Example Output

Figure 6.7 details example output for the different adaptation methods and compares it to the baseline. Both language and translation model adaptation improve over the baseline in comparison to the user feedback reference by biasing the translation towards using the right terms in the context. The combined system is able to produce an output that is almost – except for one instance of different (but still correct) word order – a perfect match to the reference using the information of both local models.

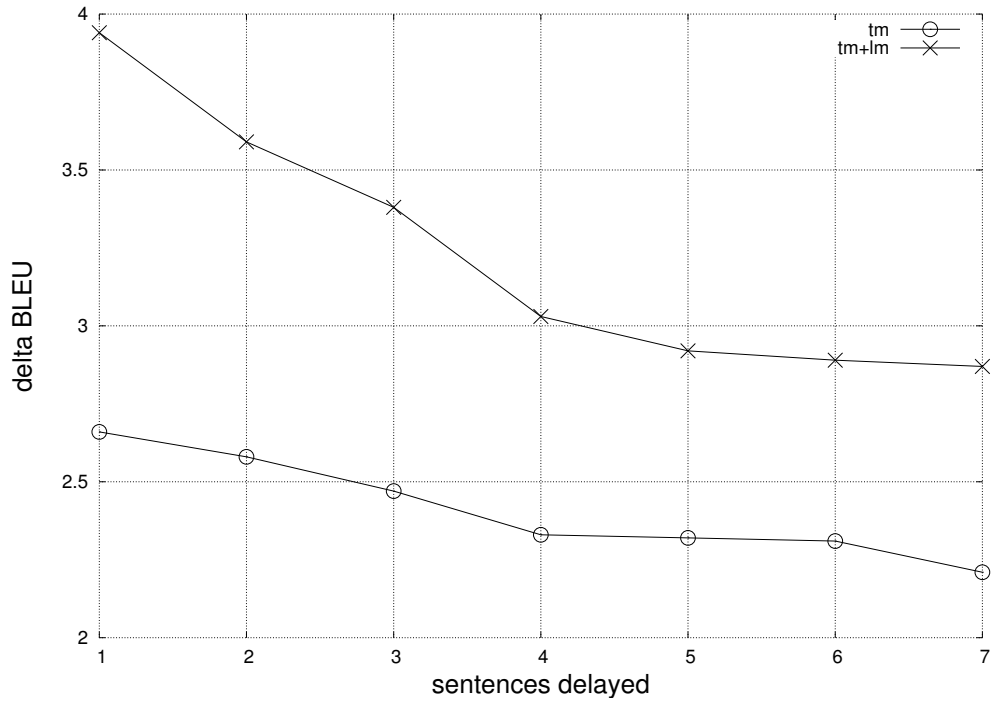

Figure 6.8: Effect of recency of the feedback entered in the cache on adaptation performance.

6.4.2 Delayed Feedback

In a real-world application our adaptation method, although lightweight, could prove to be too slow for practical use. For users to accept SMT output as helpful, it has to be available in short time, usually less than a second. If the adaptation procedure takes longer, feedback will arrive in the system not at the translation of the next sentences, but with a delay. We therefore studied the effect of the feedback recency on adaptation performance. Figure 6.8 plots the drop in score against the number of iterations (i.e. processed sentences) that the feedback exploitation is delayed on an IT development set. There is a significant drop in score that is more pronounced for the language model, but there is still a significant improvement. The *n*-gram cache seems to be especially sensitive to very recent data, while the language model is not that dependent on the locality of the data.

CHAPTER 7

Summary of Contributions and Concluding Remarks

In this thesis we presented models for quantifying cross-lingual similarity between text in source and target language. We applied these models to a number of natural language processing tasks: recognizing cross-lingual textual entailment as a first step towards content synchronization, detecting and deleting noise in a parallel corpus and its influence on translation quality, cross-lingual fuzzy match retrieval for a TM-SMT hybrid and online learning from user post-edits for improved SMT performance. All models are mainly data-driven and are in line with the current state-of-the art in statistical machine translation and machine learning. We anchored our notion of cross-lingual semantic similarity in the alignment links between words and phrases in the source and target language and adapted word and phrase alignment models to the needs of the introduced applications.

We framed the task of recognizing cross-lingual textual entailment for content synchronization as detecting additional, deleted or paraphrased content in a cross-lingual setting. In a statistical learning framework, we combined a collection of features that quantified cross-lingual similarity based on purely structural attributes, cross-lingual and monolingual word alignments and content overlap using the full SMT pipeline. We investigated different machine learning strategies, breaking down entailment relations into directional entailment with binary classification, multi-label and multi-class classification. The two-binary-classifier approach emerged as the winner. We demonstrated the ability of our features and investigated their individual contributions by performing ablation tests. An independent evaluation of our system was conducted by participating in SemEval 2012 with encouraging results; the approach placed first for three out of four language pairs and second for the fourth (Negri et al., 2012).

On a corpus of sentence-parallel data derived from patent documents we performed a study of cross-lingual similarity by investigating noise in automatically aligned parallel data. We quantified and categorized the amount of noise and removed purely structural noise, such as incongruous sentence numbering and headlines. We designed an annotation scheme to derive a hand-annotated data set for training a content-sensitive filter based on word alignments and adapted the system designed for detecting cross-lingual textual entailment for this task. We found filter settings that resulted in a gain in accuracy and F-measure on the hand-annotated gold standard and experimented with different alignment symmetrization heuristics. However, gains in the intrinsic measure did not transfer to an extrinsic task and did not result in an improvement of SMT quality, when comparing a system trained on the filtered data to an unfiltered baseline; however, the amount of training data could be reduced compared to the baseline, while maintaining translation performance. The resulting corpus of 18 million English-French and more than 5 million German-French sentence pairs was released for research purposes and has been adopted as training data in the WMT shared task (Bojar et al., 2014) and other applications (Bertoldi et al., 2014; Pecina et al., 2014).

Motivated by the preference for translation memories over SMT in professional translation environments, we proposed a model for integrating translation-memory-like features into an SMT system. We integrated fuzzy matches by biasing translation towards a match using a statistical model for re-ranking. In contrast to previous approaches, the discriminative model is light-weight and does not rely on phrase-segmentation or alignment between TM source and target, which allows us to integrate partial matches found in the target language only. We investigated a variety of methods for computing cross-lingual similarity to obtain such target matches, inspired by cross-language information retrieval, and found that results with target-language matches are comparable to using a target reference of source-side matches. Furthermore, our hybrid system yielded significant gains in a domain adaptation scenario, where additional monolingual resources in the target language are available. We reported consistent improvements on different technical domains, which are relevant for the localization industry, such as IT, legal and patent data for several different language pairs. By using the matches to re-rank the n-best list, the decoder is treated as a black box, which makes the approach easily portable. Future work in this area might extend the approach to handle multiple fuzzy matches for one source sentence that cover different spans of the input, as proposed in Li et al. (2014). Furthermore, experiments with professional translators could evaluate, whether target fuzzy matches obtained with cross-lingual retrieval are also beneficial for post-editing.

Finally, we modeled and implemented an online learning protocol for a phrase-based SMT system that seamlessly fits in the post-editing workflow managed by the MateCat system, advancing a tighter integration of human and machine translation. We build a document-specific cache from user post-edits, on which a local language and translation model can be

estimated and used at run time to bias the global model. Novel features are the recency of the feedback – user corrections are immediately inserted into the system after becoming available – and making the MT output consistent with regard to the user. Although we did not have the opportunity to test our approach in an interactive setting, we were able to gain large improvements in BLEU over a static baseline system using real-world feedback from human translators on two different technical domains, indicating a considerable reduction in post-editing effort for users. With more refined feedback, such as alignments produced by the human translators, we believe there is a large potential in online adaptation of SMT models.

Bibliography

Abate, T. (2014). Stanford system combines software with human intelligence to improve translation. Retrieved 01/04/2015 from http://news.stanford.edu/news/2014/october/translate-human-machine-10-29-14.html.

Abdul-Rauf, S., Fishel, M., Lambert, P., Noubours, S., and Sennrich, R. (2010). Evaluation of sentence alignment systems. Technical report, Fifth MT Marathon.

Bar Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., and Szpektor, I. (2006). The second PASCAL recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Bar-Hillel, Y. (1951). The present state of research on mechanical translation. *American Documentation*, 2(4):229–237.

Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Ney, H., Tomás, J., Vidal, E., et al. (2009). Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Bentivogli, L., Clark, P., Dagan, I., Dang, H., and Giampiccolo, D. (2011). The seventh PASCAL recognizing textual entailment challenge. In *Proceedings of the Fourth Text Analysis Conference (TAC)*.

Bentivogli, L., Dagan, I., Dang, H. T., Giampiccolo, D., and Magnini, B. (2009). The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Second Text Analysis Conference (TAC)*, volume 9, pages 14–24.

Bertoldi, N. (2014). Dynamic models in Moses for online adaptation. *The Prague Bulletin of Mathematical Linguistics*, 101(1):7–28.

Bertoldi, N., Cettolo, M., and Federico, M. (2013). Cache-based online adaptation for machine translation enhanced computer assisted translation. In *Proceedings of MT Summit XIV*.

Bertoldi, N., Cettolo, M., Federico, M., and Buck, C. (2012). Evaluating the learning curve of domain adaptive statistical machine translation systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*.

Bertoldi, N., Simianer, P., Cettolo, M., Wäschle, K., Federico, M., and Riezler, S. (2014). Online adaptation to post-edits for phrase-based statistical machine translation. *Machine Translation*, 28(3-4):309–339.

Biçici, E. and Dymetman, M. (2008). Dynamic translation memory: Using statistical machine translation to improve translation memory fuzzy matches. In *Computational Linguistics and Intelligent Text Processing*, pages 454–465. Springer.

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

Blain, F., Schwenk, H., Senellart, J., and Systran, S. (2012). Incremental adaptation using translation information and post-editing analysis. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., and Ueffing, N. (2004). Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, page 315.

Bloodgood, M. and Strauss, B. (2014). Translation memory retrieval methods. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., et al. (2014). Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58.

Braune, F. and Fraser, A. (2010). Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 81–89. Association for Computational Linguistics.

Broder, A. Z. (1997). On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings*, pages 21–29. IEEE.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

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

Brown, P. F., Lai, J. C., and Mercer, R. L. (1991). Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 169–176.

Carpuat, M. (2009). One translation per discourse. *SEW-2009 Semantic Evaluations: Recent Achievements and Future Directions*.

Carpuat, M. and Simard, M. (2012). The trouble with SMT consistency. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*.

Castillo, J. and Cardenas, M. (2012). SAGAN: A machine translation approach for cross-lingual textual entailment. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 721–726.

Cettolo, M., Bertoldi, N., and Federico, M. (2013a). Project Adaptation for MT-enhanced Computer Assisted Translation. In *Proceedings of MT Summit XIV*.

Cettolo, M., Bertoldi, N., and Federico, M. (2014). The repetition rate of text as a predictor of the effectiveness of machine translation adaptation. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas (AMTA)*.

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

Cettolo, M., Servan, C., Bertoldi, N., Federico, M., Barrault, L., and Schwenk, H. (2013b). Issues in incremental adaptation of statistical MT from human post-edits. In *Proceedings of MT Summit XIV*, page 111.

Chahuneau, V., Smith, N., and Dyer, C. (2012). pycdec: A python interface to cdec. *The Prague Bulletin of Mathematical Linguistics*, 98:51–61.

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

Chiang, D., Knight, K., and Wang, W. (2009). 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The Annual Conference of*

the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), pages 218–226.

Chiang, D., Lopez, A., Madnani, N., Monz, C., Resnik, P., and Subotin, M. (2005). The hiero machine translation system: Extensions, evaluation, and analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 779–786.

Chiang, D., Marton, Y., and Resnik, P. (2008). Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chokkattu, J. (2014). Unbabel Raises $1.5m To Expand Translation Service, Grow Customer Base. Retrieved 01/04/2015 from http://social.techcrunch.com/2014/07/18/unbabel-raises-1-5m-to-expand-translation-service-grow-customer-base/.

Christensen, T. P. and Schjoldager, A. (2010). Translation-memory (TM) research: What do we know and how do we know it? *Hermes, Journal of Language and Communication Studies*, 44:64–89.

Clark, J. H., Dyer, C., Lavie, A., and 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 (ACL)*, pages 176–181.

Ştefănescu, D., Ion, R., and Hunsicker, S. (2012). Hybrid parallel sentence mining from comparable corpora. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 137–144.

Cui, L., Zhang, D., Liu, S., Li, M., and Zhou, M. (2013). Bilingual data cleaning for SMT using graph-based random walk. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 340–345.

Dagan, I. and Glickman, O. (2004). Probabilistic textual entailment: Generic applied modeling of language variability. In *PASCAL Workshop on Learning Methods for Text Understanding and Mining*.

Dagan, I., Glickman, O., and Magnini, B. (2006a). The PASCAL recognising textual entailment challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190.

Dagan, I., Glickman, O., and Magnini, B. (2006b). The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190. Springer.

Dagan, I., Roth, D., Sammons, M., and Zanzotto, F. M. (2013). *Recognizing Textual Entailment: Models and Applications*, volume 6. Morgan & Claypool Publishers.

Dara, A. A., Dandapat, S., Groves, D., and Van Genabith, J. (2013). TMTprime: A recommender system for MT and TM integration. In *Proceedings of the 2013 NAACL-HLT Demonstration Session*, pages 10–13.

De Souza, J. G., Espla-Gomis, M., Turchi, M., and Negri, M. (2013). Exploiting qualitative information from automatic word alignment for cross-lingual nlp tasks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 771–776.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Denkowski, M., Dyer, C., and Lavie, A. (2014). Learning from post-editing: Online model adaptation for statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Denkowski, M. and Lavie, A. (2011). Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of Sixth Workshop on Statistical Machine Translation*.

Denkowski, M. and Lavie, A. (2012). Challenges in predicting machine translation utility for human post-editors. In *Proceedings of the 10th Conference of the Association for Machine Translation in the Americas (AMTA)*.

Dong, M., Cheng, Y., Liu, Y., Xu, J., Sun, M., Izuha, T., and Hao, J. (2014). Query lattice for translation retrieval. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 2031–2041.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of IBM Model 2. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 644–648.

Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., Setiawan, H., Eidelman, V., and 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)*.

Esplà-Gomis, M., Sánchez-Martínez, F., and Forcada, M. L. (2012). UAlacant: Using online machine translation for cross-lingual textual entailment. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 472–476.

Etzioni, O., Reiter, K., Soderland, S., Sammer, M., and Center, T. (2007). Lexical translation with application to image search on the web. In *Proceedings of MT Summit XI*.

Federico, M., Bertoldi, N., and Cettolo, M. (2008). IRSTLM: An open source toolkit for handling large scale language models. In *Proceedings of Interspeech*.

Fendrich, S. (2012). Sol – stochastic learning toolkit. Technical report, Department of Computational Linguistics, Heidelberg University.

Finch, A., Hwang, Y., and Sumita, E. (2005). Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, pages 17–24.

Fondazione Bruno Kessler (2012). Matecat media kit. Retrieved 10/2012 from http://www.matecat.com/about/media-kit/.

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

Fung, P. and Cheung, P. (2004). Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 57–63.

Gale, W. A. and Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Giampiccolo, D., Dang, H. T., Magnini, B., Dagan, I., Cabrio, E., and Dolan, B. (2008). The fourth PASCAL recognizing textual entailment challenge. In *Proceedings of the First Text Analysis Conference (TAC)*.

Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. (2007). The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

Goodman, J. (1999). Semiring parsing. *Computational Linguistics*, 25(4):573–605.

Goutte, C., Carpuat, M., and Foster, G. (2012). The impact of sentence alignment errors on phrase-based machine translation performance. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas*.

Graham, Y., Salehi, B., and Baldwin, T. (2013). Umelb: Cross-lingual textual entailment with word alignment and string similarity features. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Green, S., Chuang, J., Heer, J., and Manning, C. D. (2014a). Predictive translation memory: A mixed-initiative system for human language translation. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 177–187.

Green, S., Heer, J., and Manning, C. D. (2013). The efficacy of human post-editing for language translation. In *ACM Human Factors in Computing Systems (CHI)*.

Green, S., Wang, S., Chuang, J., Heer, J., and Schuster, S. (2014b). Human effort and machine learnability in computer aided translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 25–29.

Gupta, R. and Constantin, O. (2014). Incorporating paraphrasing in translation memory matching and retrieval. In *Proceedings of the 17th Conference of the European Association for Machine Translation (EAMT)*.

Hardt, D. and Elming, J. (2010). Incremental re-training for post-editing SMT. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas (AMTA)*.

He, H., Lin, J., and Lopez, A. (2015). Gappy pattern matching on GPUs for on-demand extraction of hierarchical translation grammars. *Transactions of the Association for Computational Linguistics (TACL)*, 3:87–100.

He, Y., Ma, Y., van Genabith, J., and Way, A. (2010a). Bridging SMT and TM with translation recommendation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 622–630.

He, Y., Ma, Y., Way, A., and Van Genabith, J. (2010b). Integrating n-best SMT outputs into a TM system. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 374–382.

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

Hopkins, M. and May, J. (2011). Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1352–1362.

Hutchins, J. (1998). The origins of the translator's workstation. *Machine Translation*, 13(4):287–307.

Jehl, L., Hieber, F., and Riezler, S. (2012). Twitter translation using translation-based cross-lingual retrieval. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 410–421.

Jimenez, S., Becerra, C., and Gelbukh, A. (2012). Soft cardinality + ml: Learning adaptive similarity functions for cross-lingual textual entailment. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 684–688.

Jimenez, S., Becerra, C., Gelbukh, A., Bátiz, A. J. D., and Mendizábal, A. (2013). SOFTCARDINALITY: Learning to identify directional cross-lingual entailment from cardinalities and SMT. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Joachims, T. (1999). Making large-scale SVM learning practical. *Advances in Kernel Methods Support Vector Learning*, pages 169–184.

Kay, M. (1980). The proper place of men and machines in language translation. Technical report, Xerox PARC. http://mt-archive.info/Kay-1980.pdf.

Kay, M. (1997a). It's still the proper place. *Machine Translation*, 12(1–2):35–38.

Kay, M. (1997b). The proper place of men and machines in language translation. *Machine Translation*, 12(1-2):3–23.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*, volume 5.

Koehn, P., Axelrod, A., Birch, A., Callison-Burch, C., Osborne, M., Talbot, D., and White, M. (2005). Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 68–75.

Koehn, P. and Haddow, B. (2009). Interactive assistance to human translators using statistical machine translation methods. In *Proceedings of MT Summit XII*.

Koehn, P., Hoang, H., Birch, A., Callison-Birch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 48–54.

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

Koehn, P. and Senellart, J. (2010a). Convergence of translation memory and statistical machine translation. In *Proceedings of the AMTA Workshop on MT Research and the Translation Industry*, pages 21–31.

Koehn, P. and Senellart, J. (2010b). Fast approximate string matching with suffix arrays and A* parsing. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas (AMTA)*.

Kouylekov, M. (2013). Celi: Edits and generic text pair classification. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Kouylekov, M., Dini, L., Bosca, A., and Trevisan, M. (2012). CELI: An experiment with cross language textual entailment. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 696–700.

Kouylekov, M. and Negri, M. (2010). An open-source package for recognizing textual entailment. In *Proceedings of the ACL 2010 System Demonstrations*, pages 42–47.

Krings, H.-P. (2001). *Repairing texts: Empirical investigations of machine translation post-editing processes*, volume 5. Kent State University Press.

Kumar, S., Macherey, W., Dyer, C., and Och, F. (2009). Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL) and the 4th International Joint Conference on Natural Language Processing*, pages 163–171.

Laxström, N., Giner, P., and Thottingal, S. (2015). Content translation: Computer-assisted translation tool for wikipedia articles. In *Proceedings of the 18th Conference of the European Association for Machine Translation (EAMT)*, pages 194–197.

Levenberg, A., Callison-Burch, C., and Osborne, M. (2010). Stream-based translation models for statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Levenberg, A., Osborne, M., and Matthews, D. (2011). Multiple-stream language models for statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Li, L., Way, A., and Liu, Q. (2014). A discriminative framework of integrating translation memory features into SMT. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas (AMTA)*, volume 1, pages 249–260.

Li, Z., Callison-Burch, C., Dyer, C., Ganitkevitch, J., Khudanpur, S., Schwartz, L., Thornton, W. N., Weese, J., and Zaidan, O. F. (2009). Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139.

Liang, P., Bouchard-Côté, A., Klein, D., and Taskar, B. (2006a). An end-to-end discriminative approach to machine translation. In *Proceedings of the Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL)*.

Liang, P. and Klein, D. (2009). Online EM for unsupervised models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 611–619.

Liang, P., Taskar, B., and Klein, D. (2006b). Alignment by agreement. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL*, pages 104–111.

Liu, C., Liu, Q., Liu, Y., and Sun, M. (2012). THUTR: A translation retrieval system. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 321–328, Mumbai, India.

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)*, pages 976–985.

López-Salcedo, F.-J., Sanchis-Trilles, G., and Casacuberta, F. (2012). Online learning of log-linear weights in interactive machine translation. In *Advances in Speech and Language Technologies for Iberian Languages*, pages 277–286. Springer.

Lu, B., Tsou, B. K., Zhu, J., Jiang, T., and Kwong, O. Y. (2009). The construction of a Chinese-English patent parallel corpus. In *Proceedings of MT Summit XII*.

Ma, Y., He, Y., Way, A., and van Genabith, J. (2011). Consistent translation using discriminative learning: A translation memory-inspired approach. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.

MacCartney, B., Galley, M., and Manning, C. D. (2008). A phrase-based alignment model for natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 802–811.

MacCartney, B. and Manning, C. D. (2007). Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200.

Martínez-Gómez, P., Sanchis-Trilles, G., and Casacuberta, F. (2012). Online adaptation strategies for statistical machine translation in post-editing scenarios. *Pattern Recognition*, 45(9):3193–3203.

Mathur, P., Cettolo, M., and Federico, M. (2013). Online learning approaches in computer assisted translation. In *Proceedings of the Eight Workshop on Statistical Machine Translation*.

Mehdad, Y., Negri, M., and C. de Souza, J. G. (2012). FBK: Cross-lingual textual entailment without translation. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 701–705.

Mehdad, Y., Negri, M., and Federico, M. (2010). Towards cross-lingual textual entailment. In *Proccedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 321–324.

Mehdad, Y., Negri, M., and Federico, M. (2011). Using bilingual parallel corpora for cross-lingual textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*.

Melamed, I. D. (2000). Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

Meng, F., Xiong, H., and Liu, Q. (2012). ICT: A translation based method for cross-lingual textual entailment. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 715–720.

Monz, C., Nastase, V., Negri, M., Fahrni, A., Mehdad, Y., and Strube, M. (2011). Cosyne: A framework for multilingual content synchronization of wikis. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, pages 217–218.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *Machine Translation: From Research to Real Users*, pages 135–144. Springer.

Mújdricza-Maydt, É., Körkel-Qu, H., Riezler, S., and Padó, S. (2013). High-precision sentence alignment by bootstrapping from wood standard annotations. *The Prague Bulletin of Mathematical Linguistics*, 99(1):5–16.

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

Munteanu, D. S. and 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)*, pages 81–88.

Nagao, M. (1984). A framework of a mechanical translation between Japanese and English by analogy principle. *Artificial and Human Intelligence*, pages 351–354.

Negri, M., Bentivogli, L., Mehdad, Y., Giampiccolo, D., and Marchetti, A. (2011). Divide and conquer: Crowdsourcing the creation of cross-lingual textual entailment corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–679.

Negri, M., Marchetti, A., Mehdad, Y., Bentivogli, L., and Giampiccolo, D. (2012). Semeval-2012 task 8: cross-lingual textual entailment for content synchronization. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 399–407.

Negri, M., Marchetti, A., Mehdad, Y., Bentivogli, L., and Giampiccolo, D. (2013). Semeval-2013 task 8: cross-lingual textual entailment for content synchronization. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Neogi, S., Pakray, P., Bandyopadhyay, S., and Gelbukh, A. (2012). JU_CSE_NLP: Language independent cross-lingual textual entailment system. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 689–695.

Nepveu, L., Lapalme, G., Langlais, P., and Foster, G. (2004). Adaptive language and translation models for interactive machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Newmark, P. (1988). *A textbook of translation*, volume 1. Prentice Hall New York.

Nie, J.-Y. (2010). Cross-language information retrieval. *Synthesis Lectures on Human Language Technologies*, 3(1):1–125.

Noreen, E. W. (1989). *Computer Intensive Methods for Testing Hypotheses. An Introduction.* Wiley, New York.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), pages 160–167.

Och, F. J. and 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)*, pages 295–302.

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

Ortiz-Martínez, D., García-Varea, I., and Casacuberta, F. (2010). Online lerning for interactive statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Padó, S., Cer, D., Galley, M., Jurafsky, D., and Manning, C. D. (2009). Measuring machine translation quality as semantic equivalence: A metric based on entailment features. *Machine Translation*, 23(2):181–193.

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

Pecina, P., Dušek, O., Goeuriot, L., Hajič, J., Hlaváčová, J., Jones, G. J., Kelly, L., Leveling, J., Mareček, D., Novák, M., et al. (2014). Adaptation of machine translation for multilingual information retrieval in the medical domain. *Artificial intelligence in medicine*, 61(3):165–185.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

Perini, A. (2012). DirRelCond: Detecting textual entailment across languages with conditions on directional text relatedness scores. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 710–714.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Potthast, M., Barrón-Cedeño, A., Stein, B., and Rosso, P. (2011). Cross-language plagiarism detection. *Language Resources and Evaluation*, 45(1):45–62.

Quirk, C., Udupa, R., and Menezes, A. (2007). Generative models of noisy translations with applications to parallel fragment extraction. In *Proceedings of MT Summit XI*, pages 377–384. Citeseer.

Rajaraman, A. and Ullman, J. D. (2012). *Mining of massive datasets*. Cambridge University Press.

Resnik, P. and Smith, N. A. (2003). The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

Sikes, R. (2007). Fuzzy matching in theory and practice. *Multilingual*, 18(6):39–43.

Simard, M. (2014). Clean data for training statistical MT: The case of MT contamination. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas (AMTA)*.

Simard, M. and Foster, G. (2013). PEPr: Post-edit propagation using phrase-based statistical machine translation. In *Proceedings of MT Summit XIV*, pages 191–198.

Simard, M. and Fujita, A. (2012). A poor man's translation memory using machine translation evaluation metrics. In *Proceedings of the 10th Conference of the Association for Machine Translation in the Americas (AMTA)*.

Simard, M., Goutte, C., and Isabelle, P. (2007). Statistical phrase-based post-editing. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 508–515.

Simard, M. and Isabelle, P. (2009). Phrase-based machine translation in a computer-assisted translation environment. In *Proceedings of MT Summit XII*.

Smith, J. and Clark, S. (2009). EBMT for SMT: A new EBMT-SMT hybrid. In *Proceedings of the 3rd International Workshop on Example-Based Machine Translation*, pages 3–10.

Smith, N. A. (2002). From words to corpora: Recognizing translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 95–102. Association for Computational Linguistics.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA)*.

Specia, L., Turchi, M., Cancedda, N., Dymetman, M., and Cristianini, N. (2009). Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Conference of the European Association for Machine Translation (EAMT)*, pages 28–37.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., and Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*.

Stolcke, A. (2002). SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 901–904.

Sykes, T. A. (2009). Growth in translation. Retrieved 24/05/2015 from http://www.inc.com/articles/2009/08/translation.html.

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*.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 2214–2218.

Tillmann, C. (2004). A unigram orientation model for statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 101–104.

Tillmann, C. (2009). A beam-search extraction algorithm for comparable data. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 225–228. Association for Computational Linguistics.

Tillmann, C. and Ney, H. (2003). Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., and Manning, C. (2005). A conditional random field word segmenter for SIGHAN bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, volume 171.

Turchi, M. and Negri, M. (2013). ALTN: Word alignment features for cross-lingual textual entailment. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Ture, F., Elsayed, T., and Lin, J. (2011). No free lunch: Brute force vs. locality-sensitive hashing for cross-lingual pairwise similarity. In *Proceedings of the 34th Annual International ACM Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 943–952.

Ture, F., Oard, D., and Resnik, P. (2012). Encouraging consistent translation choices. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Ueffing, N., Macherey, K., and Ney, H. (2003). Confidence measures for statistical machine translation. In *Proceedings of MT Summit IX*.

Utiyama, M. and Isahara, H. (2003). Reliable measures for aligning Japanese-English news articles and sentences. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 72–79. Association for Computational Linguistics.

Utiyama, M. and Isahara, H. (2007). A Japanese-English patent parallel corpus. In *Proceedings of MT Summit XI*, pages 475–482.

Vanallemeersch, T. and Vandeghinste, V. (2014). Improving fuzzy matching through syntactic knowledge. In *Translating and the Computer 36*.

Vapnik, V. N. and Vapnik, V. (1998). *Statistical learning theory*, volume 1. Wiley New York.

Vilariño, D., Pinto, D., Tovar, M., León, S., and Castillo, E. (2012). BUAP: Lexical and semantic similarity for cross-lingual textual entailment. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 706–709.

Vilarino, D., Pinto, D., León, S., Alemán, Y., and Gómez-Adorno, H. (2013). BUAP: N-gram based feature evaluation for the cross-lingual textual entailment task. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Volokh, A. and Neumann, G. (2011). Using MT-based metrics for RTE. In *Proceedings of the Fourth Text Analysis Conference (TAC)*.

Wang, K., Zong, C., and Su, K.-Y. (2013). Integrating translation memory into phrase-based machine translation during decoding. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 11–21.

Wang, K., Zong, C., and Su, K.-Y. (2014). Dynamically integrating cross-domain translation memory into phrase-based machine translation during decoding. In *Proceedings the 25th International Conference on Computational Linguistics (COLING)*, pages 398–408.

Wäschle, K. (2011). Constructing a German-English parallel patent corpus for statistical machine translation. Master's thesis, University of Heidelberg, Germany.

Wäschle, K. and Fendrich, S. (2012). HDU: Cross-lingual textual entailment with SMT features. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*.

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

Wäschle, K. and Riezler, S. (2012). *PatTR: Patent Translation Resource*. Universität Heidelberg. http://dx.doi.org/10.11588/data/10002 V3 [Version].

Wäschle, K. and Riezler, S. (2012). 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)*.

Wäschle, K. and Riezler, S. (2015). Integrating a large, monolingual corpus as translation memory into statistical machine translation. In *Proceedings of the 18th Conference of the European Association for Machine Translation (EAMT)*.

Wäschle, K., Simianer, P., Bertoldi, N., Riezler, S., and Federico, M. (2013). Generative and discriminative methods for online adaptation in SMT. In *Proceedings of MT Summit XIV*.

Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H. (2007). Online large-margin training for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhao, J., Lan, M., and Niu, Z.-y. (2013). ECNUCS: Recognizing cross-lingual textual entailment using multiple text similarity and text difference measures. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Zhechev, V. and van Genabith, J. (2010). Seeding statistical machine translation with translation memory output through tree-based structural alignment. In *Proceedings of the Fourth Workshop on Syntax and Structure in Statistical Translation (SSST-4)*, pages 43–51.

Acknowledgments

This thesis was supported by a significant number of people, whom I would like to thank profoundly. My supervisor, Prof. Stefan Riezler, for sparking my interest in MT, encouraging me to believe in my ideas and giving me the opportunity to shape my Ph.D. autonomously while always being ready to discuss, develop and sort out ideas. Dr. Karin Haenelt, who kindly agreed to be my second advisor on this dissertation many years after supervising my Bachelor thesis. Dr. Marcello Federico and Dr. Nicola Bertoldi, who advised me during my internship at Fondazione Bruno Kessler, which turned into a fruitful collaboration. Current and former Statistical NLP Group members, Patrick Simianer, Laura Jehl, Sascha Fendrich, Schigehiko Schamoni, Artem Sokolov, and Felix Hieber for the great work environment and both inspiring and entertaining discussions on SMT, ML, NLP and beyond. This thesis would not have been possible without the generous funding through the LGFG Baden-Württemberg and the administrative support by the employees of the Graduate Academy Heidelberg, who facilitated a part-time scholarship. I owe special thanks to Laura, Patrick and Johannes for proof-reading parts of this thesis and providing detailed and helpful feedback on short notice.

Finally, special recognition goes to my family, for their patience and support behind the scenes.