DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS
66

DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS
66

MARK FIŠEL

Optimizing Statistical Machine Translation via Input Modification

TARTU UNIVERSITY
PRESS

Institute of Computer Science, Faculty of Mathematics and Computer Science, University of Tartu, Estonia

Dissertation accepted for the commencement of the degree of Doctor of Philosophy (PhD) on January 13, 2010 by the Council of the Institute of Computer Science, University of Tartu.

Supervisors:

Prof. PhD	Mare Koit University of Tartu Tartu, Estonia
Prof. PhD	Joakim Nivre University of Uppsala Uppsala, Sweden

Opponents:

Prof. PhD	Jörg Tiedemann University of Uppsala Uppsala, Sweden
Senior researcher, PhD	Tanel Alumäe Tallinn University of Technology Tallinn, Estonia

The public defense will take place on March 11, 2011 at 14:15 in Liivi 2-403.

The publication of this dissertation was financed by Institute of Computer Science, University of Tartu.

ISSN 1024-4212
ISBN 978-9949-19-577-0 (trükis)
ISBN 978-9949-19-578-7 (PDF)

Tartu Ülikooli Kirjastus
www.tyk.ee
Tellimus nr. 25

CONTENTS

Acknowledgments

The copmletion of this work would not have been possible without the help of several people. First of all I would like to thank my wife Zane, without whose support and understanding this dissertation would not have been possible. Also, I thank my parents, who greatly supported us during my working on the dissertation.

Next, I express my deepest gratitude to prof. Mare Koit and dr. Heiki-Jaan Kaalep of the University of Tartu and prof. Joakim Nivre of Uppsala University for being my invaluable teachers and guiding me throughout my graduate studies.

I have had the luck and pleasure to have been a student of the Nordic Graduate School of Language Technology (NGSLT), which gave me opportunities that would otherwise have been impossible. In addition I was supported by the Graduate School of Language Theory and Technology and the Graduate School of Information and Communication Technology.

I also thank my friends, especially Jevgeni Kabanov and Konstantin Tretjakov for always being there for me and for sharing my geekiness. In addition I am thankful to my colleague Harri Kirik, who performed additional reviewing of this dissertation and with whom it has been interesting to work together.

The writing of this dissertation has been financially supported by the Center of Excellence in Computer Science (EXCS), the Tiger University program of the Estonian Information Technology Foundation, the Estonian Science Foundation projects 7503 and 5685, the National programme for Estonian Language Technology projects EKKTT09-57, EKKTT06-15, EKKTT09-64 and EKKTT06-9 and the target financed theme SF0180078s08.

The memory- and CPU-intensive computations of this dissertation have been performed on the *krokodill* server of the Bioinformatics, Algorithmics and Data Mining Group (BIIT). I would like to thank its users for having been very tolerant in the past few months and not drowning me with hate mail for abusing the server to finish this dissertation.

Finally, I thank the pre-reviewers of this dissertation, Jörg Tiedemann and Tanel Alumäe for great feedback.

List of Original Publications

1. Fishel, M., Kaalep, H.-J., and Muischnek, K.[1,3]: 2007, *Estonian-English Statistical Machine Translations: the First Results*, in Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA), pp. 278–283, Tartu, Estonia

2. Fishel, M. and Kaalep, H.-J.[1]: 2008, *Experiments on Processing Overlapping Parallel Corpora*, in Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC), pp. 3057–3061, Marrakech, Morocco (indexed in DBLP)

3. Kirik, H. and Fishel, M.[2]: 2008, *Modelling Linguistic Phenomena with Unsupervised Morphology for Improving Statistical Machine Translation*, in Proceedings of the 2nd Swedish Language Technology Conference (SLTC) Workshop on Unsupervised Methods in NLP (3 p.), Stockholm, Sweden

4. Fishel, M.: 2009, *Deeper than Words: Morph-based Alignment for Statistical Machine Translation*, in Proceedings of the 11th Conference of the Pacific Association for Computational Linguistics, (6 p.) Sapporo, Japan

5. Fishel, M. and Kirik, H.[2,3]: 2010, *Linguistically Motivated Unsupervised Segmentation for Machine Translation*, in Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC), pp. 1741–1745, Valletta, Malta (indexed in DBLP)

6. Fishel, M. and Kaalep, H.-J.[1]: 2010, *CorporAl: a Method and Tool for Handling Overlapping Parallel Corpora*, The Prague Bulletin of Mathematical Linguistics **94**, pp. 67–76

7. Fishel, M.: 2010, *Simpler is better: Re-evaluation of Default Word Alignment Models in Statistical MT*, in Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation (PACLIC), pp. 381–388, Sendai, Japan (to be indexed in the ISI Web of Science)

[1] Shared ideas (our contribution estimate at least 50%).
[2] Shared experimental evaluation (equal contribution).
[3] Shared text writing (equal contribution).

CHAPTER 1

INTRODUCTION

Natural human language is a phenomenon that emerges, exists and evolves independently of and is not controlled by any single being or organized group and is therefore analogical to a natural phenomenon. This makes the discipline of natural language processing,[1] which attempts to model the aspects of human languages, a natural science, despite being based on such formal sciences as mathematics, statistics and computer science, and linguistics, the approach of which is formal and more similar to mathematics, rather than natural sciences (Koster, 2005).

By modeling a natural phenomenon we mean finding a formal description, or model, for it; the model may be, and in case of aspects of natural language almost always is, approximate. The purely linguistic approach to modeling natural language phenomena is manually describing it, based on linguistic theory. In contrast to that in corpus linguistics one first manually inspects a large set of examples of the target phenomena and then attempts to generalize them into a formal description.

The data-driven, or machine learning-based approach automates the approach of corpus linguistics by defining a measure of model quality, a model subspace and then searching for a model in this subspace that optimizes the quality measure. In practice a generic model that is based on a set of free parameters is :ften taken and the model search consists of finding the parameter set that maximizes quality.

One of the tasks of natural language processing is machine translation, the aim of which is modeling translation between natural languages. A popular data-driven approach to the task, currently producing state-of-the-art results for many language pairs, is statistical machine translation.

Using a large dataset of translation examples (pairs of sentences or phrases in two languages, corresponding to each other by meaning) statistical machine translation uses statistical learning to find a general model, capable of translating unseen sentences similarly to the provided examples. Therefore in principle

[1] or (human) language technology, or computational linguistics.

the approach is language-independent, meaning that having a sufficient dataset of translations between any two languages anyone can train a machine translation model between these languages without having to know either of them.

This dissertation focuses on optimizing different aspects of statistical machine translation. All of the suggested improvements affect the data, used to train the models, or other initial steps in the training process that are used as input to the succeeding steps – in other words, we are modifying the input to the stages of learning and applying the statistical machine translation models, without changing their core functionality. Naturally this approach is vastly diverse and too general to cover in a single work. Instead we propose several different methods of improving statistical machine translation, all of which share this approach.

Presenting several different improvements under the same title makes the main question of this dissertation composite; the general research question that we want to answer is whether our contributions can produce a significant effect on statistical machine translation by modifying its input without affecting the core functionality of the used models. The more detailed research questions are discussed separately together with each our contribution, to which they are specific.

Every contribution is evaluated according to the effect it produces on the resulting quality of translations, produced by different translation systems; therefore the nature of this work is experimental. We start by giving background information on statistical machine translation and the models, utilized in our experiments, in chapter 2. The experimental setup, common to all our experiments is also presented there.

The next three chapters describe our contributions, whereas every chapter has the same structure. First, the specific area of research is introduced, along with the main questions and the aspect of statistical machine translation that we want to optimize. Then, background information on the area of research is given. After that, we introduce the method of improving the selected aspect. Next, the experimental evaluation of the introduced method is described. Finally, every chapter is concluded with a description of future work.

Our first contribution is presented in chapter 3. We target overlapping parallel corpora – corpora that are based on fully or partially overlapping source documents, and propose a method of gracefully handling such cases, despite possible minor differences in the corpora texts and their level of segmentation. The method can be used to find potential spots of erroneous sentence alignments and to analyze the overlapping parts of the corpora. With regard to machine translation the method is applicable to produce corpora combinations of higher quality and larger size.

Our next contribution (chapter 4) focuses on the technique of segmenting the words of highly inflectional languages into smaller segments prior to training translation models in order to alleviate the sparse data effect, which arises due to the rich morphology of the languages. The introduced method applies the prin-

ciples of linguistics-based segmentation to unsupervised segmentation, with the aim of achieving the same improvement without being dependent on language-specific linguistic tools.

Finally, in chapter 5 the last contribution of this dissertation is presented. We challenge the default models of word alignment – which is used by many state-of-the-art machine translation frameworks as the base of more complex translation units, like phrases or context-free grammar rules. We focus on the default method of unsupervised learning of word alignments and compare its commonly used alignment model to simpler models that precede it with the aim of making word alignment faster and simpler without significant loss in translation quality.

We proceed with a brief description of statistical machine translation together with the specific models that are later used in all our experiments.

CHAPTER 2

THEORETICAL BACKGROUND AND EXPERIMENTAL SETUP

In this chapter we describe statistical machine translation, with a focus on the specific framework and models that our contributions are evaluated on in all the experiments. We extend the description to some preliminary steps that are not specific to the chosen framework or machine translation in general; this is done in order to cover all the pipeline steps that our contributions affect. The covered steps therefore include data preparation, language and translation models, minimum error rate training and result evaluation. The chapter ends with a description of the experimental setup, common to all our experiments. The following description is loosely based on several sources (Koehn, 2010; Och and Ney, 2002; Koehn et al., 2003; Chiang, 2007).

Statistical machine translation treats both the source sentence $\mathbf{f}$ and the target sentence $\mathbf{e}$ as random variables and models the conditional likelihood of $\mathbf{e}$, given $\mathbf{f}$: $p(\mathbf{e}|\mathbf{f})$. In this dissertation we follow the log-linear approach to machine translation (Och and Ney, 2002), where the conditional likelihood of $\mathbf{e}$ is expressed (in vector form) as

$$p(\mathbf{e}|\mathbf{f}) \;=\; p_{\boldsymbol{\lambda}}(\mathbf{e}|\mathbf{f}) \;\propto\; \exp\left(\boldsymbol{\lambda} \cdot \mathbf{h}(\mathbf{e}, \mathbf{f})\right), \tag{2.1}$$

where $\mathbf{h}$ is a vector of feature functions and $\boldsymbol{\lambda}$ – a weight vector. The proportionality sign means that the provided expression is normalized over all possible $\mathbf{e}$'s to obtain a well-formed probability distribution.

The feature functions $\mathbf{h}$ can be used to enforce any kind of dependency between $\mathbf{f}$ and $\mathbf{e}$. The two "classical" feature functions are the language model and the translation model. The language model ignores the source sentence $\mathbf{f}$ and only ensures that the output sentence $\mathbf{e}$ is a grammatically correct sentence and is trained using monolingual corpora of sentences in the target language. The translation model is focused more on conveying the meaning of the source into the target and is trained on translation examples between two languages.

Having the feature functions learned the next step is to tune the parameters $\boldsymbol{\lambda}$. The state-of-the-art approach is minimum error-rate training (Och, 2003), which searches for the parameter values that maximize the quality of the translations, using an automatic translation quality measure.

The search for the best translation estimate $\widehat{\mathbf{e}}$ is defined as

$$\begin{aligned}\widehat{\mathbf{e}} &= \underset{\mathbf{e}}{\operatorname{argmax}}\ p_{\boldsymbol{\lambda}}(\mathbf{e}|\mathbf{f}) \\ &= \underset{\mathbf{e}}{\operatorname{argmax}}\ \boldsymbol{\lambda}\cdot\mathbf{h}(\mathbf{e},\mathbf{f}) \\ &= \underset{\mathbf{e}}{\operatorname{argmax}}\ \sum_{m=1}^{M}\lambda_k h_k(\mathbf{e},\mathbf{f}).\end{aligned}$$

The exponentiation and normalization from eqn. 2.1 can be omitted because the normalization does not depend on the argument $\mathbf{e}$ of the argmax operator and because the optima of any function f and its logarithm $\log f$ are the same. Implementing the argmax search directly is inefficient; the exact approximation of the search depends on the feature functions and especially the translation model feature function.

2.1 Data Preparation

The data for training any statistical models has to be acquired first. In case of machine translation the used data are pairs of text units (sentences or phrases) that are translations of each other. Sets of such translated pairs are commonly referred to as parallel corpora.

In some cases parallel corpora are obtained by manually translating every sentence of a text in one language into another (e.g. (Čmejrek et al., 2004)); however more typically an already translated pair of texts is used. Some examples of text sources are technical documentation (Bojar and Žabokrtský, 2009), subtitles (Tiedemann, 2009), parliamentary proceedings (Koehn, 2005) and legislation (Steinberger et al., 2006; Bojar and Žabokrtský, 2009; Tiedemann, 2009).

Another source of parallel corpora is the world wide web. Web pages with the same content are found using heuristic rules: for instance if the URLs of two pages differ only in that one contains "en" and the other one – "fr" in place of it, it is probable that the pages are translations of each other. Similarly, page pairs can be found from parent pages (i.e. pages linking to the same content in different languages) or sibling pages (i.e. pages with content, linking to its translation). Mining parallel corpora from the world wide web is thoroughly described in (Resnik and Smith, 2003).

In all cases when translations are done without parallel corpora in mind, the exact correspondence between sentences or phrases is not provided. Thus in addition to usual filtering and cleaning of the texts the text units have to be aligned,

which is either done manually by human annotators or automatically. Automatic alignment of text units in parallel corpora frequently uses dynamic programming and sentence similarity features, e.g. typical unit length ratios (Danielsson and Ridings, 1997) or word pair co-occurrences (Varga et al., 2005).

In some cases it is necessary or desirable to further preprocess the parallel corpus before using it for training the models. For example in languages without clear word boundaries (e.g. Chinese or Japanese) first the words are separated to make the translation task easier.

2.2 Language Model

A very typical feature function of any statistical machine translation system, including the ones based on the log-linear framework, is the language model. Its aim is to focus on the output sentence $\mathbf{e}$ and evaluate its quality in the context of its language – i.e. how grammatically and semantically correct the sentence is.

The most common language models, which are also used in our experiments, are statistical language models based on n-grams. Similarly to statistical machine translation, the sentence $\mathbf{e}$, which is a vector of words $(e_1, \ldots, e_m)$, is treated as a random variable and the quality of the sentence is estimated as its probability $p(\mathbf{e})$. N-gram language models make the Markov assumption of the (n-1)-st rank about the words in the sentence; thus

$$\begin{aligned} p(\mathbf{e}) &= \prod_{i=1}^{m} p(e_i|e_{i-1}, \ldots, e_1) \\ &\approx \prod_{i=1}^{m} p(e_i|e_{i-1}, \ldots, e_{\max(1,i-n+1)}). \end{aligned}$$

The conditional probabilities $p(e_i|\ldots)$ can be easily estimated with the maximum likelihood principle. For instance in case $n = 3$:

$$p(e_i|e_{i-1}, e_{i-2}) = \frac{\mathrm{freq}(e_{i-2}, e_{i-1}, e_i)}{\mathrm{freq}(e_{i-2}, e_{i-1})},$$

where $\mathrm{freq}(x, y, z)$ is the frequency of the words x, y and z occurring one after another in the training corpus.

This basic model of sentence probability is augmented with several methods of handling data sparsity and estimating singleton probabilities; a frequently used technique is described in (Kneser and Ney, 1995).

To fit into the log-linear framework the language model feature function is defined as

$$h_{LM}(\mathbf{e}, \mathbf{f}) = \log p(\mathbf{e}).$$

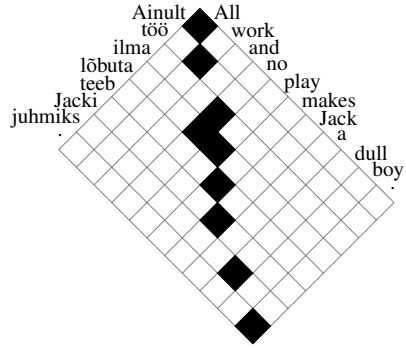


Figure 2.1: An example of word alignment between a pair of English and Estonian sentences. The English words *and*, *a* and *boy* are unaligned

2.3 Translation Model

The aim of the translation model is to estimate how well $\mathbf{e}$ translates $\mathbf{f}$. Direct maximum likelihood estimation is not possible due to the vast amount of different sentences, which is impossible to cover in a parallel corpus. To alleviate this problem most translation models model the sentences by splitting them into words or other smaller chunks.

In this dissertation experiments are performed with two translation models – phrase-based (Koehn et al., 2003) and hierarchical phrase-based (Chiang, 2005). The choice is motivated by the two models being substantially different by approach and representing the current state-of-the-art of statistical machine translation that is based on the surface word forms only (as opposed to, e.g., syntax-based translation, which uses the sentence structure in addition to the surface forms).

During the learning phase both phrase-based and hierarchical phrase-based translation models first align the words of the paired sentences to each other and use the word alignment to construct larger translation units.

2.3.1 Word Alignment

Finding the word alignment of two sentences means finding which words (or phrazes) are translations of each other. An example of word alignment is given in figure 2.1. The method of automatically finding word alignments that is currently considered as default is fully described in (Och and Ney, 2003); it uses a set of models with progressive complexity and representation power, commonly referred to as "the IBM[1] models", originally introduced in (Brown et al., 1993),

[1]International Business Machines corporation, word alignment and initial approaches to statistical machine translation were developed at the IBM Watson Research Center.

plus the HMM-based[2] model, introduced in (Vogel et al., 1996).

All the models treat the word alignment task as asymmetrical by searching for at most one corresponding word from the source sentence $\mathbf{f}$ for each word of $\mathbf{e}$; here $\mathbf{f}$ and $\mathbf{e}$ are vectors of words: $(f_1, \ldots, f_l),\ (e_1, \ldots, e_m)$. The alignment is defined as a vector of indexes:

$$\mathbf{a} : (a_1, \ldots, a_m);\ \forall i\,(a_i \in [0, l])\,,$$

– thus for every i the $\mathbf{e}$ word e_i is aligned to the $\mathbf{f}$ word f_{a_i}; if $a_i = 0$, e_i is said to be unaligned.

Since during translation only the input sentence $\mathbf{f}$ is given and both the output sentence $\mathbf{e}$ and the alignment $\mathbf{a}$ between them are unknown, the key item for the IBM and HMM-based models is the joint conditional probability $p(\mathbf{e}, \mathbf{a}|\mathbf{f})$. Having a pair of sentences their alignment can be found as

$$\widehat{\mathbf{a}} \quad = \quad \underset{\mathbf{a}}{\operatorname{argmax}}\, p(\mathbf{a}|\mathbf{e}, \mathbf{f}),$$

where

$$p(\mathbf{a}|\mathbf{e}, \mathbf{f}) \quad = \quad \frac{p(\mathbf{e}, \mathbf{a}|\mathbf{f})}{p(\mathbf{e}|\mathbf{f})} \quad = \quad \frac{p(\mathbf{e}, \mathbf{a}|\mathbf{f})}{\sum_{\mathbf{a}'} p(\mathbf{e}, \mathbf{a}'|\mathbf{f})}.$$

Without making any independence assumptions and without loss of generality we can write

$$p(\mathbf{e}, \mathbf{a}|\mathbf{f}) \quad = \quad p(m|\mathbf{f}) \prod_{i=1}^{m} p(a_i|\mathbf{a}_{1..i-1}, \mathbf{e}_{1..i-1}, m, \mathbf{f}) p(e_i|\mathbf{a}_{1..i}, \mathbf{e}_{1..i-1}, m, \mathbf{f}); \tag{2.2}$$

the difference between the models is in which independence assumptions are made in this expansion.

The first IBM model (IBM model 1) assumes the output length m probability distribution to be independent of m and $\mathbf{f}$, the distribution of the alignment elements a_i to be uniform and the words of $\mathbf{e}$ to only depend on their corresponding word in $\mathbf{f}$:

$$\begin{aligned} p(m|\mathbf{f}) &\approx \varepsilon, \\ p(a_i|\ldots) &\approx \frac{1}{l+1}, \\ p(e_i|\ldots) &\approx p(e_i|f_{a_i}), \end{aligned}$$

which makes the joint likelihood of $\mathbf{e}$ and $\mathbf{a}$ look like

$$p(\mathbf{e}, \mathbf{a}|\mathbf{f}) \quad = \quad \frac{\varepsilon}{(l+1)^m} \prod_{i=1}^{m} p(e_i|f_{a_i}).$$

[2]Hidden Markov Model (Jelinek, 1976).

In practice ε is canceled out during estimation and the model is only parametrized by the word translation probabilities $p(e|f)$.

IBM model 2 is similar to model 1, except that it explicitly models the alignment by assuming the distribution of a_i to be dependent on m and i in addition to l:

$$p(a_i|\dots) \approx p(a_i|i,l,m),$$

where the conditional probabilities of a_i are also included into the parameter set.

The HMM-based alignment model further refines modeling of the alignment by assuming first-order dependency of the a_i elements and replacing the dependency on indexes with dependency on the relative shift width between the current and the previous alignment indexes:

$$p(a_i|\dots) \approx p(a_i - a_{i-1}).$$

IBM model 3 introduces the concept of fertility, which models the number of words in $\mathbf{e}$ that a single word f_j translates into – in other words, the number of words for which $a_i = j$; its distribution is dependent on the fertility value itself and on the word in $\mathbf{f}$ that it models: $p(n_j|f_j)$, where $n_j \in [1, \infty]$. The alignment model in model 3 is the same as in model 2.

IBM model 4 refines the modeling of alignment in two ways. First, all words in $\mathbf{e}$ that are aligned to a single word in $\mathbf{f}$ are treated as a single group. The position of the first word in a group is chosen, relative to the previously placed group (first-order dependency) and the other words depend on the previously placed word from the same group.

Secondly, in model 4 the position of the aligned word a_i additionally depends on the source and target words. To reduce data sparsity the words are replaced by their parts of speech or any other general classes. This is usually referred to as lexicalization and in practice is used in the HMM-based model and IBM model 3 as well.

Finally, IBM models 3 and 4 introduce deficiency into the models, meaning that some probability mass is wasted on impossible alignments (e.g. the ones placing two words in $\mathbf{e}$ in the same position or leaving some positions in $\mathbf{e}$ unfilled). This is fixed in IBM model 5.

Learning the parameters in the IBM and HMM-based models is done with the Expectation-Maximization (EM) algorithm (Baum, 1972): the source and the target sentences $\mathbf{f}$ and $\mathbf{e}$ are the observed variables and the alignment $\mathbf{a}$ is the hidden variable; the distribution of the variables is expressed through the parameters (word translation probabilities, alignment probabilities, etc.).

The simpler models are introduced not only as a basis of the final model 5; the EM learning starts with the simpler models and after a few iterations uses their parameters as the initialization for the parameters of the next model. Och and

Ney (2003) experiment with the IBM and the HMM-based models and different model combinations and sequences and find that the HMM-based model is a good replacement for the IBM model 2 and that by model 4 the optimal alignment error rate is mostly reached.

IBM models 1 and 2 are simple enough to be trained efficiently with the simple EM; the only thing missing is the likelihood of the observed variables alone, which is also computable efficiently:

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_{\mathbf{a}} p(\mathbf{e},\mathbf{a}|\mathbf{f}) = \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \varepsilon \prod_{i=1}^{m} p(e_i|f_{a_i})\, p(a_i|i,l,m) \\ &= \varepsilon \prod_{i=1}^{m} \sum_{j=0}^{l} p(e_i|f_j)\, p(j|i,l,m). \end{aligned} \tag{2.3}$$

In case of the HMM-based model the same likelihood is computable with the Viterbi algorithm (Viterbi, 1967).

Due to fertility modeling in IBM models, the estimation step becomes even more complex, as the trick of switching places of the sums over $(a_1, \ldots, a_m)$ and the product over i in eqn. 2.3 is not possible anymore. Instead of summing over all alignments, fertility-based models do approximate EM iterations by summing over the most probable alignments.

Finally, the output of the models is still asymmetrical alignment. Och and Ney (2003) describe symmetrization techniques of combining the output of two models for the $\mathbf{e} \rightarrow \mathbf{f}$ and the $\mathbf{f} \rightarrow \mathbf{e}$ direction. The alternatives include the union and the intersection of the two alignments, but the optimal technique, as found by Och and Ney (2003), is a refined method of starting with the intersection and then adding only these alignments from the asymmetrical versions that are adjacent to already existing alignments and cover one of the so far unaligned words.

Detailed surveys of word alignment methods and the recent advances in the field can be found for instance in (Tiedemann, 2004) and (Lopez, 2008).

2.3.2 Phrase-Based Translation Model

Phrase-based translation models (Koehn et al., 2003) are currently one of the state-of-the-art approaches to machine translations. They model translation between sentences in sequences of words – although these are referred to as phrases, they need not be grammatical phrases as dictated by linguistics, but can rather be arbitrary word sequences.

A phrase-based translation model consists of a lexical component, which models the correspondence of the phrases and a reordering component which is responsible for the correct order of the output phrases. Both require finding the corresponding phrase pairs in the source and target sentences.

Phrase pairs can be extracted directly from the unaligned sentence pairs (e.g. Marcu and Wong, 2002). However the current state-of-the-art is the alignment template approach (Koehn et al., 2003; Och and Ney, 2004), which extracts phrase pairs from word alignments. The simple rule is that any two consecutive sequences of words in the source and target sentences can form a phrase pair, as long as no word from either phrases is aligned to word outside the phrases.

Using the word alignment example on figure 2.1, the following pairs (and many others) are all "legal": *work/töö*, *all work and/ainult töö*, *work and no play makes/töö ilma lõbuta teeb* while the pair *play makes/lõbuta teeb* is not legal, since the Estonian phrase includes the word *lõbuta*, which is also aligned to the English word *no*, which is outside the English phrase in the pair.

The resulting phrase pairs are used to create a phrase-based translation lexicon and a reordering model. The weight of a phrase pair is computed as a maximum likelihood conditional probability estimate of the phrases being aligned to each other from $\mathbf{e}$ being aligned to the phrase from $\mathbf{f}$:

$$p(\tilde{e}|\tilde{f}) \quad = \quad \frac{\text{freq}(\tilde{e}, \tilde{f})}{\sum_{\tilde{e}'} \text{freq}(\tilde{e}', \tilde{f})}.$$

The total lexical probability of a segmented sentence pair is then

$$p_{lex}(\mathbf{e}|\mathbf{f}) \quad = \quad \prod_i p(\tilde{e}_i|\tilde{f}_i).$$

Reordering is modeled with the position of the phrase relative to the previously produced phrase: $p(a_i - b_{i-1})$, where a_i is the start position of the i-th phrase in $\mathbf{e}$ and b_{i-1} – the end position of the previously produced ($(i-1)$-st) phrase. A simple solution is penalizing any reordering, independent of the phrases based on the shift width:

$$p(a_i - b_{i-1}) \quad = \quad \xi^{|a_i - b_{i-1} - 1|}, \ \xi \in [0, 1].$$

More advanced solutions include lexicalized reordering, where the changes in the phrase order are conditioned on the phrases $\tilde{e}$ and $\tilde{f}$ themselves, hierarchical phrase-based reordering (Galley and Manning, 2008) and others.

The total probability of the phrase order in a segmented sentence pair is then computed as

$$p_{order}(\mathbf{e}|\mathbf{f}) \quad = \quad \prod_i p(a_i - b_{i-1}),$$

and both the lexical and reordering model are used as separate features in the log-linear framework:

$$h_{lex}(\mathbf{e}, \mathbf{f}) \quad = \quad \log p_{lex}(\mathbf{e}|\mathbf{f}), \ h_{order}(\mathbf{e}, \mathbf{f}) \quad = \quad \log p_{order}(\mathbf{e}|\mathbf{f}).$$

An additional component is commonly included in phrase-based translation models – phrase number penalty. Its main goal is to penalize high numbers of phrases, so that excessive segmentation of the sentences would be avoided.

Translating in the phrase-based framework can e.g. be done with a beam-search algorithm: the phrases in $\mathbf{e}$ are generated left-to-right, every possible continuation of decoding, consistent with the so far translated part and both sentences is considered and weighed with the language, translation and reordering model like in Viterbi search, and the best N paths are kept on every step to avoid exponential search complexity.

Further description of the phrase-based translation model can be found in (Koehn et al., 2003).

2.3.3 Hierarchical Phrase-Based Translation Model

Similarly to phrase-based models, hierarchical phrase-based models (Chiang, 2005) first align the words in the sentence pair and then extract phrase pairs with the alignment template approach. Next, any aligned sub-phrase pair can be replaced with a generic symbol X indicating a gap, which can be filled by any other phrase.

This results in a synchronous context-free grammar, consisting of rewrite rules of the form

$$X \ \rightarrow \ \langle \gamma, \alpha, \sim \rangle,$$

where X is a non-terminal, γ and α are sequences of terminals and non-terminals and $\sim$ specifies a one-to-one correspondence between the occurrences of non-terminals in γ and α.

For example the following (and many other) rules can be extracted from the word alignment on figure 2.1:

$$X \ \rightarrow \ \langle \textit{all } X_1 \textit{ and no } X_2, \textit{ainult } X_1 \textit{ ilma } X_2 \rangle,$$
$$X \ \rightarrow \ \langle \textit{makes Jack } X_1, \textit{teeb Jacki } X_1 \rangle,$$

where the correspondence of the non-terminals is shown with indexes.

In order to give preference to some derivations (and thus also some outputs) over the others weights are assigned to them. The weights of individual rules are computed using a set of weighed features:

$$w(X \ \rightarrow \ \langle \gamma, \alpha, \sim \rangle) \ = \ \prod_i \phi_i(X \ \rightarrow \ \langle \gamma, \alpha, \sim \rangle)^{\lambda_i},$$

where the features include the frequency-based lexical weight of the phrase pair γ/α for both directions ($p(\gamma|\alpha)$ and $p(\alpha|\gamma)$) and others, described in detail in (Chiang, 2007). The feature weights λ_i are passed on to the log-linear framework to be later tuned together with all the other parameters.

The weight of a derivation of the pair $\langle \mathbf{f}, \mathbf{e} \rangle$, which is a set of rules leading from the initial non-terminal S to the pair, is computed as

$$w(D) \quad = \quad \prod_{(X \rightarrow \langle \gamma, \alpha, \sim \rangle) \in D} w(X \rightarrow \langle \gamma, \alpha, \sim \rangle).$$

Finally, the feature function representing the hierarchical phrase-based translation model is defined as

$$h_{hier}(\mathbf{e}, \mathbf{f}) \quad = \quad \log w(D_{\mathbf{e},\mathbf{f}}),$$

where $D_{\mathbf{e},\mathbf{f}}$ is the most probable derivation, corresponding to the sentence pair $\langle \mathbf{e}, \mathbf{f} \rangle$.

Translation for the hierarchical phrase-based translation models is achieved with a combination of beam search with CKY (Cocke-Kasami-Younger) parsing using the grammar. The most probable derivation of $\mathbf{f}$ is constructed, resulting in the translation hypothesis $\widehat{\mathbf{e}}$.

In terms of weaknesses and advantages hierarchical phrase-based translation is expected to generate a better structured output, in comparison to phrase-based translation Chiang (2005). On the other hand, hierarchical phrase-based translation relies on heuristics even more than simple phrase-based translation when estimating its translation unit table, which is a good reason to think that it might be less robust. In practice the two approaches compare differently, depending mostly on the implementation.

A detailed description of hierarchical phrase-based translation models can be found in (Chiang, 2007).

2.4 Minimum Error Rate Training

The parameters $\boldsymbol{\lambda}$ from eqn. 2.1 are learned separately from the feature functions. The initial approach to tuning the parameters, proposed by Och and Ney (2002), is to use maximum class posterior probability criterion over a development set $\{\mathbf{e}_s, \mathbf{f}_s\}$, $s \in [1, S]$:

$$\widehat{\boldsymbol{\lambda}} \quad = \quad \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \sum_{s=1}^{S} \log p_{\boldsymbol{\lambda}}(\mathbf{e}_s | \mathbf{f}_s).$$

However the current state-of-the-art approach is minimum error rate training (Och, 2003). The core idea is to tune the parameters in order to minimize the total error rate of the system on a development set:

$$\widehat{\boldsymbol{\lambda}} \quad = \quad \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} \sum_{s=1}^{S} \mathcal{E}(\mathbf{r}_s, \widehat{\mathbf{e}}(\mathbf{f}_s, \boldsymbol{\lambda})),$$

where $\mathbf{r}_s$ is a reference translation, $\mathcal{E}$ is the error rate measure and

$$\widehat{\mathbf{e}}(\mathbf{f}_s, \boldsymbol{\lambda}) \quad = \quad \underset{\mathbf{e}}{\operatorname{argmax}}\, p_{\boldsymbol{\lambda}}(\mathbf{e}|\mathbf{f}_s).$$

The error rate is commonly defined via any automatic metric of translation quality (e.g. BLEU (Papieni et al., 2001)).

The latter equation for obtaining a translation hypothesis $\widehat{\mathbf{e}}(\mathbf{f}_s, \boldsymbol{\lambda})$ essentially means performing translation all over again, which is expensive. To optimize the criterion first a set of N most probable translations $\mathbf{e}_{s,1}, \ldots, \mathbf{e}_{s,N}$ (the "n-best list") for $\mathbf{f}_s$ is computed and the search is conducted inside the list instead of unrestricted $\mathbf{e}$ space:

$$\widehat{\mathbf{e}}(\mathbf{f}_s, \boldsymbol{\lambda}) \quad = \quad \underset{\mathbf{e} \in \{\mathbf{e}_{s,1}, \ldots, \mathbf{e}_{s,N}\}}{\operatorname{argmax}} \; p_{\boldsymbol{\lambda}}(\mathbf{e}|\mathbf{f}_s).$$

The algorithm for searching for the optimal set of parameters $\boldsymbol{\lambda}$ works iteratively. Each iteration consists of producing an n-best list for the development set, based on the feature functions and the current parameter estimate and then finding a new parameter estimate; this is repeated until convergence. The algorithm and experiments with using different quality metrics for the error rate are presented in (Och, 2003).

2.5 Evaluation

Finally, having a ready translation system it is necessary to assess the quality of its translations. This can be done either manually or automatically. In this dissertation, due to a rich selection of language pairs in all experiments we only use automatic evaluation.

2.5.1 Manual Evaluation

Manual evaluation by a human evaluator is usually made easier by being split into parts – evaluating the adequacy of the translation in relation to the input and the fluency of the translation with regard to the target language. To evaluate the adequacy the person has to be (near-)bilingual in the source and target languages while fluency only requires the person to be fluent in the target language.

Both fluency and adequacy are typically measured with an absolute scale of numeric grades, corresponding to levels of quality.

It is necessary to note that fluency and adequacy are not entirely independent. The more errors the translation has, the more likely it is that the errors affect the conveyed meaning. In the extreme case a translation with the worst fluency cannot possibly have high adequacy.

Manual evaluation is the most exact method of assessing translation quality, but it is also time-consuming and expensive.

2.5.2 Automatic Evaluation

An alternative to manual evaluation is provided by several existing automatic metrics of machine translation quality. These metrics are themselves evaluated by finding how highly they correlate with human judgment. The correlation may vary for the same metric when applied to different language pairs and domains (see e.g. Callison-Burch et al., 2010). The majority of automatic metrics works by comparing the translation produced by the system to a reference translation (or several translations) produced by a human translator.

The most popular metric is the BLEU (bi-lingual evaluation understudy) score (Papieni et al., 2001). The score of a translation is a number between 0 and 1 and is based on the geometric mean of the precision of words, bigrams, trigrams and so forth until 9-grams. Including n-grams indirectly evaluates the order of the words in the translation. The geometric mean of the n-gram precisions is additionally multiplied by a brevity penalty, which does the role of recall by not allowing the translations to "cheat" by including a small number of words with high confidence and achieving high precision.

(Papieni et al., 2001) reported the BLEU score to have high correlation with human judgments. Although Callison-Burch et al. (2006) showed that BLEU gives preference to a certain type of translation system and contradicts human judgments in case of other types of systems, it is still extensively used in machine translation research.

Another popular metric is the NIST score (NIST, 2002). It is also based on n-gram precision, but adds additional weight to rare n-grams, assuming that it is easier to translate frequent n-grams than rare ones. In addition the geometric mean is replaced with arithmetic mean.

Other metrics include METEOR (Banerjee and Lavie, 2005), WER (word error rate) and several others (see e.g. (Callison-Burch et al., 2010)).

2.6 Experimental Setup

Here we describe the experimental setup, common to all evaluations in this dissertation. As mentioned in the previous chapter, we use the framework of log-linear machine translation models, which is the current state-of-the-art in statistical machine translation. It is used for evaluating all of the introduced methods and modifications of this dissertation with two kinds of translation models phrase-based (Koehn et al., 2003) and hierarchical phrase-based (Chiang, 2005).

We use the Moses toolkit (Koehn et al., 2007) as the implementation of phrase-based models and the Joshua toolkit (Li et al., 2009) as the implementation of hierarchical phrase-based models; minimum error rate training is included in both toolkits. Word alignment for both toolkits is done with GIZA++ (Och and Ney, 2003).

The used implementation of language models is the SRI LM toolkit (Stolcke, 2002). All experiments use interpolated 5-gram language models with Kneser-Ney discounting (Kneser and Ney, 1995).

Every introduced method is evaluated on several language pairs in order for the results not to be language pair-specific. The choice of the exact language pairs is different in every experiments and depends on the specific task at hand; however the Estonian-English pair is always included, which is motivated geographically.

Translation quality is evaluated with the BLEU and NIST scores; due to the number of language pairs manual evaluation was unfortunately not feasible for us. The size of the randomly selected held-out sets for parameter tuning and evaluation is always 2500 sentences, which is motivated by the example of the latest machine translation open evaluation events (Callison-Burch et al., 2009, 2010).

Where required, the statistical significance of the score differences between two translations is found with the enhanced paired bootstrap resampling method of Riezler and Maxwell (2005). Significance testing is applied to both the BLEU and NIST scores.

CHAPTER 3

HANDLING OVERLAPPING PARALLEL CORPORA

This chapter presents our first contribution of this dissertation. The presented material is published in (Fishel and Kaalep, 2008) and (Fishel and Kaalep, 2010).[1]

The work described in this chapter belongs to the domain of corpora preparation before using them to train statistical systems. Specifically we focus on overlapping parallel corpora – i.e. pairs of sentence-aligned bilingual corpora with the same language pair that are based on partially or fully overlapping sources. Such a situation can occur, for instance, when the same source documents are independently used to create corpora at different times or different institutions. Alternatively, independent adding of any kind of markup to the corpus can result in slightly different versions of the underlying text.

Processing overlapping corpora can be quite problematic. Simply concatenating them is not a valid solution: as a result the data distribution of the combined corpus will be skewed, since the samples from the overlapping part will be overrepresented. At the same time using the standard `diff` utility is not guaranteed to elegantly solve the problem of detecting the repeated and unique samples. Typically the texts have differences in representation, or some typing or aligning errors might have been fixed or introduced due to different versions of the source documents. In addition some samples (i.e. sentence pairs) could have been omitted from one of the corpora. Finally, the level of segmentation might differ because of the approach (e.g. sentence vs. paragraph-level segmentation) or the tools used for aligning the corpora.

On the other hand, if those difficulties could be overcome, the overlap could be exploited to many advantages. By comparing the two corpora the potential alignment error spots can be found and the size of both can be increased on the account of omitted sentence pairs from one or the other corpus; also, the level of

[1] Our contribution in both papers includes shared ideas and their design with the other author, fully performing the experimental evaluation and writing the text.

segmentation of both corpora can be increased, which reduces the average sentence length and makes the subsequent steps, like word alignment and phrase pair extraction, more accurate. Finally, if it can be assumed that one of the corpora is much more accurate, the other corpus can be proofed against it to assess or improve its quality.

We propose a method for processing the overlapping parts of parallel corpora, aimed at detecting the matching and mismatching samples. Using our method it is possible to compare the overlapping corpora or combine them and use the result for training translation systems. The method and its implementation are described in detail in sections 3.2 and 3.3.

The main questions that we want to answer are whether processing the corpora can provide useful information about them and whether translation models trained on the resulting combined corpora have better scores than the ones, trained on the baseline corpora. Results of the experiments, conducted in order to address these questions, are described in section 3.4. Finally, in section 3.5 we discuss the possible future development of this research.

We start by giving some background information on the related work and overlapping parallel corpora in section 3.1.

3.1 Background and Related Work

To our knowledge the only work, addressing the issue of overlapping parallel corpora, is (Kaalep and Veskis, 2007), who analyze the quality of the target data manually and propose an automatic quality metric, based on the observations. The pair of corpora that Kaalep and Veskis (2007) analyze is also used in our research and consists of the JRC-Acquis multilingual parallel corpus (version 2.2) (Steinberger et al., 2006) and the corpus of the University of Tartu.[2] Both include "unique" parts (i.e. documents, not present in the other corpus) since in the second corpus several documents were omitted from the joint part, but it also includes Estonian laws with their English translations, in addition to the EU legislation.

Another example is the JRC-Acquis corpus itself, since it provides two alternative alignments for every language pair it includes – done with Vanilla (Danielsson and Ridings, 1997) (implementing the algorithm of Gale and Church (1993)) and HunAlign (Varga et al., 2005) (implementing an custom algorithm, similar by approach to the one of Moore (2002)). This means that, although the text might be exactly the same, the level of segmentation can be different in the two versions (due to possible different grouping of sentences by the two aligners into N-to-M sentence pair chunks). In addition, it is common practice for sentence aligners to exclude sentence pairs that seem to be untrustworthy, which also means that the material of the two versions is not the same.

[2] `http://www.cl.ut.ee/korpused/paralleel/?lang=en`.

In the experimental part of this work we focus on the two presented cases; however there are other examples as well. The Hunglish corpus (Varga et al., 2005) includes EU legislation in Hungarian and English, obtained from the same sources as the JRC-Acquis. One part of the CzEng corpus (Bojar and Žabokrtský, 2009) also consists of EU legislation in Czech and English, whereas the source documents were taken directly from JRC-Acquis, but the text processing and alignment was done all over.

A whole domain of corpora is a potential source for multiple versions of the same text – movie subtitles. These constitute parts of CzEng, Hunglish and the OPUS corpus (Tiedemann, 2009); the University of Tartu also has a small corpus of subtitles – it shares the source with OPUS (OpenSubtitles database) and is under development.

However subtitles are in a sense a special case as there are often many subtitle translations for the same movie done by different translators. Thus versions of the same subtitle can be phrased in a too different way to be compared with simple text processing. Instead of trying to combine them into one corpus it would make more sense to use the alternative translations of the same source to create a parallel corpus with multiple references.

We proceed with the description of our method of processing the overlapping parts of parallel corpora.

3.2 Method Description

Let us start with an example of two parallel corpora containing an overlap (figure 3.1). The third sentence pair of corpus B is omitted from corpus A and the third sentence pair of corpus A is segmented into two sentence pairs (numbers four and five) in corpus B, which makes the level of segmentation of the latter slightly higher. Also there are slight differences in punctuation between the two corpora.

Knowing both English and Estonian, it is easy to see that the English sentence from second sentence pair in corpus B got distorted, which makes the pair an erroneous alignment. Without knowing either of the languages, it can still be detected that one of the second sentence pairs in both corpora is probably erroneous – since the Estonian parts are practically the same, while the English parts are nothing like each other.

Very simply put, this language-wise comparison is the basis of the method that we are about to introduce, which involves two steps.

The first step consists of aligning the corresponding language parts to each other. In the example above that means the English parts of the two corpora are aligned, as are the Estonian parts. The alignment at this stage supports approximate matching of the sentences to account for slight differences (like typing

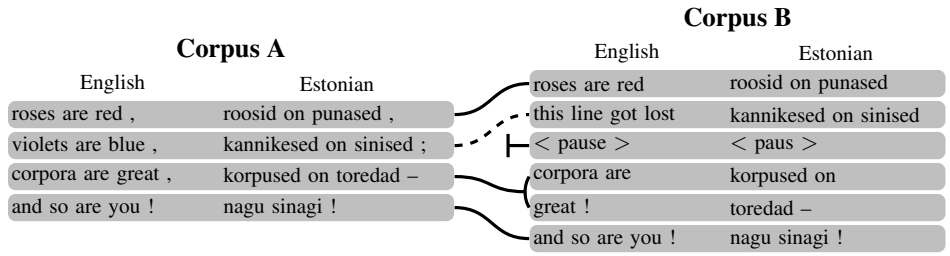


Figure 3.1: An example of overlapping parallel corpora with the correspondence of the two corpora shown. Second sentence pair of corpus B is an erroneous alignment

errors, punctuation, other document version differences) and aligning several sentences with each other to account for different segmentation levels. See figure 3.2 (a) for an illustration of this step.

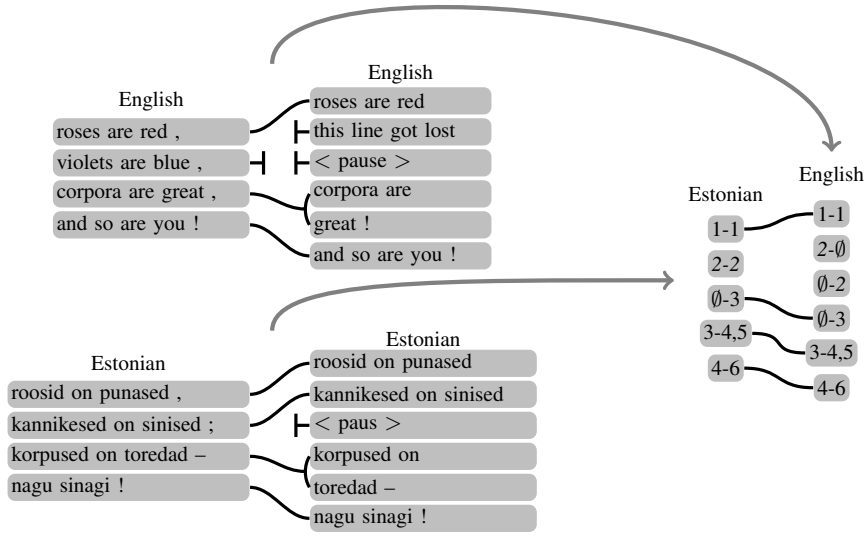
In the second step the resulting language alignments are themselves aligned to each other. Here the aim is to find the matching and mismatching alignment chunks. This way whenever in one language two sentences match while in the other language the sentences from the same sentence pairs do not match, this will be detected as an alignment error. The same result will occur when the alignment errors are caused by additional segmentation of the sentences; this is rather typical of non-lexical aligners, such as Vanilla. See figure 3.2 (b) for an illustration of the second step; notice the resemblance between the resulting alignment and the correspondence of the parallel corpora in the example on figure 3.1 (a match, a mismatch and three matches).

In the following subsections we will describe in detail the two steps of the algorithm, as well as sentence approximate matching.

3.2.1 Aligning the Corresponding Language Parts

The first step is in essence very similar to the original task of bilingual sentence alignment itself. However, whereas the latter means comparing different languages and therefore requires, for instance, probabilistic solutions, in this case the task is much simpler, since both parts are in the same language and it suffices to compare the sentences using simple text processing. The only problem is that instead of strict comparison of the sentences, here approximate comparison is required due to possible slight differences in different corpora.

The aligning task is therefore analogical to the longest common subsequence problem, where corpora units (i.e. sentences or paragraphs) are matched to each other. Here the alignment of the two texts is computed using generalized edit distance. The cost of substituting a unit for another equals the similarity between



(a) First step: aligning the corresponding language parts

(b) Second step: aligning the resulting alignments

Figure 3.2: The two steps of processing the overlapping parallel corpora from the example on figure 3.1. ∅ stands for an empty counterpart, i.e. in zero-to-one alignments

them, which is obtained using approximate sentence matching, explained in the next subsection. The cost of insertion/deletion is always 1.

In addition to 1-to-1 matches all N-to-M pairs are also considered up to a predefined limit (in our implementation – 10 by default). This enables detecting matching units even if the segmentation level is very different in the two corpora.

3.2.2 Approximate Sentence Matching

Kaalep and Veskis (2007) use Levenshtein distance and check whether the distance between two sentences exceeds 1% of the average length of the two sentences. Other string similarity metrics applied to written text include several from the edit distance family (the Needleman-Wunsch metric, the Smith-Waterman metric, etc), the Jaro metric and others.

In the current work we use practically the same sentence matching method, but use generalized edit distance instead of Levenshtein distance. For instance the weight of replacing/inserting digits is extremely high, so that e.g. sentences “article 3” and “article 5” will not be considered to match with no matter what edit

distance percentage threshold. On the other hand operations on empty symbols (spaces, tabs) and punctuation have low weights. This allows to set the percentage threshold higher without adding obvious matching errors.

3.2.3 Reducing Computational Complexity

Both the language-wise alignment and the approximate sentence matching in our method is based on computing the edit distance matrix. Its time complexity is $O(mn)$, where m and n are the lengths of the sentences to compare – this mainly means that full corpora cannot be directly processed, since that would mean enormous processing time. Our solution is splitting the corpora into separate documents, aligned to each other. The way corpora are split is not part of the tasks of this work and is considered to be solved by the time of applying our method.

The corpora used in our experiments (the JRC-Acquis and the corpus of the University of Tartu) are originally composed out of separate documents, augmented with unique codes, which solves the problem for our case and at the same time enables matching the documents to each other easily.

At this point the method still has quadratic complexity in terms of the number of sentences in a document. However, two facts about the weights of the symbol replacements in our generalized edit distance enable us to reduce its search space: the weight of "replacing" a symbol with exactly the same symbol is always zero and all the other weights are greater than zero. Using these two facts it is possible to avoid traversing the whole $m \times n$ matrix in case of both approximate sentence matching and document language-wise alignment with the following optimization techniques:

- the beginnings and ends of both sequences are checked on the subject of identical initial/final subsequences; i.e. if the first/last N symbols in the sequences are the same, this head/tail can be immediately selected as the optimal path and search can be continued from their ending points
- if the aim is to just compute the distance between two sentences and reject it, if the distance is above a certain threshold, it is possible to discontinue the search at the alternative paths where the threshold is already reached. In practice it usually means "trimming" the corners of the search matrices.
- the previous technique also means that as soon as all the alternatives reach the threshold, the search is halted. This way no time is wasted on aligning the sentences that are known to mismatch.
- to speed up document alignment first the document pair is processed with a simple Levenstein distance search (strict comparison, only one-to-one or one-to-zero matches) and the sentence pairs that exactly match each other

are used as milestones for the full generalized edit distance search – the matrix of the whole document is therefore split by certainly matching sentence pairs into smaller squares

3.2.4 Aligning the Alignments

As soon as the language part alignments are obtained, their correspondence to each other is to be determined. Although different language parts are to be compared here, only the alignments between unit numbers are compared, which again enables using direct comparison. In this case it is accomplished again using edit distance, but this time with the simple Levenshtein distance of the alignment cells. Thus equality of the alignment elements indicates matching alignments while 1-to-1 inequality or 1-to-0/0-to-1 matches indicate mismatching alignments.

It is important to note that a mismatch between two alignments does not indicate, which of the corpora has an erroneous alignment; instead, it shows a potential spot, where at least one of the corpora has an error. If one of the corpora is known to be accurately aligned, the errors of the other corpus can be corrected automatically this way. Otherwise the spots can be manually post-processed and the errors in the appropriate corpus – corrected.

On the other hand a match between alignments also merely indicates that the two corpora have matching alignments. This can occur both in case of correct alignments and coinciding erroneous alignments, though the latter is less likely (depending on the used alignment method).

3.3 Implementation

In this section we will focus on the tool implementing the introduced method and the specifics of the implementation. We named the tool "CorporAl", reflecting the core idea of the method, which is aligning the parallel corpora to each other.

3.3.1 Functionality

The main functionality is of course the method, introduced in the previous section. The tool expects the corpora to be split into a set of matching document pairs; it then finds the alignments between the corresponding language parts and then aligns the resulting alignments to each other. The result is the correspondence of the overlapping parts: that is the matches and mismatches between the corpora sentence pairs for each document, together with the match types (i.e. a sentence pair is omitted, matches one or more sentence pairs in the document of the second corpus).

The final expected result of the tool is however not the correspondence, but rather a ready new combined corpus. In order to generate it, however, it is required

to define the exact behavior for the program. Namely, depending on the purpose the user might want to include or exclude:

- sentence pairs that are only present in one or the other corpus
- sentence pairs that are present in both corpora and match
- sentence pairs that are a mismatch between the corpora

For instance, if the aim is to generate the biggest possible corpus, the correct strategy would be to include all matching and omitted sentences. Alternatively, if the aim is to maximize the quality of the resulting corpus, then everything should be excluded except the matching sentence pairs, present in both corpora.

Our tool accepts input arguments, defining what to do with every case presented above. For the first two cases it trivially allows to either include or exclude the sentence pairs unique to the first or the second corpus and the matching sentence pairs. In case of mismatches it is possible to either skip the whole chunk or define one of the corpora as the more trustworthy one and include the sentence pairs from it. If the matching sentence pairs are to be included, the tool automatically includes the sentence pairs with a higher level of segmentation – that is, if two sentence pairs in one corpus match three sentence pairs in the other corpus, the latter will be included, regardless of which corpus is defined as more trustworthy.

Naturally in addition to corpora combination it is possible to configure the tool to just output the alignment of one corpus to the other and then use it for further processing.

3.3.2 Usage Information

The CorporAl tool is distributed as an open-source project and is available from SourceForge.[3] It is available for both downloading and checking out its code through the Subversion versioning system. The implementation is done as a PERL script and thus can be run on any platform that has a PERL interpreter. The interface of the tool is command-line-based. The main script is `bin/comb.pl`; when evoked with a `--help` switch, it provides detailed information on the expected arguments and available options.

The expected format of the parallel corpora is the one of the JRC-Acquis corpus: both languages are included in a single file and every sentence is surrounded with XML-tags, specifying which language the sentence belongs to; aligned sentences are put one after another. For instance in case of the example on figure 3.1 corpus A would look this way:

[3] `http://corporal.sf.net`.

Max-size

English	Estonian
roses are red ,	roosid on punased ,
violets are blue ,	kannikesed on sinised ;
< pause >	< paus >
corpora are	korpused on
great !	toredad –
and so are you !	nagu sinagi !

Max-accuracy

English	Estonian
roses are red ,	roosid on punased ,
corpora are	korpused on
great	toredad –
and so are you !	nagu sinagi !

Table 3.1: The max-size and max-accuracy combinations of the example corpora on figure 3.1

```
<en>roses are red ,</en>
<et>roosid on punased ,</et>
<en>violets are blue ,</en>
<et>kannikesed on sinised ;</et>
...
```

with `et` denoting Estonian and `en` – English. The package of the tool also includes scripts for transforming the JRC-Acquis corpus format into the format expected by tools like Moses and GIZA++ and back.

3.4 Experiments

Our final aim was to test the presented method in practice. We focused the experiments on two cases of overlapping parallel corpora, described in section 3.1: first, the corpus of the University of Tartu (further referred to as UT) and the Estonian-English part of JRC-Acquis version 2.2 (further – JRC2) and second, the HunAlign and Vanilla versions of JRC-Acquis version 3 (further – JRC3 hun/van). In the second case we used four language pairs: English-Estonian, Estonian-Latvian, English-Latvian and German-English – thus a total of five corpus pairs were used (one UT+JRC2 language pair and four JRC3 hun/van language pairs).

First we present the results of processing the corpora and some conclusions

UT+JRC2, et-en	#docs	#snt pairs	#en words	#et words
Just UT	2048	134684	$3.12 \cdot 10^6$	$2.17 \cdot 10^6$
Just JRC2	5807	205025	$4.86 \cdot 10^6$	$3.25 \cdot 10^6$
Common UT	2009	93152	$1.88 \cdot 10^6$	$1.27 \cdot 10^6$
Common JRC2	2009	68165	$1.67 \cdot 10^6$	$1.09 \cdot 10^6$
Max-size	2009	98946	$2.03 \cdot 10^6$	$1.36 \cdot 10^6$
Max-acc	2009	56234	$1.35 \cdot 10^6$	$0.88 \cdot 10^6$

Table 3.2: Results of processing the UT and JRC2 corpora pair: number and sizes of the documents in the common parts of the corpora, documents present in just one corpus and the resulting max-size and max-accuracy combinations

that can be drawn from the results. We then go on to testing whether our method of corpora processing leads to improved translation scores of a phrase-based and a parsing-based statistical translation system.

3.4.1 Processing Overlapping Parallel Corpora

The first step in processing the parallel corpora was to identify the matching document pairs. In case of both UT and JRC it is easy, since each document is marked with a unique CELEX code, which was preserved while constructing both corpora.

However, unlike the UT corpus, where each document with its CELEX code is stored in a separate file, the JRC2 and JRC3 corpora are given in one XML file. Therefore the first step was extract each document into a separate file. Then the documents could be grouped by their CELEX codes into three groups: documents unique to one of the corpora in a pair and the ones present in both corpora in a pair. The last group was further split into identical and differing documents.

Finally the differing documents present in both corpora of a pair were processed with the CorporAl tool. We generated two different versions of the combination: one (called max-size) prioritized the resulting corpus size and the other one (called max-accuracy) prioritized the resulting accuracy. The latter thus included only the matching sentence pairs, present in both corpora. The former in addition to that included the sentence pairs, unique to one of the corpora, and in case of mismatches – the sentence pairs of the HunAlign version of JRC2 and JRC3, which we defined as the more trustworthy; this choice was motivated by the work of Kaalep and Veskis (2007), who found that the Vanilla version of JRC2 contains much more errors.

Applying the max-size and max-accuracy principles to the example on figure 3.1 we obtain the two corpora in table 3.1; here corpus A is defined as the more trustworthy one. Thus all the matches are included from corpus A, except for

	UT, %	JRC2, %
∅	7.12	9.89
0-1	0.00	8.25
1-0	32.57	0.00
1-1	59.30	81.04
1-2	0.06	0.17
2-1	0.91	0.62
2-2	0.00	0.00
3-1	0.01	0.00

Table 3.3: Frequency of the match types between sentence pairs of the UT and JRC2 corpora pair; given as proportion of sentences from a match type in a corpus. The ∅ stands for mismatching (i.e. probably erroneously aligned) sentence pairs

its third sentence pair, which is segmented in corpus B into two pairs and thus included from there. In addition in the max-size version the second sentence pair from corpus A is included in place of the mismatch and the third sentence pair of corpus B, missing from corpus A, is also added.

The sizes of the documents and the resulting corpora parts for the UT+JRC2 pair are presented in table 3.2. The frequencies of the types of sentence pair matches are given in table 3.3, showing how many N-to-M matches between sentence pairs have been detected; the ∅ stands for mismatching sentence pairs. The same information for the four JRC3 pairs is given in tables 3.4 and 3.5.

No identical document pairs were found between UT and JRC2, which is expected, given the difference of how and when the two corpora were composed. The max-size version of the combination is bigger than both its sources, while the max-accuracy version is noticeably smaller. Still, some conclusions can be drawn from the result of processing the UT+JRC2 pair.

Looking at the match type frequencies (table 3.3), it can be seen that more many-to-one matches have the "many" part on the side of the UT corpus than on the JRC2 side. However, since the many-to-one matches constitute just a small percent of all the matches (below 1% on both sides), it can be concluded, contrary to our initial assumption, that the levels of segmentation of the UT and JRC2 corpora overlapping parts are practically the same.

It can also be seen that a lot of material has been omitted from the JRC2 corpus: one third of the UT corpus comprised sentence pairs, not present in JRC2.

It is hard to make any conclusions about the quality of the alignments in the corpora, since neither corpora can be assumed to be mostly correct. According to Kaalep and Veskis (2007), the Vanilla alignments of the UT corpus include many shift errors while in the HunAligned JRC2 most errors reside around zero-to-one sentence alignments. However the portion of mismatches between the two

JRC3, en-et	#docs	#snt pairs	#en words	#et words
Just Hun	5	63529	$0.80 \cdot 10^6$	$0.73 \cdot 10^6$
Just Van	173	8392	$0.28 \cdot 10^6$	$0.22 \cdot 10^6$
Identical	12536	535577	$13.02 \cdot 10^6$	$9.44 \cdot 10^6$
Common Hun	10645	711739	$18.24 \cdot 10^6$	$13.05 \cdot 10^6$
Common Van	10645	648319	$18.10 \cdot 10^6$	$12.85 \cdot 10^6$
Max-size	10645	711815	$18.24 \cdot 10^6$	$13.05 \cdot 10^6$
Max-acc	9976	548851	$15.25 \cdot 10^6$	$10.56 \cdot 10^6$

JRC3, en-lv	#docs	#snt pairs	#en words	#lv words
Just Hun	4	63528	$0.80 \cdot 10^6$	$0.75 \cdot 10^6$
Just Van	183	9076	$0.26 \cdot 10^6$	$0.30 \cdot 10^6$
Identical	12206	537370	$13.22 \cdot 10^6$	$10.69 \cdot 10^6$
Common Hun	10354	697787	$17.62 \cdot 10^6$	$14.65 \cdot 10^6$
Common Van	10354	638393	$17.55 \cdot 10^6$	$14.41 \cdot 10^6$
Max-size	10354	697867	$17.62 \cdot 10^6$	$14.65 \cdot 10^6$
Max-acc	9769	542661	$15.00 \cdot 10^6$	$11.74 \cdot 10^6$

JRC3, et-lv	#docs	#snt pairs	#et words	#lv words
Just Hun	3	63528	$0.73 \cdot 10^6$	$0.75 \cdot 10^6$
Just Van	54	3374	$0.06 \cdot 10^6$	$0.14 \cdot 10^6$
Identical	19145	835000	$14.00 \cdot 10^6$	$15.79 \cdot 10^6$
Common Hun	3536	458746	$8.31 \cdot 10^6$	$9.72 \cdot 10^6$
Common Van	3536	438005	$8.29 \cdot 10^6$	$9.62 \cdot 10^6$
Max-size	3536	458898	$8.31 \cdot 10^6$	$9.72 \cdot 10^6$
Max-acc	3443	407281	$7.67 \cdot 10^6$	$8.65 \cdot 10^6$

JRC3, de-en	#docs	#snt pairs	#de words	#en words
Just Hun	4	66148	$0.84 \cdot 10^6$	$0.80 \cdot 10^6$
Just Van	83	3716	$0.11 \cdot 10^6$	$0.08 \cdot 10^6$
Identical	14733	614199	$13.79 \cdot 10^6$	$15.03 \cdot 10^6$
Common Hun	8598	658532	$15.75 \cdot 10^6$	$16.97 \cdot 10^6$
Common Van	8598	621816	$15.65 \cdot 10^6$	$16.94 \cdot 10^6$
Max-size	8598	658583	$15.75 \cdot 10^6$	$16.97 \cdot 10^6$
Max-acc	8072	575749	$14.19 \cdot 10^6$	$15.67 \cdot 10^6$

Table 3.4: Results of processing the four JRC3 corpora pairs: number and sizes of the documents in the common parts of the corpora, documents present in just one corpus and the resulting max-size and max-accuracy combinations

	JRC3 en-et		JRC3 en-lv		JRC3 et-lv		JRC3 de-en	
	Hun	Van	Hun	Van	Hun	Van	Hun	Van
∅	21.5%	15.4%	20.9%	15.1%	11.1%	7.2%	11.9%	7.8%
0-1	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
1-0	1.4%	0.0%	1.4%	0.0%	0.2%	0.0%	0.6%	0.0%
1-1	76.9%	84.5%	77.6%	84.8%	88.3%	92.4%	86.7%	91.8%
1-2	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%
2-1	0.2%	0.1%	0.1%	0.1%	0.5%	0.2%	0.8%	0.4%
2-2	–	–	0.0%	0.0%	–	–	0.0%	0.0%

Table 3.5: Frequency of the match types between sentence pairs of the four JRC3 corpora pairs; given as proportion of sentence from a match type in a corpus. The ∅ stands for mismatching (i.e. probably erroneously aligned) sentence pairs

corpora has been found. Restricting the evaluation to the matched parts of the corpora (i.e. omitting the zero-to-one matches, since these are not matched and therefore the portion of potential alignment errors within them is unknown), we obtain the following percentages of mismatched sentences: 10.5% in the UT and 12.1% in the JRC2 corpus.

Moving on to the JRC3 pairs, the first interesting observation is the difference between the documents included only in the Vanilla or HunAlign versions. It can be seen in table 3.4 that while the HunAlign versions of all the four pairs include only three to five documents that are not included in the Vanilla versions, the total numbers of words and sentence pairs in these documents are much higher than their counterparts in the Vanilla versions. After inspecting the files, unique to either alignment versions, we found that the HunAlign files are enormously large (50 – 60 thousand sentence pairs) while all the Vanilla files are very small (up to 100 sentence pairs). Although this means that the HunAlign aligner is much more confident with longer documents, nothing can be said about the resulting quality without further inspection of these aligned documents.

In addition the total sizes of the common parts of the HunAlign versions are bigger than the same document sets of Vanilla versions. These two facts might indicate that in the HunAlign version documents and sentences were more confidently included into the corpus than in the Vanilla versions, which is quite the opposite of the UT+JRC2 case.

Roughly half of the overlapping parts of all four pairs consists of identical document pairs. In addition, the average number of sentence pairs and words per document is noticeably lower in case of identically matching documents than the other matching documents. This further indicates that both aligners are more precise in case of shorter documents.

It seems that all four language pairs include a large portion of mismatching

sentence pairs, especially the English-Estonian and English-Latvian pairs (table 3.5). However, this is a result of treating the identically matching document pairs separately – if these had been included, the proportions of the mismatching sentence pairs would have been at least twice as small. Still, the max-accuracy combinations are considerably smaller than both the HunAlign and Vanilla common parts. On the other hand, the max-size combinations are practically of the same size as the bigger HunAlign common parts (with only 100-150 extra sentence pairs).

Finally, almost no many-to-one alignments were found between the HunAlign and Vanilla JRC3 versions of all four pairs. A majority of the few existing many-to-one alignments are segmented better in the HunAlign version.

To conclude, analysis of the overlapping parts of the corpora has proved useful in getting additional information about the corpora and our method and the CorporAl tool are capable of successfully retrieving a portion of it.

3.4.2 Influence on Machine Translation

In (Fishel et al., 2007) we present experiments on translation from Estonian into English using the UT and JRC2 corpora. Using both corpora is solved by concatenating them prior to translation model training.

The concatenation tactic employed in that work results in skewing the original distribution of the data: sentence pairs, present in both parts of the overlap will be overrepresented since their relative frequency will increase in comparison to the sentence pairs outside the overlap or the ones that are present in only one corpus.

The correct baseline method of combining overlapping corpora is taking the non-overlapping parts of both corpora and the overlapping part from just one of them. In our case instead of giving preference to either part of UT+JRC2 or JRC3 pairs we used both versions of the baseline; thus every experiment includes two baselines: in case of UT+JRC2 an UT-based and a JRC-based one, and in case of the JRC3 pairs – a HunAlign-based and a Vanilla-based one.

Our goal here is to compare the two baselines to the max-size and max-accuracy combinations done with the CorporAl tool. Naturally these two corpora versions also include the non-overlapping parts of both corpora from each pair.

In order for the development and testing sets to be the same for all experiments these were picked randomly from the two baselines prior to processing the common parts. Later the same files were removed from all four training set versions to ensure independent data sets.

The resulting scores of the UT+JRC2 translation systems are presented in figure 3.3 and the scores of the JRC3 pairs – in figures 3.4 and 3.5. The score differences and the significance testing results are presented in table 3.6 for comparisons of the max-size versions to the baselines and in table 3.7 – for comparisons of the max-accuracy versions.

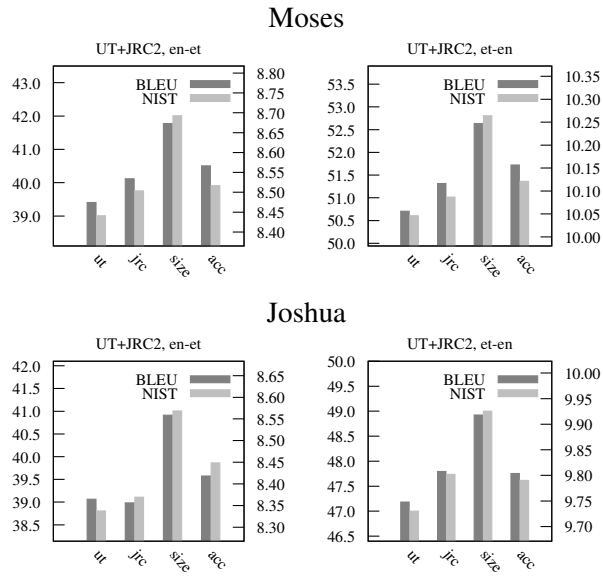


Figure 3.3: Results of the machine translation experiments for the UT and JRC2 corpora. The BLEU score scale is on the left, and the NIST score scale – on the right

In case of both decoders and both translation directions of the UT+JRC2 pair a clear pattern is visible: although in some cases the JRC-based results are better than the UT-based results, in general the max-size results noticeably exceed the max-accuracy and both baseline results, and max-accuracy results are slightly better than the UT-based baseline and practically the same as the JRC-based baseline. The JRC3 pairs on the other hand do not exhibit any clear pattern; looking at the scales of the differences we can conclude that there is no systematic difference between systems based on all four corpora. Both conclusions are invariant to different translation models and are further confirmed by the significance test results in tables 3.6 and 3.7.

The two opposite conclusions for UT+JRC2 and JRC3 experiments can be explained by two main factors. First of all, the UT+JRC2 max-size combinations are considerably bigger than the other parts while the JRC3 combinations are almost of the same size as the base corpora. Secondly, based on our analysis of the results of processing the corpora we can claim that the UT and JRC2 corpora are rather heterogeneous while the JRC3 pairs – rather homogeneous.

An interesting result is the max-accuracy combinations causing roughly the same scores as the baselines. Similarly, despite Kaalep and Veskis (2007) having showed the Vanilla-based alignments to be of worse quality than HunAlign-based alignments, the HunAlign results are either roughly the same as Vanilla or just

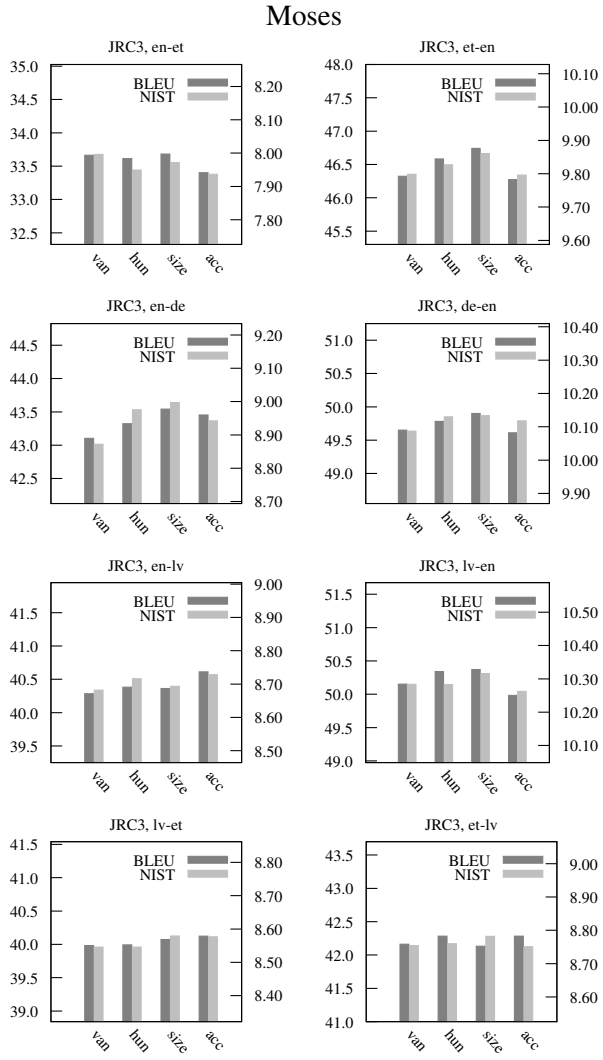


Figure 3.4: Results of the machine translation experiments for the four JRC3 corpora pairs with Moses. The BLEU score scale is on the left, and the NIST score scale – on the right

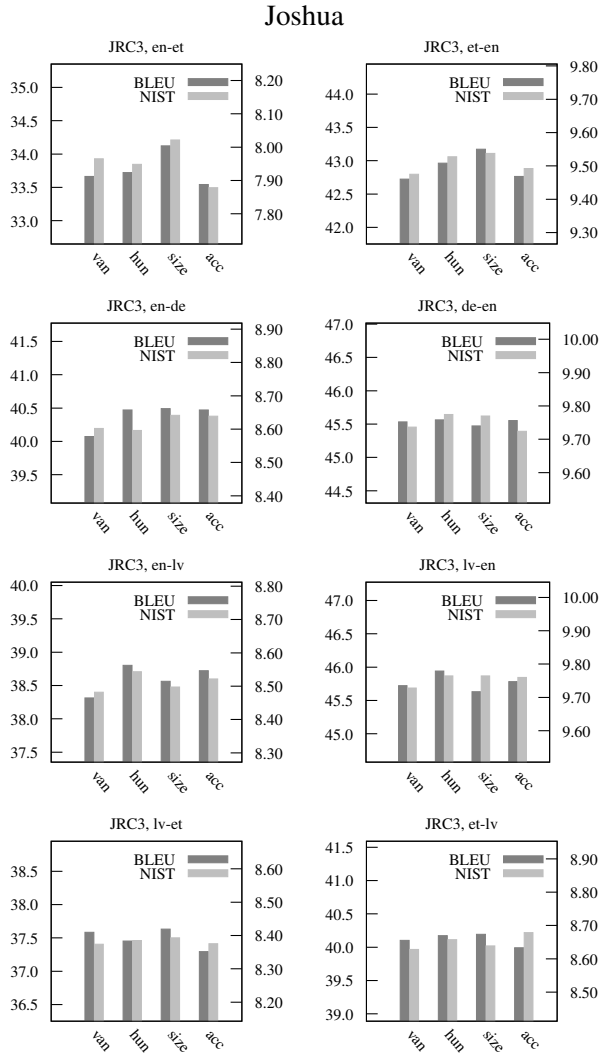


Figure 3.5: Results of the machine translation experiments for the four JRC3 corpora pairs with Joshua. The BLEU score scale is on the left, and the NIST score scale – on the right

Baselines compared to the max-size combination

	Moses		Joshua	
	BLEU	NIST	BLEU	NIST
UT:				
et-en	1.93 (0.000)	0.2178 (0.000)	1.74 (0.000)	0.1954 (0.000)
en-et	2.37 (0.000)	0.2515 (0.000)	1.85 (0.000)	0.2309 (0.000)
Van:				
et-en	0.42 (0.012)	0.0619 (0.005)	0.45 (0.010)	0.0623 (0.005)
en-et	0.02 (0.374)	-0.0252 (0.140)	0.46 (0.008)	0.0567 (0.017)
lv-en	0.22 (0.064)	0.0321 (0.060)	-0.09 (0.246)	0.0363 (0.063)
en-lv	0.08 (0.215)	0.0113 (0.207)	0.25 (0.071)	0.0157 (0.191)
et-lv	-0.03 (0.310)	0.0276 (0.069)	0.09 (0.183)	0.0104 (0.204)
lv-et	0.09 (0.160)	0.0336 (0.049)	0.05 (0.268)	0.0196 (0.110)
de-en	0.25 (0.043)	0.0468 (0.011)	-0.06 (0.247)	0.0331 (0.062)
en-de	0.44 (0.001)	0.1253 (0.000)	0.42 (0.009)	0.0395 (0.036)

Baselines compared to the max-size combination

	Moses		Joshua	
	BLEU	NIST	BLEU	NIST
JRC:				
et-en	1.32 (0.000)	0.1769 (0.000)	1.13 (0.000)	0.1235 (0.000)
en-et	1.66 (0.000)	0.1888 (0.000)	1.93 (0.000)	0.1988 (0.000)
Hun:				
et-en	0.16 (0.110)	0.0338 (0.044)	0.21 (0.107)	0.0100 (0.241)
en-et	0.07 (0.223)	0.0217 (0.144)	0.40 (0.017)	0.0731 (0.003)
lv-en	0.03 (0.316)	0.0326 (0.052)	-0.31 (0.050)	0.0002 (0.418)
en-lv	-0.02 (0.340)	-0.0225 (0.125)	-0.24 (0.078)	-0.0456 (0.030)
et-lv	-0.15 (0.086)	0.0222 (0.069)	0.02 (0.351)	-0.0189 (0.126)
lv-et	0.08 (0.182)	0.0333 (0.050)	0.18 (0.094)	0.0084 (0.219)
de-en	0.12 (0.136)	0.0036 (0.310)	-0.09 (0.197)	-0.0048 (0.307)
en-de	0.22 (0.059)	0.0216 (0.135)	0.02 (0.350)	0.0454 (0.021)

Table 3.6: Score difference and their p-values, resulting from the significance tests between the baselines and the max-size combination-based translation. Insignificant differences are marked with gray, significantly better scores of the max-size combination – with green and significantly worse scores – with blue

Baselines compared to the max-accuracy combination

	Moses		Joshua	
	BLEU	NIST	BLEU	NIST
UT:				
et-en	1.02 (0.000)	0.0754 (0.001)	0.57 (0.006)	0.0599 (0.024)
en-et	1.10 (0.000)	0.0757 (0.009)	0.51 (0.031)	0.1112 (0.000)
Van:				
et-en	-0.05 (0.289)	-0.0026 (0.365)	0.04 (0.306)	0.0169 (0.171)
en-et	-0.26 (0.064)	-0.0600 (0.015)	-0.12 (0.190)	-0.0869 (0.001)
lv-en	-0.17 (0.114)	-0.0211 (0.126)	0.06 (0.257)	0.0318 (0.071)
en-lv	0.33 (0.013)	0.0459 (0.022)	0.41 (0.007)	0.0395 (0.055)
et-lv	0.12 (0.143)	-0.0033 (0.340)	-0.11 (0.169)	0.0505 (0.013)
lv-et	0.14 (0.124)	0.0308 (0.066)	-0.29 (0.015)	0.0016 (0.360)
de-en	-0.04 (0.299)	0.0311 (0.060)	0.02 (0.384)	-0.0126 (0.219)
en-de	0.35 (0.025)	0.0701 (0.004)	0.40 (0.011)	0.0372 (0.059)

Baselines compared to the max-accuracy combination

	Moses		Joshua	
	BLEU	NIST	BLEU	NIST
JRC:				
et-en	0.41 (0.024)	0.0345 (0.091)	-0.04 (0.333)	-0.0120 (0.241)
en-et	0.39 (0.032)	0.0130 (0.248)	0.59 (0.011)	0.0791 (0.009)
Hun:				
et-en	-0.31 (0.027)	-0.0307 (0.070)	-0.20 (0.128)	-0.0354 (0.070)
en-et	-0.21 (0.093)	-0.0131 (0.210)	-0.18 (0.147)	-0.0705 (0.008)
lv-en	-0.36 (0.014)	-0.0206 (0.127)	-0.16 (0.150)	-0.0043 (0.321)
en-lv	0.23 (0.088)	0.0121 (0.222)	-0.08 (0.225)	-0.0218 (0.146)
et-lv	0.00 (0.414)	-0.0087 (0.244)	-0.18 (0.078)	0.0212 (0.119)
lv-et	0.13 (0.143)	0.0305 (0.059)	-0.16 (0.117)	-0.0096 (0.234)
de-en	-0.17 (0.106)	-0.0121 (0.185)	-0.01 (0.399)	-0.0505 (0.009)
en-de	0.13 (0.133)	-0.0336 (0.066)	0.00 (0.423)	0.0431 (0.027)

Table 3.7: Score difference and their p-values, resulting from the significance tests between the baselines and the max-accuracy combination-based translation. Insignificant differences are marked with gray, significantly better scores of the max-accuracy combination – with green and significantly worse scores – with blue

slightly higher (not more than 0.4-0.5 BLEU and 0.05-0.07 NIST points).

This phenomenon can be attributed to the frequency-based re-estimation of parameters in statistical machine translation: for instance, the phrase table weights are estimated based on the relative frequency of aligned word pairs, which in its turn is based on how often sentence pairs containing these word pairs are aligned together. As a result, noise in the training data is automatically discarded, being irregular and thus its instances having low frequencies. In other words, unless the sentence or word alignment errors are systematic, they are automatically ignored, which results in the translation model learning methods being robust to noise in the data and having low sensitivity to sentence alignment quality.

In practice this means that the translation models trained on max-size combinations most probably successfully use the sub-set of the corpus that is present in the max-accuracy combination, plus other sentence pairs that were excluded from the latter but are nevertheless well aligned pairs. The main difference between the models trained on max-size and max-accuracy combinations is that the former model had more material to learn from, hence better output quality. The same explanation applies to comparisons of max-size combination with the baselines.

To conclude, translation quality seems to be influenced greatly by the size of the training corpus, and not so much by its quality, as the learning models are rather robust to alignment errors. However, our method of combining overlapping corpora leads to higher translation scores when applied to smaller and more heterogeneous corpora.

3.5 Future Work

The main direction of development is extending the tool to support monolingual corpora with annotation, in addition to parallel corpora. If the two overlapping corpora are augmented with the same annotation then both the text and the annotation can be compared, just like the two language parts of parallel corpora.

Alternatively, if the annotations are different, the aim is slightly different too – instead of matching the alignments of text and annotation, just the text can be aligned. This just requires making the alignment of the second part (be it the second part of a parallel corpus or annotation of a monolingual corpus) and the matching of the alignments optional.

As a result the tool would allow to produce a text corpus with both annotations, regardless of differences of the texts. The latter might occur, for example, if a corpus is analyzed with a morphological analyzer and also parsed syntactically, and the resulting texts differ in segmentation, or in case of human processing – also in spelling and omitted/included units.

In addition it could be used to process multilingual parallel corpora where alignments are present for only some of the language pairs. For example in Eu-

roparl (Koehn, 2005) only alignments to English are released. In that case it would be possible to use English as a pivot language and generate, for instance, an aligned Swedish-Greek parallel corpus, as an alternative to re-applying the alignment software.

CHAPTER 4

LINGUISTICALLY MOTIVATED UNSUPERVISED SEGMENTATION

This chapter presents our second contribution of this dissertation. The presented material is partially published in (Kirik and Fishel, 2008) and (Fishel and Kirik, 2010).[1]

Here we focus on the domain of altering the training data in such a way that the task of the statistical learning system would be easier. This includes preprocessing the data before applying the learning models to "translate" it into the altered form and/or postprocessing the output of the trained system to "translate" the altered form back into the original one.

One of the approaches from this domain used in machine translation addresses data sparsity that arises from the morphological complexity of the language(s) of the used corpora. It consists of segmenting highly inflecting word forms into morphs or other segments prior to training the translation models and joining the segments back into word forms after translation. Thus the assumption made here is that the word segments can be treated as independent of each other.

Intuitively this approach should work best with agglutinative languages: languages where words are formed via morpheme concatenation without (or with little) altering them (Shibatani and Bynon, 1999). This includes most Uralic languages (Estonian, Finnish, Hungarian, etc.), Turkish, Korean and others. The approach has also been applied to heavily compounding languages like German and other morphologically rich languages like Arabic and Czech.

Two most common varieties of this approach are linguistics-driven and data-driven. The first approach uses morphological analysis, while in the second segmentations are derived from unsegmented corpora to optimize some predefined objective; it is commonly called unsupervised segmentation. An overview of re-

[1] Our contribution includes the original ideas and their design, shared experimental evaluation with the other author, writing the first paper alone and the second paper – together with the other author.

lated work is presented in section 4.1.

In section 4.2 we introduce a method that is based on a combination of both approach variations, which is the central topic of this chapter. Using unsupervised segmentation and classification of the segments into stems, prefixes and suffixes it applies the principles of the morphology-based approach to obtain the final segmentations by selectively re-grouping some of the segments.

The main research question of this chapter is whether our method yields better translation scores in comparison to the word-based baseline translation system. Additionally we compare our method to the segmentation baseline, i.e. without re-grouping.

Experimental evaluation of the method is described in section 4.3. Initial results of using it for machine translation between Estonian and English, published in (Kirik and Fishel, 2008) and (Fishel and Kirik, 2010), come to different conclusions – depending on the used corpora the method either improves the scores of the translation systems or decreases them. We discuss the possible reasons behind the variable success of the introduced method and present results of additional experiments, where we apply it to a new corpus of Estonian-English and two other language pairs: Finnish-English and German-English.

In regards to the two translation approaches, used in the experiments of this dissertation, hierarchical phrase-based translation has a potential additional advantage over phrase-based translation, when used with segmentation methods. Hierarchical phrase-based translation models explicitly the internal structure between the tokens of the sentences, regardless of whether the tokens are actual word forms or their segments. This could alleviate the segment independence assumption, by modelling the word structure jointly with the sentence structure.

The chapter is concluded with a discussion of possible future work in section 4.4.

4.1 Background and Related Work

Here we review several approaches to morphological segmentation for machine translation. All of these are restricted to text-to-text translation, like the current dissertation. The reason is that tree-to-tree approaches would require syntactic parsers that would be able to parse sequences of morphs or other segments instead of sequences of words and we know of no experiments on using segmentation in text-to-tree or tree-to-text approaches.

Most of the cited works report moderate to significant improvements as a result of their technique. A recurring conclusion in many works, e.g. (Habash and Sadat, 2006; Sereewattana, 2003), is that the positive effect of segmentation on translation quality is much more visible on small corpora; in case of larger corpora the effect is smaller (Stymne and Holmqvist, 2008; Carpuat, 2009).

The only method reporting negative results is unsupervised segmentation of Morfessor (Creutz and Lagus, 2005), which decreases translation scores but still improves the ratio of untranslated words and partially translated sentences (Virpioja et al., 2007; Kurimo et al., 2010). Still, when used in combination with a word-based model, it leads to improvement in comparison with the baseline (de Gispert et al., 2009; Kurimo et al., 2010).

4.1.1 Linguistic Segmentation

Depending on the language that is segmented as well as the other language in the translation direction, different levels of segmentations have been utilized.

One of the approaches is segmenting words forms into separate morphemes – for instance in translation into English from Turkish (Oflazer and Durgar El-Kahlout, 2007) and Arabic (Lee, 2004; Habash and Sadat, 2006; Zollmann et al., 2006). This includes separating clitics and all affixes from the word stems and from each other. The same approach has been tested on English-to-Arabic translation to segment the output (Badr et al., 2008); they also test grouping all prefixes and all suffixes together, thus segmenting each word into at most three parts.

In some cases, when the morphology of the language(s) is less productive, simpler segmentation schemes have been tested. For example Nießen and Ney (2004) separate German verb prefixes from the verbs to translate from German into English. Popovic and Ney (2004) split the word forms into either stems and suffixes or, alternatively, lemmas and tags with morphological information in translation from Spanish, Catalan and Serbian into English.

Explicitly separating the compound parts has been extensively tested for translation between English and German – for example in (Stymne et al., 2008) and (Popović et al., 2006). When translating from German the two works use the same approach. To translate into German Stymne et al. (2008) later join the generated separate compound parts back into compound words while Popović et al. (2006) merge English words, based on their PoS tags, prior to translation.

Another alternative is replacing every word with its lemma and tag with morphological information. Unlike segmentation, this usually increases the vocabulary size, but reduces ambiguity of the word forms. This technique has been tested, e.g. for Czech-to-English (Bojar et al., 2006), Turkish-to-English (Oflazer and Durgar El-Kahlout, 2007), German-to-English (Nießen and Ney, 2004) and French-to-English (Carpuat, 2009) translation.

4.1.2 Unsupervised Segmentation

In contrast to letting linguistic theory dictate the segmentations, in the unsupervised data-driven approach it is derived automatically based on the corpus at hand. The exact way to segment word forms is learned to optimize a predefined

objective, such as the compactness of the learned model or the probability of the segmented corpus. Extensive overviews of unsupervised segmentation algorithms are given in (Hammarström, 2009) and (Creutz and Lagus, 2007).

Similarly to linguistic segmentation different levels of segmentation are employed here; in this case these are typically included into the segmentation algorithm.

The deepest level of segmentation is segmenting word forms without constraining the number of morphs. In context of machine translation the most popular method is Morfessor (Creutz and Lagus, 2005). It has been used for translation from German and Czech into English (Virpioja et al., 2010), Finnish to English (de Gispert et al., 2009), Farsi to English (Kathol and Zheng, 2008) and translation between Danish, Swedish and Finnish (Virpioja et al., 2007). While the latter segment both the source and the target words, the other works only segment the source, i.e. the input of the translation system.

An event called Morpho Challenge has taken place since 2005 where contestants submit algorithms of unsupervised segmentation, which are evaluated in comparison to linguistic segmentation and also by their effect on information retrieval and in 2009 and 2010 – machine translation from Finnish and German into English. The results of the latest Morpho Challenge are summarized in (Kurimo et al., 2010); so far Morfessor resulted in the highest BLEU scores in all cases.

Sereewattana (2003) design an algorithm similar to Morfessor to segment word forms into stems and suffixes and apply it to translation from German and French into English. Karageorgakis et al. (2005) use Linguistica, a free implementation of the algorithm described in (Goldsmith, 2001), to stem both the English input and Greek output; this stem-to-stem translation model is used in conjunction with a baseline word form-based model.

A separate line of work is automatic segmentation of compounds. A popular approach introduced in (Koehn and Knight, 2003) segments a compound only if its frequency is lower than the geometric mean of the frequencies of the parts in the segmentation. In several cases a compound is frequent enough in the corpus for the translation model to learn it without segmentation, which is the main advantage over linguistics-based compound segmentation. Koehn and Knight (2003) use the method in German-to-English translation to segment the German compounds. The method is slightly extended and applied to Swedish-to-English translation in (Stymne and Holmqvist, 2008) and to German-to-English and English-to-German translation in (Popović et al., 2006).

4.1.3 Bilingual Optimization of Segmentation

Both linguistic and unsupervised segmentation are essentially monolingual – i.e. are driven by either morphological analysis of the segmented language or the corpus of the same language. Several works use the bilingual context of applying

these methods to further optimize the segmentations for machine translation.

In linguistic segmentation a popular approach is grouping some morphemes back together, based on manual analysis of the correspondence of the morphs in the language pair. This has been tested for translation into English from Arabic (Habash and Sadat, 2006; Lee, 2004) and Turkish (Oflazer and Durgar El-Kahlout, 2007), where the re-grouping is meant to make the input language segmentation more similar to English, i.e. the output language.

In (Zollmann et al., 2006) the bilingual context is considered automatically, using a dictionary of segmentations and their corresponding English translations. To disambiguate segmentations, the one with most matches in the English target sentence is always chosen.

Finally, to further optimize the segmentation in some cases part of the morphological information is intentionally discarded. For example in Arabic-to-English translation experiments in (Zollmann et al., 2006) some features of Arabic not present in English (such as gender markers) are removed from the segmented Arabic morphemes. In Czech-to-English translation in (Bojar et al., 2006) one of the approaches is simply lemmatizing the Czech input, thus removing all of the morphological info altogether.

Unlike linguistic segmentation, in unsupervised segmentation manual analysis of the alignments can only help decide whether to split the word forms into stems and suffixes, or compounds, or an unrestricted number of morphemes. Truly bilingual unsupervised learning of morphological word segmentation has only been introduced recently and applied to Korean-to-English (Chung and Gildea, 2009), Arabic-to-English (Nguyen et al., 2010) and Turkish-to-English (Mermer and Akın, 2010) translation; all of these approaches support segmenting only one of the languages. In addition, Snyder and Barzilay (2008) introduce a model, supporting bilingual segmentation of both involved languages and apply it to Arabic and Hebrew, paired with each other, English or Aramaic; the model is not applied to translation and is evaluated in comparison to linguistic segmentation only.

Bilingual learning is extensively applied to other tasks, similar to morphological segmentation. These include word boundary detection in Chinese-to-English translation (Chung and Gildea, 2009; Ma and Way, 2009; Huang et al., 2008; Nguyen et al., 2010) and bilingual chunking of Chinese (Wang et al., 2002; Liu et al., 2004). By the nature of the tasks we can guess that these models could be applied to morphological segmentation, but since the aims are different, it is hard to know whether they would be successful, or whether specialized models are required.

4.1.4 Alternative approaches

A lot of other methods other than segmentation have been proposed to handle rich morphology in machine translation. One of these is backoff models, where

the word forms are augmented with lemmas and sometimes more general categories. In case a word form is unknown, the translation model attempts to translate its lemma in the basic form. This approach is tested on German-to-English (Nießen and Ney, 2004) and French-to-English (Carpuat, 2009) translation.

A method, somewhat similar to discarding some morphological information, is clustering together some of the words and later making no distinction between them. For instance in (Talbot and Osborne, 2006) clustering and translation models are learned with a joint learning algorithm.

Finally a completely opposite approach to segmenting or removing some of the information from the morphologically rich language is artificially augmenting a morphologically poor language with the information, required to correctly generate a morphologically rich output. This approach has been tested for translation from English into Spanish and Catalan in (Ueffing and Ney, 2003) as well as into Arabic and Russian in (Minkov et al., 2007).

4.2 Method Description

Here we describe our own contribution, which is a method of segmentation, based on both unsupervised and linguistic segmentation. It is used for machine translation in the standard manner: the training corpus is segmented prior to training the translation models; if the source language is segmented, then the input to the decoder is segmented, prior to translation, and if the output language is segmented, the segments, that the output of the decoder consists of, have to be grouped back into word forms.

Our segmentation method consists of first applying unsupervised segmentation with segment categorization and then re-group the segments using the obtained categories according to predefined re-grouping schemes. The schemes are what is motivated by linguistic theory and are meant to model various phenomena of natural language morphology. We use the Morfessor tool (Creutz and Lagus, 2005) for unsupervised segmentation and segment categorization.

In other words the proposed method replaces linguistic morphological analysis with Morfessor, using its output in a very similar way. The assumption here is that, given how state-of-the-art methods of statistical machine translation are very much heuristics-based and very weakly motivated by linguistic theory, linguistic morphological analysis is not necessarily the most optimal way of segmentation that will be used for translation.

The main advantage of the method is its independence of linguistic tools. Unlike linguistic segmentation, which requires morphological analysis for the language of interest, Morfessor can be trained with raw text corpora, just like statistical machine translation. It therefore does not increase the requirements for building the enhanced translation models.

An additional motivation for using Morfessor as the base of the method is that it implements categorized segmentation unlike other commonly used implementations of unsupervised segmentation (e.g. Linguistica (Goldsmith, 2001)). Also, while many works use unsupervised segmentation for various purposes besides machine translation, we do not know a single example of the segment categorization being used.

We start by describing the functionality and principles of Morfessor. Then we introduce the re-grouping schemes based on its output.

4.2.1 Morfessor

Morfessor was created as part of the Morpho project of the department of computer science and engineering at the Helsinki University of Technology. It is openly accessible from the homepage of the project.[2]

The first developed algorithm, Morfessor-Baseline was introduced in (Creutz and Lagus, 2002) and only supported word segmentation. It is based on the minimum description length (MDL) principle, which states that more compact models describe the data in a better way; see e.g. (Chen, 1996) for elaboration on this topic with several examples. The baseline model is still used as the starting point of the newer version, which supports categorization – Morfessor Categories-MAP introduced in (Creutz and Lagus, 2005) and later described in greater detail in (Creutz and Lagus, 2007).

Morfessor-Baseline

The search for the optimal model $\mathcal{M}$ for the data $\mathcal{D}$ is here guided by the maximum a posteriori (MAP) estimation principle:[3]

$$\begin{aligned}\widehat{\mathcal{M}} &= \operatorname*{argmax}_{\mathcal{M}} p(\mathcal{M}|\mathcal{D}) \\ &= \operatorname*{argmax}_{\mathcal{M}} p(\mathcal{D}|\mathcal{M}) \cdot p(\mathcal{M}) \\ &= \operatorname*{argmax}_{\mathcal{M}} \log p(\mathcal{D}|\mathcal{M}) + \log p(\mathcal{M}).^{4} \end{aligned} \tag{4.1}$$

As suggested by Chen (1996), the model prior corresponding to the MDL principle is defined as

$$p(\mathcal{M}) = 2^{-\operatorname{len}(\mathcal{M})}, \tag{4.2}$$

[2] `http://www.cis.hut.fi/projects/morpho/`.

[3] Originally in (Creutz and Lagus, 2002) the baseline is described purely in the MDL framework, but the authors note the equivalence between MDL and MAP, shown e.g. by Chen (1996); here we describe both the baseline and the Categories-MAP in the same framework.

[4] equivalent to $\operatorname{argmin}_{\mathcal{M}} -\log p(\mathcal{D}|\mathcal{M}) - \log p(\mathcal{M})$, showing that the objective is to minimize the total information content (measured with $-\log_2 p(x)$).

where $\mathrm{len}(\mathcal{M})$ is the length of the model in bits. The model consists of a lexicon of segments, or morphs: $(\{m_i\}),\ i \in [1..I]$, where I is the lexicon size. The length can be therefore computed using the character length of each morph in the lexicon:

$$\mathrm{len}(\mathcal{M}) \quad = \quad \sum_{i=1}^{I} b \cdot \mathrm{len}(m_i). \tag{4.3}$$

Here b stands for the number of bits, necessary to encode a character.

Finally the likelihood of the data needs to be defined. Given a specific model and having the data optimally segmented with it the segments are treated as independent and therefore having W words in the text corpus (i.e. the data) and each word being segmented into n_j segments we have

$$p(\mathcal{D}|\mathcal{M}) \quad = \quad \prod_{j=1}^{W} \prod_{k=1}^{n_j} p(m_{jk}), \tag{4.4}$$

where the probabilities $p(m)$ are maximum likelihood estimates and are equal to the relative frequency of their morphs in the corpus.

Finally substituting (4.3) into (4.2) and (4.4) and (4.2) into (4.1) we get

$$\sum_{j=1}^{W} \sum_{k=1}^{n_j} \log p(m_{jk}) - \sum_{i=1}^{I} b \cdot \mathrm{len}(m_i).$$

The aim is thus to maximize data likelihood and minimize the total description length of the model.

The employed search procedure is online learning. The corpus words are processed one at a time; when processing a word every possible segmentation of the word into two morphs is evaluated – not segmenting it is also included in the evaluation. The segmentation with the biggest decrease of the total cost is selected.

Unless not segmenting the word was selected, the segmentation process is repeated recursively for both new segments of the word. As a result words get segmented into an unrestricted number of morphs.

If a word has been seen and added as a whole to the lexicon, the search would never select segmenting it when it is seen again since this would increase the lexicon size. In order for this effect not to cause getting stuck in local optima every time a word or segment is processed it is temporarily removed from the lexicon until its processing is finished.

Finally, words encountered in the beginning might have suboptimal segmentation. This is remedied with occasionally occurring "dreaming" sessions: before continuing to process new corpus words the words from the lexicon are picked

and re-segmented the same way as new corpus words. This further optimizes the lexicon and compresses its length.

Applying the models to segment the text is implemented as exhaustive search through all segmentations and selecting the most probable one.

Morfessor Categories-MAP

The Categories-MAP version of Morfessor adds categories to the morphs: every morph can either be a prefix (`PRE`), suffix (`SUF`), stem (`STM`) or a non-morpheme (`NON`). The affix category morphs typically carry syntactic functions in the text and the stems – semantic functions; this assumption is used to define the likeliness of all morphs to be of certain categories. The last, non-morpheme, category is used for the cases when the morph is not likely to be any of the first three categories and is considered "as a mere sound pattern without a syntactic or a semantic function" (Creutz and Lagus, 2005).

The data likelihood is in this case defined as a first-order hidden Markov model with the categories C_{jk} as the hidden variables:

$$p(\mathcal{D}|\mathcal{M}) \quad = \quad \prod_{j=1}^{W} p(C_{j1}|C_{j0}) \prod_{k=1}^{n_j} p(m_{jk}|C_{jk}) \cdot p(C_{j(k+1)}|C_{jk}).$$

The word-initial and -final categories (C_{j0} and $C_{j(n_j+1)}$) are set to a special "word boundary" value.

The probability mass of the transition probability distributions $p(C_{j(k+1)}|C_{jk})$ is distributed between so-called *legal* pairs of categories in order to impose a certain structure on them. The structure only affects the first three categories (`PRE`, `SUF` and `STM`) and can be described with the following regular expression:

```
(PRE* STM SUF*)+
```

Thus a word consists of at least one stem, which can have any number of prefixes and suffixes around it. A non-morpheme can occur at any position.

Unlike Morfessor baseline, here the MDL principle is not used and the model prior is defined through the priors of the lexicon morph properties. The latter includes, besides others, the right- and left-perplexity of the morph – i.e. the predictability of the next or previous morph, respectively, based on the current morph. It is assumed that a morph is likely to be a prefix if its next morph is hardly predictable, i.e. its right-perplexity is low; in the same way a likely suffix morph will have a low left-perplexity.

Stems are classified mainly based on the length of the morph. The motivation is that while there is always a limited number of functional morphs in the language, the number of different stems is much higher, which means that most stems will not be encoded with a small number of characters.

Using the length, right- and left-perplexity (right-ppl and left-ppl) of the morph the measures of stem-, prefix- and suffix-likeness are defined:

$$\begin{aligned}
\text{prefix-like}(m) &= \frac{1}{1+e^{-a\cdot(\text{right-ppl}(m)-b)}}, \\
\text{suffix-like}(m) &= \frac{1}{1+e^{-a\cdot(\text{left-ppl}(m)-b)}}, \\
\text{stem-like}(m) &= \frac{1}{1+e^{-c\cdot(\text{len}(m)-d)}}.
\end{aligned}$$

Finally, defining the probability of the non-morpheme category as

$$\begin{aligned}
p(\texttt{NON}|m) &= (1-\text{prefix-like}(m))\cdot(1-\text{suffix-like}(m))\times \\
&\quad \times(1-\text{stem-like}(m))
\end{aligned}$$

and distributing the remaining probability mass between stems, prefixes and suffixes:

$$p(\texttt{PRE}|m) = \frac{\text{prefix-like}(m)\ \cdot\ (1-p(\texttt{NON}|m))}{\text{prefix-like}(m)+\text{suffix-like}(m)+\text{stem-like}(m)}$$

(and likewise for stems and suffixes) we arrive at the conditional PDF of the morph categories.

More detailed description and the exact definition of the model prior for Morfessor Categories-MAP is given in (Creutz and Lagus, 2007), we now describe the utilized model search and applying procedures.

The search is initialized with training and applying Morfessor baseline. Having the corpus segmented as a result, the Viterbi algorithm (Viterbi, 1967) is used to tag each segment and the probabilities $p(m|C)$ and $(p(C_{j(k+1)}|C_{jk})$ are estimated from the corpus.

Following initialization two iterations of learning are performed. Each iteration includes an enhanced "dreaming" session, which in addition to testing re-segmentation of the lexicon morphs also attempts to re-group them morphs, evaluating the impact of each re-grouping and keeping or rejecting it. After the dreaming session the corpus is re-segmented and the probabilities are re-estimated until convergence.

New text tagging consists of segmenting the words and then tagging the resulting morphs with categories again using the Viterbi algorithm. During recursive segmentation of the morphs the structure is kept in order to later re-group the morphs, tagged with the non-morpheme category, with their sibling.

Functionality

Morfessor the implementation consists of two functional modules. One implements learning segmentation and categorization models from text corpora. The

Word form	Analysis
упячка	у/PRE+пяч/STM+ка/SUF
(upyachka, a russian internet-meme)	
lennupiletitega	lenn/STM+u/SUF+pileti/STM+te/SUF+ga/SUF
("with airplane tickets" in Estonian)	
skyscraper	sky/STM+scrap/STM+er/SUF

Table 4.1: Examples of the output of Morfessor and our front-end to it. PRE stands for prefix, SUF – for suffix and STM – for stem. The second and third examples are compound nouns, which is shown by the two stems in the analysis. All three examples are constructed manually

other implements applying the learned models to new text data. Since all word forms are assumed to be independent, both modules expect lists of word forms and their frequencies as input and the application module produces the same kind of list as its output, which means that word order of the original text is lost.

To restore the word order and apply the segmentations to the words in the original texts we built a front-end to Morfessor, which takes usual text corpora and transforms it automatically into the form, expected by Morfessor. Similarly it takes Morfessor's output and applies each segmentation to every word form – this is attainable via simple look-up, again due to Morfessor's assuming words not to depend on their context and thus having the same segmentation regardless of the context. Examples of the final output of our front-end are given in table 4.1.

4.2.2 Re-grouping Schemes

The final step after obtaining the segmentation and segment categories from Morfessor is using them to re-group the segments. We propose some ways to preform the re-grouping, which are linguistically motivated; this research is restricted to modeling compounding and stems and affixes.

Compounding

As mentioned previously, Morfessor separates stems and affixes, whereas stems have dominantly semantic and affixes – syntactic functions. By grouping the affixes with their adjacent stems and separating these groups from each other we can hopefully model the compounding process of natural language with applying it to segmentation. We assume that the prefixes belong in one group with the stem

Taking the examples from table 4.1, the stem *scrap* and suffix *er* are grouped

together for the English word *skyscraper*; the result is

sky+ +scraper

Similarly, in case of the Estonian *lennupiletitega*, the stem *lenn* and its adjacent suffix *u* are grouped together, as are the stem *pileti* and its suffixes *te* and *ga*:

lennu+ +piletitega

Stems and Affixes

The aim of the second scheme is to imitate the separation of suffixes and/or prefixes from the stem of the word. In contrast with the full segmentation of Morfessor this means grouping together everything but the final suffix(es) or the initial prefix(es). For example several unsupervised approaches described in the related work section are initially designed to segment words into at most two parts (typically stem and suffix) and the aim of this scheme is to imitate them.

Depending on the language only suffixes or only prefixes can be separated. For example Estonian, one of the languages of the experiments with this approach, is predominantly a suffix language and has very few prefixes; therefore we restrict the scheme to segmenting the suffixes off the end of word forms.

The remaining question is what to do when a word ends with several suffixes, like the third example from table 4.1. We define two different strategies and evaluate all of them in practice:

- "One-suffix": only the final suffix is separated from the end of the word. The third example from table 4.1 then becomes

  lennupiletite+ +ga

- "All-suffices": all suffixes are separated from the end of the word. The same example becomes

  lennupileti+ +tega

Both the first and second examples have just one suffix and therefore are segmented in the same way in case of both strategies:

упяч+ +ка,
skyscrap+ +er.

Naturally, in case a word ends with a stem (`STM`) it is not segmented at all.

4.2.3 Translation into a segmented language

Segmenting the source language hardly affects the process of training the translation models and evaluating the quality – if the output is not segmented, it does not need any post-processing and can be evaluated or used directly. On the other hand, if the target language is segmented, the segments, constituting the output of the decoder need to be grouped back into word forms. This requires the information on whether each segment is word-initial, word-final, word-internal or a full word form.

A common approach, used by all related work with the segmentation of the translation output, is augmenting the segments with symbols (e.g. "+"), indicating segmentation spots – similarly to our previous examples. In the output consecutive segments with segmentation indicators are grouped:

$$\text{упяч+ +ка} \Rightarrow \text{упячка},$$
$$\text{sky+ +scrap+ +er} \Rightarrow \text{skyscraper}.$$

In order to make sure that no malformed output is generated, e.g.:

$$\text{sky+ +er +scrap+},$$

the decoder can be modified to give low scores to such hypothesis outputs. We take an alternative, easier approach, which in theory achieves the same effect: we add a second language model into the set of feature functions of the log-linear setup, which is trained on the types of segments (i.e. word-initial, word-internal, etc.) instead of their surface forms. The language model contains only bigrams, which are enough to restrict the output to correct order.

In our experiments the additional language model is used both with our re-grouping schemes and the Morfessor baseline. On the other hand it is only included in phrase-based experiments and not the parsing-based ones: the phrase-based Moses toolkit includes support for additional language and translation models (called factored translation models (Koehn and Hoang, 2007)) while the hierarchical phrase-based Joshua toolkit does not.

4.3 Experiments

In the following we evaluate the proposed method of segmentation experimentally. The main question is whether applying the segmentation schemes results in translation score improvement when the corpus, used for training and applying translation systems, is segmented according to them. The baselines that are used to measure the improvement against are out-of-the-box translation systems with no segmentation. An additional baseline is using Morfessor to segment the words with no re-grouping, the approach of Virpioja et al. (2007).

Corpus	#snt. pairs $\times 10^6$	#words $\times 10^6$		OOV rate, %	
		(English)	(foreign)	(English)	(foreign)
KDE4 et-en	0.18	1.75	1.41	1.1	3.2
Europarl de-en	1.52	42.0	39.8	0.1	0.4
Europarl fi-en	1.59	43.9	31.6	0.1	1.0

Table 4.2: Sizes of the training parts of the corpora, used in the additional experimental evaluation

We use the following abbreviations to mark the baselines and the re-grouping schemes:

- `no-segm` – word-based baseline with no segmentation
- `full-segm` – Morfessor's segmentation with no re-grouping
- `decomp` – modeling compound segmentation with re-grouping schemes
- `one-suf` – modeling stem-suffix segmentation with the last suffix morph of the word
- `all-suf` – modeling stem-suffix segmentation with all the final suffix morphs of the word

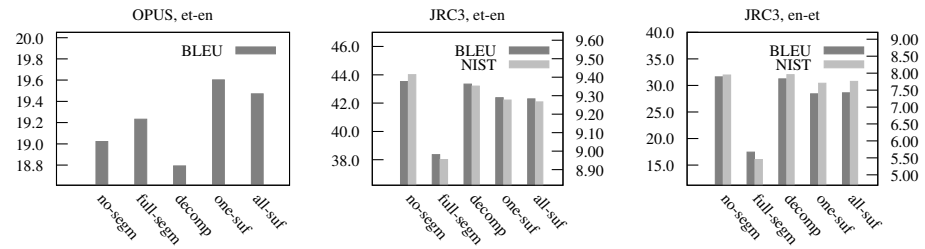
4.3.1 Initial Evaluation

Initial experimental evaluation of our method is published in (Kirik and Fishel, 2008) and (Fishel and Kirik, 2010). In (Kirik and Fishel, 2008) it is applied to segmenting Estonian, prior to translating into English. The used corpus is OPUS OpenSubtitles, with the Estonian-English part being rather small (less than 30 thousand sentence pairs total) and the out-of-vocabulary rate is rather high (7.2%).

In (Fishel and Kirik, 2010) the method is applied to the Estonian-English part of the JRC-Acquis corpus (Steinberger et al., 2006); whereas both translation directions between English and Estonian are tested. The corpus is rather large (over a million sentence pairs) and the ratio of out-of-vocabulary words is below 1%.

The results of those initial experiments are summarized in figure 4.1; it can be seen that translation with segmented corpora compares to non-segmented baselines differently, depending on the corpus.

In case of the OpenSubtitles corpus (part *(a)* of figure 4.1) modeling compound segmentation (`decomp`) is the only scheme that does worse than both baselines (with no segmentation, `no-segm`, and with full Morfessor's segmentation, `full-segm`); both methods of modeling stem-suffix segmentation (using one



(a) (b) (c)

Figure 4.1: Results of applying the introduced method of segmentation prior to training and applying machine translation systems on the example of Estonian-English translation with the OPUS OpenSubtitles corpus and Estonian-English and English-Estonian translation with the JRC-Acquis corpus. The compared models are the non-segmented baseline (`no-segm`), Morfessor baseline (`full-segm`), modeling compound segmentation (`decomp`) and stem-suffix segmentation (`one-suf` and `all-suf`)

word-final suffix, `one-suf`, or all word-final suffices) show noticeable improvement of the BLEU score.

The results of experiments with the JRC-Acquis corpus are on the other hand pessimistic. The Morfessor baseline segmentation causes a large drop in the scores, which is even bigger in case of translation into the segmented language. Our re-grouping schemes perform better than the Morfessor baseline, but the scores never exceed the word-based baseline.

This difference between the experiments on the OpenSubtitles and the JRC-Acquis corpora is understandable – many works on using different methods of segmentation conclude that the influence of using segmentation is much better on smaller corpora and much smaller or negative – on larger corpora.

4.3.2 Additional Evaluation

In addition to the initial evaluation we applied the same method to another corpus of Estonian-English, OPUS KDE4 (Tiedemann, 2009), and two more language pairs from Europarl (Koehn, 2005): Finnish-English and German-English. The sizes of the training parts of the corpora are given in table 4.2.

The choice is motivated by the fact that the KDE4 Estonian-English corpus is larger than the OpenSubtitles and considerably smaller than the JRC-Acquis Estonian-English corpora; Finnish is the language, for which Morfessor was originally developed and is highly agglutinative and German is a heavily compounding language.

Results of the additional evaluation are given in figure 4.2 for Moses and 4.3

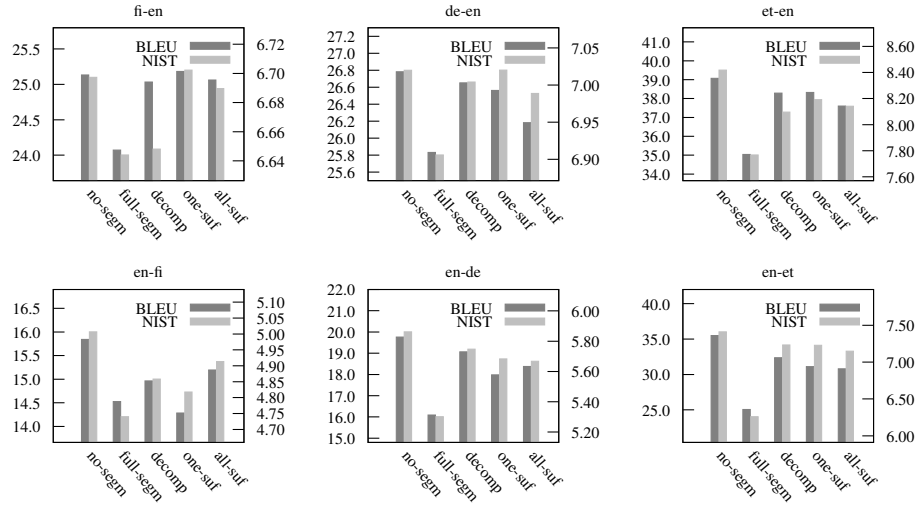


Figure 4.2: Additional experimental evaluation results using Moses. The compared models are the non-segmented baseline (`no-segm`), Morfessor baseline (`full-segm`), modeling compound segmentation (`decomp`) and stem-suffix segmentation (`one-suf` and `all-suf`). The BLEU score scale is on the left, and the NIST score scale – on the right

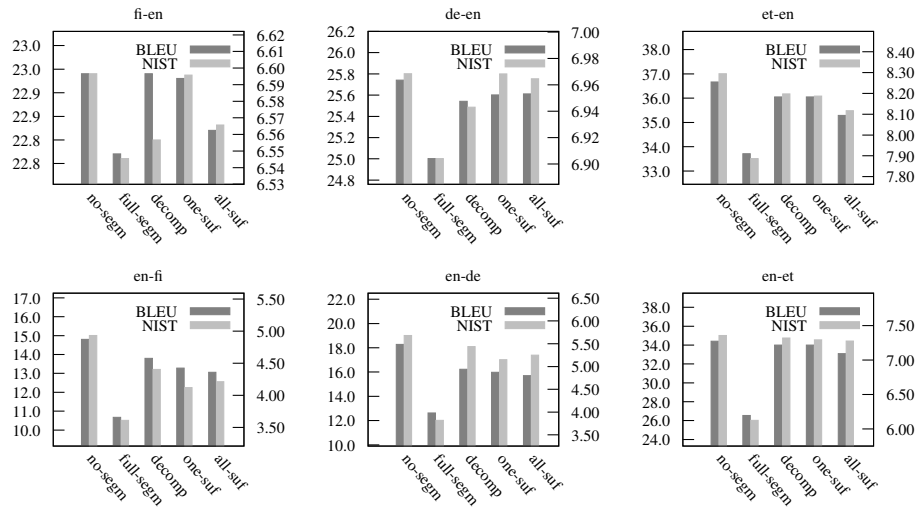


Figure 4.3: Additional experimental evaluation results using Joshua. The compared models are the non-segmented baseline (`no-segm`), Morfessor baseline (`full-segm`), modeling compound segmentation (`decomp`) and stem-suffix segmentation (`one-suf` and `all-suf`). The BLEU score scale is on the left, and the NIST score scale – on the right

Source	sea järjehoidjaribaks
Segmentation	sea/STM järje/PRE+hoid/STM+ +ja/SUF+*riba/SUF*+ks/SUF
Reference	set as bookmark toolbar
no-segm input	sea järjehoidjaribaks
Moses translation	set *järjehoidjaribaks*
Joshua translation	set *järjehoidjaribaks*
full-segm input	sea järje+ +hoid+ +ja+ +riba+ +ks
Moses translation	set *a* bookmark toolbar
Joshua translation	set as bookmark toolbar
decomp input	sea järjehoidjaribaks
Moses translation	set *järjehoidjaribaks*
Joshua translation	set *järjehoidjaribaks*
one-suf input	sea järjehoidjariba+ +ks
Moses translation	set *to* bookmark toolbar
Joshua translation	set as bookmark toolbar
all-suf input	sea järjehoid+ +jaribaks
Moses translation	set *a* bookmark *jaribaks*
Joshua translation	set *jaribaks* bookmarks

Table 4.3: Comparison of the segmentations and translations of an estonian sentence from the KDE4 corpus into English. The wrong classification of the stem *riba* as a suffix results in translation errors in the decomp and all-suf schemes.

for Joshua. Unfortunately the result pattern is the same as with the Estonian-English JRC-Acquis corpus in figure 4.1: the Morfessor baseline results are much lower than the word-based baseline and the re-grouping schemes are closer to it, but still lower. We omit significance testing in this case, since none of our systems achieved a higher score than the baseline.

A possible additional source of error is when Morfessor manages to find a wrong segmentation of a word form or classifies the morphs incorrectly. Consider the example in table 4.3. Although the segmentation of the Estonian word *järjehoidjaribaks* (*as bookmark toolbar*) is correct, it is a compound noun, consisting of *järjehoidja* (*bookmark*) and *ribaks* (*as toolbar*), and the stem *riba* (*toolbar*) is wrongly classified as a suffix. This results in translation errors in the re-grouping schemes decomp and all-suf for both decoders.

The results are again in agreement with the conclusion of most prior work: segmentation does not lead to significant (or any) score improvement on large corpora. This effect is most probably caused by two factors. First of all, the biggest expected effect of segmentation methods is to translate word forms that do not appear in the training corpus, the out-of-vocabulary word forms. On the other hand, the rate of the out-of-vocabulary words is lower for larger corpora. Secondly, the

assumption that word segments or morphs can be treated as independent tokens is not a particularly strong one and heavily depends on the language. This in its turn means that in general word-based translation models are more robust than morph-based ones.

In combination, it is important whether the first or the second factor have a greater effect on translation quality. For small corpora the number of unknown words is so high that the negative effect of the reduced robustness of the morph-based model is not seen in the shadow of the positive effect of the reduced out-of-vocabulary rate. Contrary to that, in case of larger corpora the effect of the reduced out-of-vocabulary rate is smaller than the negative effect of the reduced robustness.

Combining our results with the Morfessor model with previous work (Virpioja et al., 2007), the possible reason for the large score difference with the word-based baseline is that the segmentation is too detailed and as a result the morphs are too ambiguous. In that case our re-grouping schemes are in a sense in the middle between the two baselines, which would explain their scores being significantly higher than the Morfessor baseline, but lower than the word-based baseline scores.

The general conclusion is that our suggested method only seems to produce higher scores than the word-based baseline systems for cases with smaller corpora with a large number of out-of-vocabulary words; in case of large corpora the word-based system results in much higher translation scores. On the other hand our method exceeds the performance of the baseline Morfessor segmentation in all cases.

4.4 Future Work

Methods of segmentation potentially influence the out-of-vocabulary words the most, when used for machine translation. Instead of applying segmentation to all of the input, an alternative would be to only segment the unknown words in the input. This way it could be possible to overcome the weakness of assuming all of the word forms to consist of independent segments and still decrease the out-of-vocabulary rate.

In order for this approach to work the translation model has to have seen both the full form of the known words and the segments of the unknown ones. A possible solution is to use the words of the training corpus that appear only once (i.e. hapax legomena) as a model of the unknown words.

Thus the pipeline of the new translation method would start with training the segmentation model on the whole training corpus, then applying it only to the hapax legomena of training corpus and using it to train a translation model. All the other words of the corpus, with frequencies higher than 1, are saved in a dictionary of “known” words. During translation of the development or evaluation set the

words from the dictionary are kept intact, and all the other words are segmented.

An alternative direction of research is bilingual unsupervised segmentation, which finds a specific case of morphological segmentation for a given language pair and corpus. This way the segmentation is motivated by a similar setup to the task of translation itself. Our initial experiments on bilingual segmentation are presented in (Fishel, 2009), where a simplistic approach to the task is evaluated: all word forms are replaced with a list of all their possible sub-strings and a joint symmetrical alignment model is expected to find the best sub-string subsets without selecting intersecting segments; the experiments are unfortunately too preliminary to be part of this dissertation.

Finally, the advantage of hierarchical phrase-based translation models in conjunction with segmentation methods has to be exploited further.

CHAPTER 5

CHALLENGING DEFAULT WORD ALIGNMENT MODELS

This chapter describes our third and final contribution of this dissertation. The material presented here is published in (Fishel, 2010).

Unlike the work of the first two chapters, where the objective is to improve translation quality, here the aim is instead reducing the time required to train translation models without significant translation quality loss.

A majority of state-of-the-art statistical machine translation systems do not operate with single words, but rather with multi-word units. In phrase-based translation (Koehn et al., 2003) these units are word sequences, or phrases. In parsing-based translation (e.g. Chiang, 2005) these are trees, with words or non-terminal symbols in the tree leaves. However, in most cases word alignment is still used as an intermediate step of learning the translation models – for instance in the Moses toolkit (Koehn et al., 2007) for phrase-based and in Joshua (Li et al., 2009) for parsing-based translation.

The word alignment models that are currently considered as default are the so-called IBM models 1 to 5 (Brown et al., 1993) and the HMM-based alignment model (Vogel et al., 1996). The main work evaluating them is (Och and Ney, 2003) where they are compared in the context of word alignment only – i.e. based on the alignment error rate. Specifically, the default setup of the well known implementation of the models, GIZA++, is derived from (Och and Ney, 2003) and involves training the following models in sequence: IBM model 1, HMM-based model, IBM model 3 and IBM model 4.

Although Och and Ney (2003) mention briefly that "improved alignment quality yields an improved subjective quality of the statistical machine translation system as well", a number of recent studies suggest otherwise – namely, that the correlation between the alignment quality and the quality of the resulting translation is rather weak (see section 5.1 for an overview of the related work). This suggests that the best default word alignment models are not necessarily optimal

in terms of the resulting translation quality.

Some recent works (e.g. (Liang et al., 2006) or (DeNero and Klein, 2007)) are already based on the sequential HMM word alignment model, rather than the fertility-based IBM models 3 or 4 . The former is computationally less complex and at the same time still models the essential aspects of word alignment (i.e. lexical correspondence and changes in word order).

Finding the word alignments is the longest step in learning a phrase-based translation model, as well as takes a substantial amount of time during learning a hierarchical phrase-based model. Our motivation for using simpler word alignment models is thus mainly to save time during the learning stage.

This chapter focuses on formally evaluating the impact of using simpler word alignment models instead of the default setup, the main research question being, whether this can be done without causing decreased translation scores. It is also interesting to compare the influence of alignment quality on the different translation models, as hierarchical phrase-based translation is likely to depend on it.

In section 5.2 we discuss the aspects of word alignment (like lexical correspondence or different word order) that the default models include. Experimental evaluation is presented in section 5.3, where several steps of simplifying the process of word alignment by discarding some of the modeled aspects are tested. Section 5.4 concludes the chapter with a description of possible future work.

5.1 Background and Related Work

Several papers point out that the scores designed for word alignment (alignment error-rate, F-score) and translation (BLEU, NIST), are not heavily correlated. In particular Fraser and Marcu (2007) and Ayan and Dorr (2006) distinctly state that alignment error rate is a poor indicator of translation quality.

Lopez and Resnik (2006) artificially degrade the alignment quality in order to show that it does not cause a significant drop in translation quality. They further show that with careful feature engineering the flaws of the underlying word alignment can be compensated.

Vilar et al. (2006) give two examples of word alignment modifications which cause worse alignment quality and nevertheless better translation quality. This is achieved by adapting the alignments to the specific requirements of translation.

Guzman et al. (2009) inspect word alignments and their characteristics, especially the number of unaligned words, and their influence on phrase pair extraction. They show that an increased number of unaligned words causes degraded translation quality. Analyzing manually evaluated phrase pairs they come up with translation model features that account for the number of unaligned words and improve the translation quality.

Lambert et al. (2009) tune alignment for the F-score and the BLEU score. They show that the two objectives are not the same and produce different translation models.

Ganchev et al. (2008) use agreement-driven training of alignment models and replace Viterbi decoding with posterior decoding. This results in improvements both in the alignment quality as well as translation quality.

A brief comparison of the IBM models in SMT context is performed in (Koehn et al., 2003). The comparison is based on the BLEU scores and covers IBM models 1 to 4. The given brief conclusions are that using different alignment models does not lead to significant changes in translation quality. Model 1 is noted for lower scores and models 2 and 4 are said to produce similar results. Our results suggest the contrary for the latter point.

He (2007) introduce word dependent HMM-based word alignment. They apply fully lexicalized transition modeling by additionally conditioning the first-order dependency of the alignment on the corresponding output word. They show that this modified alignment model can match the performance of IBM model 4. As it will be shown later in this work, usual HMM-based alignment models also achieve the same result.

5.2 Word Alignment Aspects

Before experimentally assessing the impact of different word alignment models on the resulting translation quality, we discuss the aspects of word alignment, included in the IBM models (Brown et al., 1993) and the HMM-based model (Vogel et al., 1996). Including or excluding certain aspects directly influences the expressiveness and the complexity of the corresponding model. These aspects include lexical correspondence, distortion and fertility.

5.2.1 Lexical Correspondence

Lexical (or translational) correspondence of the single words in the two sentences is perhaps the main aspect of word alignment and is present in all of the described models. It denotes that the selected words have the same (or similar) meaning in their corresponding languages.

In all the models considered here lexical correspondence is treated as independent of the word positions in the sentences or any context of either words. As already described in chapter 2, it is modeled via a probability distribution $p(f|e)$ denoting the probability of translating the word e with the word f.

Although lexical correspondence is a very important aspect, constraining an alignment model to it (as it is the case with model 1) results in serious model flaws with the current independent treating of the word pairs. In case of any

lexical ambiguity the model will select the most probable word pair in all cases. In addition the model would not be able to resolve a conflict between repeated items, like punctuation marks or same words.

On the other hand context-independent treating of the lexical correspondence enables fast and easy Expectation Maximization-based learning its parameters $p(f|e)$ from unaligned data and also for aligning new sentence pairs (see eqn. 2.3 from chapter 2).

5.2.2 Distortion

The different word order in the two sentences is also referred to as distortion (Brown et al., 1993). Depending on the exact model, different independence assumptions and generalizations are made.

Absolute position-based distortion

The IBM models 2 and 3 include a distortion component based on the absolute word positions and the sentence lengths: $p(a_j|j, J, I)$, where $J = |\mathbf{f}|$ and $I = |\mathbf{e}|$. The component is independent of the source and target words and thus tries to learn typical positions for a target word, given the source word position and sentence lengths.

The problem with absolute word positions is that, simply put, same words can occur at different positions in sentences of different length. This means that a separate parameter subset models each different position and sentence length which, in addition to unnecessary treating of the same words differently, can easily suffer from the sparse data effect.

Again, in the same way as with the lexical correspondence, the independence assumption allows efficient estimation of the parameters $p(a_j|j, J, I)$.

Relative shift-based distortion

A modification of the original model 2, introduced originally in (Vogel et al., 1996) and developed further in (Och and Ney, 2000), is instead based on the relative distance between a_j and a scaled j:

$$d = a_j - \left\lfloor \frac{j \cdot I}{J} \right\rfloor .$$

Out-of-range values ($d < -N,\ d > N$) are grouped together, resulting in a multinomial distribution with a fixed number of parameters: $p(d)$. The model is called by its authors "diagonal-oriented" model 2.

In other words this modification models the typical relative reordering distances for the words. This is a strong generalization over the absolute position-based modeling, and although the latter is vulnerable to sparse data, it remains under question, whether this model is detailed enough to be useful.

Again, context independence of the parameters allows their simple estimation.

Distortion First-order Dependency

Vogel et al. (1996) make a step further from independent single pair distortions. They treat alignment as a Markov process with the source words as the observed variables and the alignments – as hidden variables, which results in a model called HMM-based alignment. With first-order Markov dependency assumption the alignment pairs are not any more dependent only on the position of the word itself, but the previously aligned word pair, which is a simple way to take into consideration the context of the aligned word pair. If a neighboring (unambiguous) word pair is aligned with an atypical relative distortion, an HMM-based model is capable of deciding to align the current word similarly.

The parameters are designed similarly do diagonal-oriented model 2 – instead of absolute position-based modeling, the HMM-based model uses the relative distance between the previous alignment and the current one:

$$d = a_j - a_{j-1},$$

the distance is truncated just like with the diagonal-oriented model and is used for the parameter set $p(d)$.

First-order dependency of the distortion parameters makes the training and estimation algorithms more complicated in comparison to the previously discussed modeling of word alignment aspects. Summing over alignments can be achieved with the Viterbi algorithm (Viterbi, 1967), as proposed by Och and Ney (2000).

Lexicalized Distortion

Och and Ney (2000) introduce lexicalized reordering. This means that reordering distribution is additionally conditioned on the words in question – thus allowing different words to be reordered in different ways, which is very sound from the point of view of linguistic intuition.

The suggested way of modeling is suitable for all models of distortion, described here: the words from the aligned pair are simply used as a condition in the distortion probability distribution; for example in case of HMM-based alignment we have $p(d|f_j, e_{a_j})$. This modification does not affect the learning algorithm complexity, as it only adds dependencies inside the aligned word pair. In order to reduce distribution sparsity some general classes of words are used instead of the

Language pair	#sentence pairs	#words (English)	#words (Foreign)
Korean-English	$64.1 \cdot 10^3$	$0.32 \cdot 10^6$	$0.33 \cdot 10^6$
Chinese-English	$103.7 \cdot 10^3$	$0.57 \cdot 10^6$	$0.78 \cdot 10^6$
Czech-English	$0.97 \cdot 10^6$	$7.27 \cdot 10^6$	$6.59 \cdot 10^6$
Estonian-English	$1.09 \cdot 10^6$	$27.91 \cdot 10^6$	$20.18 \cdot 10^6$
German-English	$1.52 \cdot 10^6$	$41.98 \cdot 10^6$	$39.81 \cdot 10^6$
Finnish-English	$1.59 \cdot 10^6$	$43.94 \cdot 10^6$	$31.58 \cdot 10^6$

Table 5.1: The size of the training parts of the used parallel corpora

words themselves: $C(f_j)$ and $C(e_{a_j})$:

$$p(d|C(f_j), C(e_{a_j})).$$

These classes can be for instance parts-of-speech; in practice, however, they are commonly derived by clustering the word forms prior to training the translation model.

By default lexicalization is used in the HMM-based model and IBM models 3 and 4.

5.2.3 Fertility

Fertility is used to model the number of target words aligned to the same source word. For example if one sentence belongs to an analytical language and the other one – to an inflectional one, naturally the words of the inflectional language will tend to be associated with several words in the analytical language. In models 3, 4 and 5 this is modeled explicitly: $p(\phi|e)$, where ϕ denotes the number of words in $\mathbf{f}$ that e is connected to.

While model 3 models distortion in the same way as model 2 (based on absolute positions), models 4 and 5 use fertility-motivated distortion, which uses both relative reordering and first-order dependency. The target words aligned to the same source word are treated as a single group (called "cept" in (Brown et al., 1993)). The position of the head of the cept i is chosen, relative to the center of the previously produced cept $(i-1)$ and the rest of the words in cept i are placed, relative to each other in consequence (word k using the relative distance to word $(k-1)$).

As described in chapter 2, fertility and cept-based distortion make it impossible to sum over all alignments efficiently. Instead the sum is done only over the most probable alignments, which still results in a much slower estimation step.

5.3 Experiments

In this section we compare the influence of the word alignments on the resulting translation quality. The main aim is to test whether it is necessary to use the complex default models or not – i.e., whether simpler models can match their performance in terms of translation scores.

The influence of word alignment is evaluated on phrase-based translation, implemented in Moses, and parsing-based translation, implemented in Joshua, using the following language pairs: the Chinese-English and Korean-English parts of the OPUS KDE4 corpus (Tiedemann, 2009), Czech-English technical documentations from CzEng (Bojar and Žabokrtský, 2009), the Estonian-English part of the JRC-Acquis (Steinberger et al., 2006) and the Finnish-English and German-English parts of Europarl (Koehn, 2005); all experiments included both translation directions. The sizes of the training parts of the corpora are given in table 5.1 and the size of the random held-out development and evaluation sets was 2500 sentence pairs, same as in all the other experiments of this dissertation. The selection of the corpora is motivated mainly by the diversity in languages and corpora sizes.

The default setup of GIZA++ (Och and Ney, 2003), which is the implementation of word alignments that we used, is to start with the simplest model (IBM model 1), proceeding with more elaborate ones (HMM-based model, IBM model 3 and finally, model 4), which use the parameters of the simpler models as a starting point. In other words, in terms of word alignment aspects the model starts by learning the lexical correspondence, then estimates distortion and then – fertility. Modeling of distortion is similar in the HMM-based model and the IBM model 4 and is much more simplistic in IBM model 3.

We first evaluate, how stopping at an earlier stage of word alignment influences translation quality.

5.3.1 Early Stopping

The results of stopping learning at the earlier models instead of IBM model 4 are presented in figure 5.1 for phrase-based and in figure 5.2 – for parsing-based translation. The general pattern, repeated for all language pairs and both translation with almost no change is that the scores of model 1 are the lowest, the HMM-based model scores are very high, followed by a drop in the scores of model 3 and finally the scores of model 4 return back to approximately the level of the HMM scores.

The scores of the HMM-based model and of model 4 reside closely, their exact difference and results of statistical significance testing are given in table 5.2. It can be seen that in most cases the HMM scores are either significantly higher or insignificantly different from model 4. The only experiment with consistently worse BLEU and NIST scores is the parsing-based German-English translation.

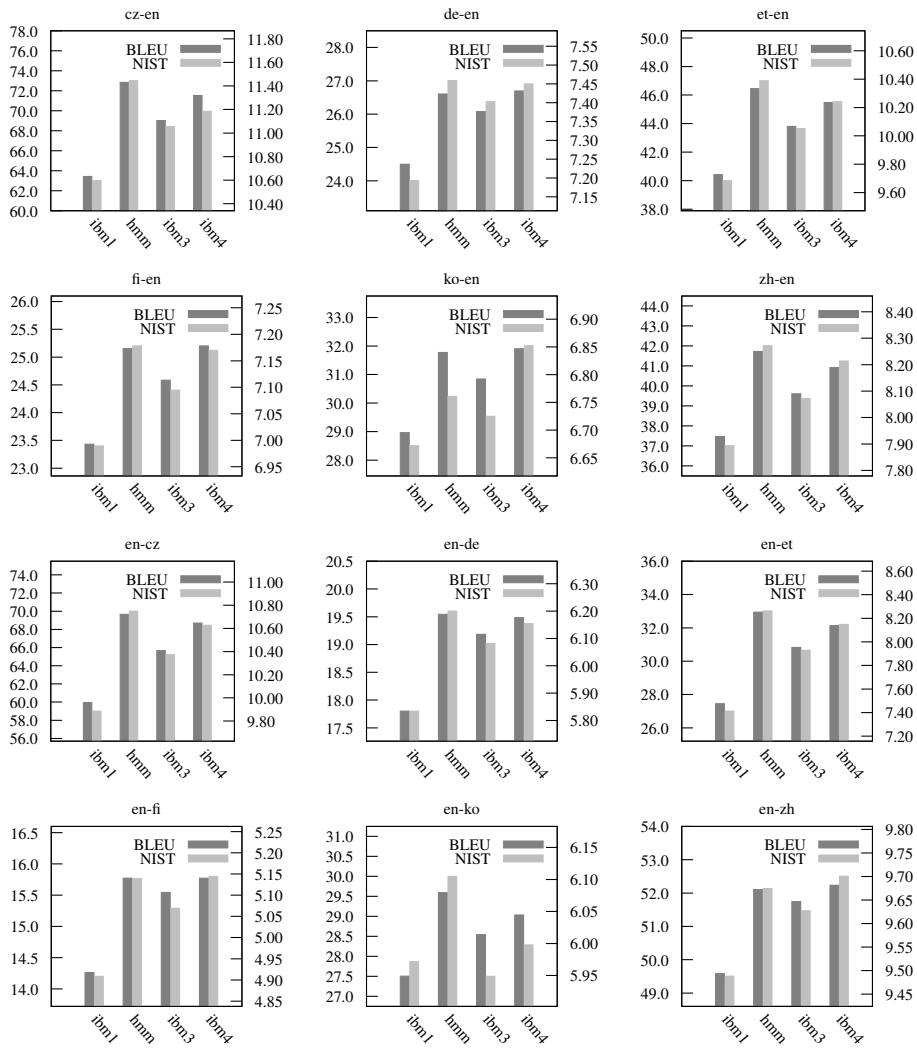


Figure 5.1: Experiment results for the phrase-based translation-based experiments on early stopping. The BLEU scale is on the left and the NIST scale – on the right

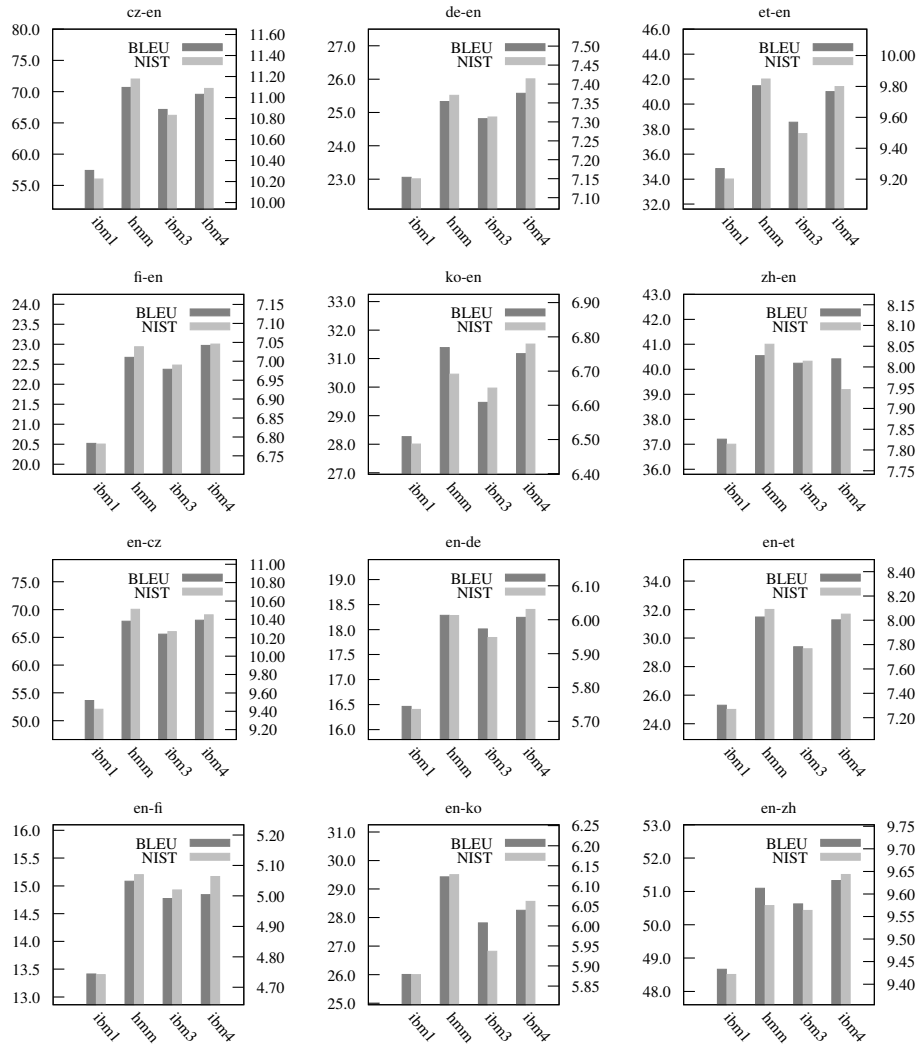


Figure 5.2: Experiment results for the parsing-based translation-based experiments on early stopping. The BLEU scale is on the left and the NIST scale – on the right

	Moses		Joshua	
	BLEU	NIST	BLEU	NIST
en-cz	0.96 (0.036)	0.1221 (0.003)	-0.18 (0.287)	0.0588 (0.123)
cz-en	1.31 (0.004)	0.2599 (0.000)	1.11 (0.028)	0.0910 (0.048)
en-de	0.06 (0.224)	0.0455 (0.008)	0.04 (0.272)	-0.0182 (0.145)
de-en	-0.09 (0.157)	0.0089 (0.215)	-0.25 (0.039)	-0.0432 (0.011)
en-et	0.80 (0.000)	0.1139 (0.000)	0.21 (0.125)	0.0399 (0.091)
et-en	0.97 (0.000)	0.1482 (0.000)	0.47 (0.015)	0.0494 (0.032)
en-fi	0.00 (0.437)	-0.0055 (0.303)	0.24 (0.054)	0.0062 (0.300)
fi-en	-0.05 (0.275)	0.0084 (0.230)	-0.30 (0.039)	-0.0067 (0.267)
en-ko	0.56 (0.060)	0.1073 (0.006)	1.17 (0.002)	0.0669 (0.064)
ko-en	-0.13 (0.263)	-0.0909 (0.006)	0.21 (0.190)	-0.0876 (0.017)
en-zh	-0.13 (0.209)	-0.0262 (0.123)	-0.23 (0.170)	-0.0689 (0.015)
zh-en	0.81 (0.009)	0.0574 (0.061)	0.13 (0.285)	0.1095 (0.007)

Table 5.2: Score difference and their p-values, resulting from the significance tests between the HMM and IBM4 models. Insignificant differences are marked with gray, significantly better scores of the HMM model – with green and significantly worse scores – with blue

Although the difference is significant, it is rather small (0.25 BLEU and 0.04 NIST points). Experiments with consistently significantly better scores include Czech-English and Estonian-English with both types of translation, as well as English-Czech and English-Estonian with phrase-based translation. Several other cases include one significantly better score with the other score being insignificantly different; also Korean-English phrase-based translation and a couple of parsing-based experiments have one score of the HMM-based model significantly lower than model 4; the biggest difference is again small (0.3 BLEU, 0.09 NIST points).

These results support the common knowledge of the HMM-based model being roughly equivalent in terms of resulting translation quality to the default setup with model 4. Considering that the main difference is the absence of fertility and cept-based distortion in the HMM model, this is a significant optimization step.

The low scores of model 1 are understandable, due to the drawbacks, discussed in section 5.2 – its assumptions make it impossible to resolve lexical conflicts. The low scores of model 3 most probably indicate that the absolute position-based distortion modeling is inferior to the relative distortion and its first-order dependency, present in both HMM-based model and model 4, which is not entirely unexpected.

Approximate estimation of the time it takes to produce the word alignments shows that the HMM-based alignment model is in average twice as fast to com-

	Moses		Joshua	
	BLEU	NIST	BLEU	NIST
en-cz	0.72 (0.031)	0.0870 (0.005)	0.09 (0.344)	0.0684 (0.058)
cz-en	0.32 (0.144)	0.0367 (0.127)	1.51 (0.001)	0.1400 (0.001)
en-de	0.30 (0.011)	0.0356 (0.025)	0.11 (0.139)	0.0092 (0.214)
de-en	-0.11 (0.132)	0.0212 (0.083)	0.08 (0.179)	-0.0227 (0.071)
en-et	0.39 (0.010)	0.0436 (0.035)	0.79 (0.000)	0.0318 (0.091)
et-en	0.81 (0.000)	0.0410 (0.031)	0.33 (0.033)	0.0271 (0.110)
en-fi	0.08 (0.182)	-0.0263 (0.092)	0.29 (0.015)	0.0141 (0.194)
fi-en	-0.18 (0.078)	-0.0445 (0.008)	0.06 (0.239)	-0.0170 (0.123)
en-ko	0.89 (0.002)	0.0650 (0.023)	-0.03 (0.369)	0.0649 (0.036)
ko-en	0.02 (0.367)	-0.0055 (0.333)	0.67 (0.036)	0.0498 (0.053)
en-zh	0.01 (0.415)	-0.0179 (0.184)	0.11 (0.265)	-0.0072 (0.312)
zh-en	0.41 (0.076)	0.0358 (0.114)	0.51 (0.056)	0.1452 (0.000)

Table 5.3: Score difference and their p-values, resulting from the significance tests between the IBM4x and IBM4 models. Insignificant differences are marked with gray, significantly better scores of the IBM4x model – with green and significantly worse scores – with blue

plete as the IBM model 4; we leave more strict evaluation of the time savings for future work.

Since the higher-order models use their predecessor resulting parameters as initial values, it is possible that model 3 serves as the bottleneck and thus causes model 4 to have worse scores than it would be otherwise possible. In the following we evaluate an attempt to fix this.

5.3.2 Skipping IBM Model 3

To avoid model 3 "dragging down" the results we test a modified model training succession with the model 3 omitted. Thus the IBM model 4 gets its scores directly from the HMM-based model. Thus the complexity of the final model is the same, but the training process includes less iterations since no time is wasted on model 3.

The results are given in figures 5.3 and 5.4 for phrase-based and parsing-based translation, respectively; the new setup is abbreviated as `IBM4x`. We can see that in most cases the scores are very close to the original setup, with some small differences in favor of the new setup. Significance testing, the results of which are given in table 5.3, confirms this comparison: a large portion of experiments have insignificantly different scores for IBM model 4 and "model 4x", the

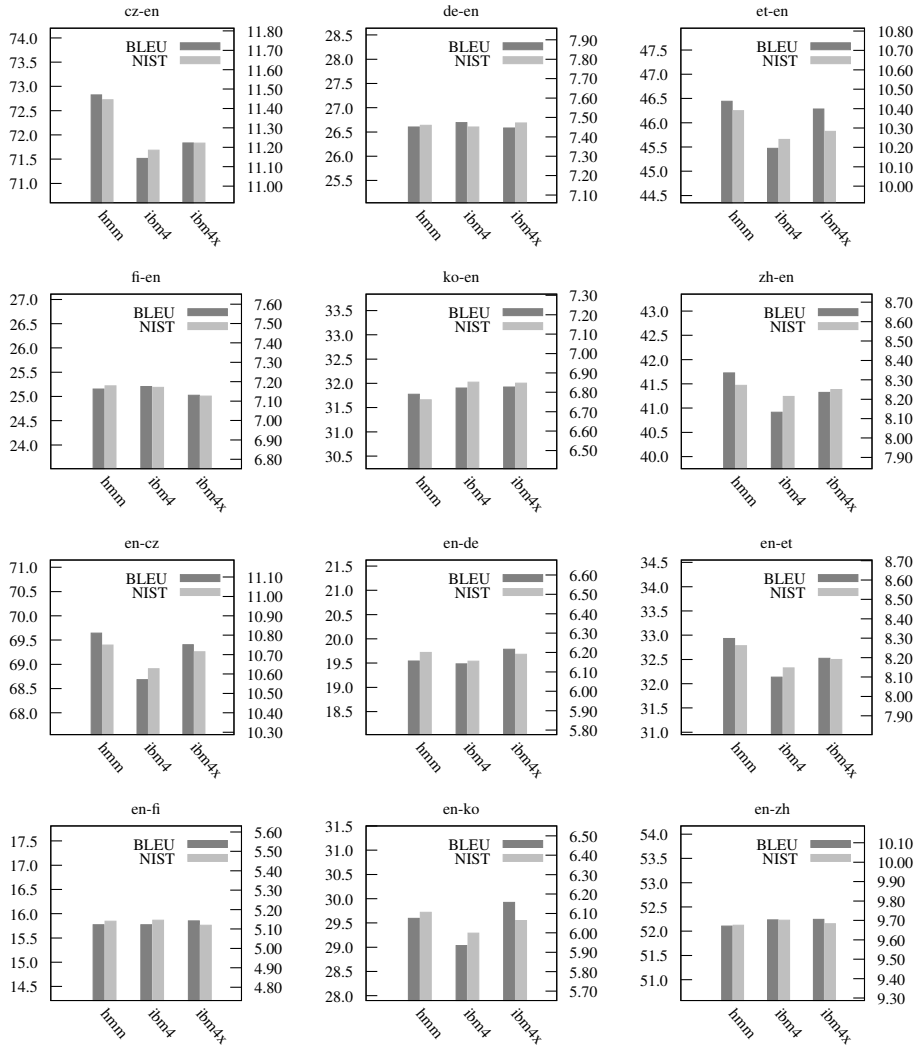


Figure 5.3: Experiment results for the phrase-based translation-based experiments on skipping model 3. The BLEU scale is on the left and the NIST scale – on the right

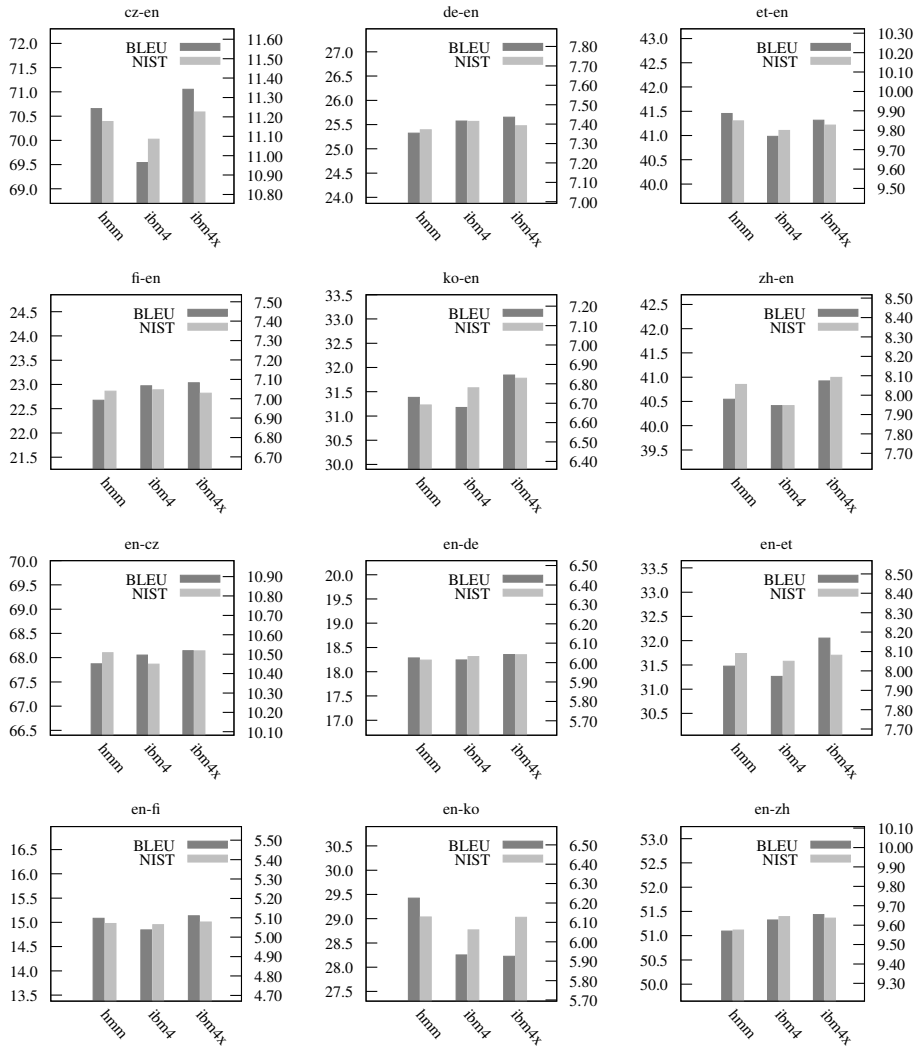


Figure 5.4: Experiment results for the parsing-based translation-based experiments on skipping model 3. The BLEU scale is on the left and the NIST scale – on the right

	Moses		Joshua	
	BLEU	NIST	BLEU	NIST
en-cz	0.24 (0.201)	0.0351 (0.138)	-0.27 (0.233)	-0.0096 (0.332)
cz-en	0.99 (0.008)	0.2232 (0.000)	-0.40 (0.152)	-0.0490 (0.126)
en-de	-0.24 (0.034)	0.0099 (0.216)	-0.07 (0.212)	-0.0274 (0.065)
de-en	0.02 (0.328)	-0.0123 (0.170)	-0.33 (0.010)	-0.0205 (0.114)
en-et	0.41 (0.012)	0.0703 (0.006)	-0.58 (0.004)	0.0081 (0.288)
et-en	0.16 (0.143)	0.1072 (0.000)	0.14 (0.173)	0.0223 (0.155)
en-fi	-0.08 (0.176)	0.0208 (0.137)	-0.05 (0.245)	-0.0079 (0.291)
fi-en	0.13 (0.128)	0.0529 (0.004)	-0.36 (0.012)	0.0103 (0.228)
en-ko	-0.33 (0.115)	0.0423 (0.125)	1.20 (0.002)	0.0020 (0.395)
ko-en	-0.15 (0.228)	-0.0854 (0.009)	-0.46 (0.089)	-0.1374 (0.000)
en-zh	-0.14 (0.209)	-0.0083 (0.280)	-0.34 (0.097)	-0.0617 (0.022)
zh-en	0.40 (0.082)	0.0216 (0.191)	-0.38 (0.137)	-0.0357 (0.143)

Table 5.4: Score difference and their p-values, resulting from the significance tests between the HMM and IBM4x models. Insignificant differences are marked with gray, significantly better scores of the HMM model – with green and significantly worse scores – with blue

only significantly lower of the latter is NIST in the Finnish-English phrase-based translation. Experiments with consistently better scores include Czech-English parsing-based and English-Czech, English-German, English-Estonian, English-Korean and Estonian-English phrase-based translation. The score differences are relatively small.

The lack of considerable improvement might also be explained by the implementation of the alignment model transitions in GIZA++: skipping model 3 can only be achieved by setting its number of iterations to 0, which still means that the HMM model parameters are transferred into model 3 and immediately transferred further into model 4. This might still serve as a bottleneck of the estimation, as the distortion in model 3 is described with a weaker absolute position-based component. This can be fixed by changing the implementation of the parameter transfer, which is currently left for future work.

Next, we compare the new setup to the still simpler setup of stopping after the HMM-based model training in table 5.4. It can be seen though that no experiments have consistent significantly worse scores; the inconsistent significantly worse scores include differences of up to 0.58 BLEU and 0.13 NIST points, which is still a rather small difference. Most experiments have insignificantly different scores between the HMM-based model and model 4 of the new setup, while phrase-based Czech-English and Estonian-English translation experiments have

	Moses		Joshua	
	BLEU	NIST	BLEU	NIST
en-cz	-1.01 (0.056)	-0.1138 (0.020)	1.28 (0.045)	0.0409 (0.193)
cz-en	-1.68 (0.001)	-0.2260 (0.000)	0.86 (0.076)	0.0695 (0.110)
en-de	-0.31 (0.013)	-0.1230 (0.000)	-0.25 (0.033)	-0.0261 (0.090)
de-en	-0.56 (0.000)	-0.0780 (0.000)	-0.30 (0.030)	-0.0234 (0.087)
en-et	-2.00 (0.000)	-0.3046 (0.000)	-1.23 (0.000)	-0.2299 (0.000)
et-en	-2.44 (0.000)	-0.3304 (0.000)	-2.17 (0.000)	-0.2494 (0.000)
en-fi	-0.59 (0.000)	-0.0803 (0.003)	-0.41 (0.005)	-0.0463 (0.045)
fi-en	-0.81 (0.000)	-0.0892 (0.000)	-0.65 (0.000)	-0.0979 (0.000)
en-ko	-0.72 (0.045)	-0.1364 (0.002)	-2.17 (0.000)	-0.0894 (0.039)
ko-en	-0.81 (0.041)	-0.1853 (0.000)	-1.31 (0.004)	-0.1302 (0.002)
en-zh	-0.62 (0.026)	0.0064 (0.338)	-0.59 (0.049)	-0.0680 (0.035)
zh-en	-1.29 (0.001)	-0.1474 (0.002)	-0.44 (0.133)	-0.0893 (0.040)

Table 5.5: Score difference and their p-values, resulting from the significance tests between the IBM2r and HMM models. Insignificant differences are marked with gray, significantly better scores of the IBM2r model – with green and significantly worse scores – with blue

consistently better scores.

The conclusion of the experiments with the new setup is still that for the most part the HMM-based model can replace model 4 with no degradation of the translation scores in most cases and very small degradation – in the remaining cases. This means that modeling fertility during the word alignment phase does not result in significant translation improvements in most tested cases.

In the following we compare the performance of the HMM-based model to yet simpler alternative IBM models.

5.3.3 Replacing the HMM-based Model

Vogel et al. (1996) describe a modification of IBM model 2, which models distortion based on relative position changes and is called diagonal-oriented model 2. While keeping the complexity of the model estimation as simple as for model 1, its distortion component is much more general than the one of the original IBM model 2. In the following we compare these variations of model 2 to the HMM-based model. We modified GIZA++, which originally supports only the absolute position-based model 2, to implement the diagonal-oriented model 2.

The GIZA++ implementation of the HMM model uses lexicalization, which is suggested by Och and Ney (2000) for all models. Since lexicalization is a part of

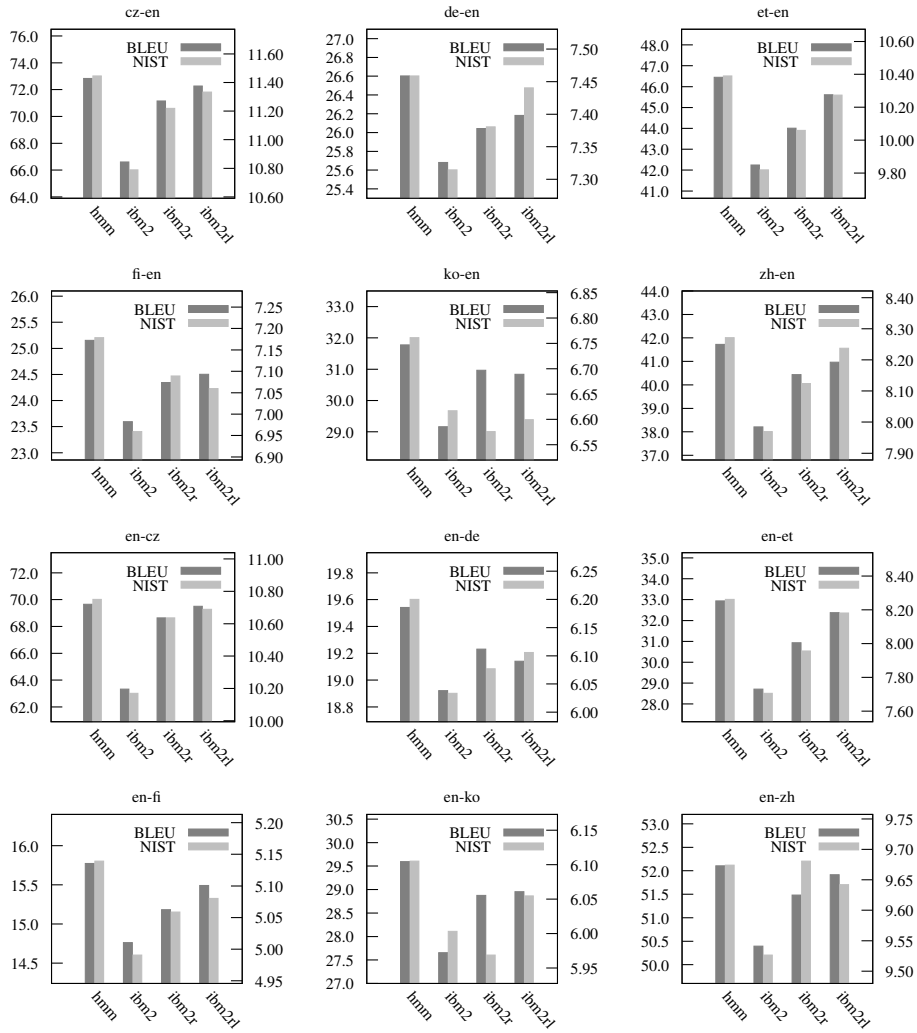


Figure 5.5: Experiment results for the phrase-based translation-based experiments on replacing the HMM-based model. The BLEU scale is on the left and the NIST scale – on the right

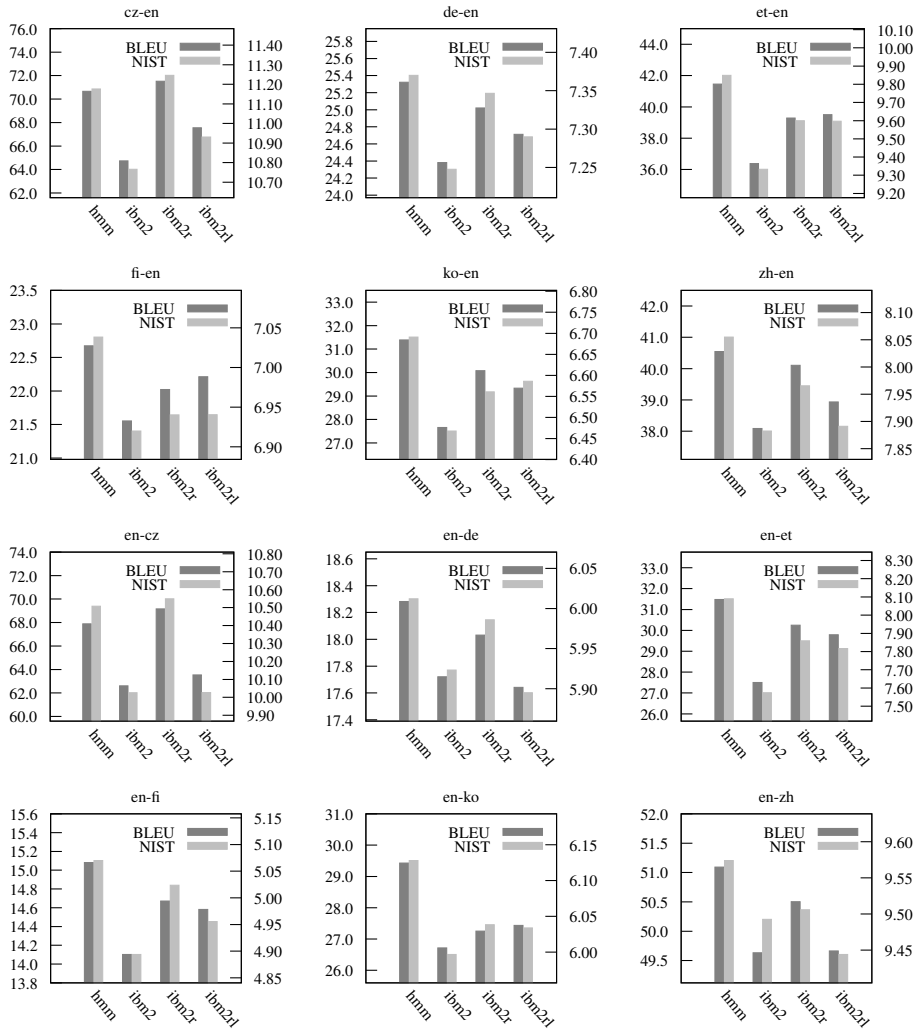


Figure 5.6: Experiment results for the parsing-based translation-based experiments on replacing the HMM-based model. The BLEU scale is on the left and the NIST scale – on the right

	Moses		Joshua	
	BLEU	NIST	BLEU	NIST
en-cz	-0.15 (0.303)	-0.0605 (0.092)	-4.35 (0.000)	-0.4816 (0.000)
cz-en	-0.57 (0.092)	-0.1119 (0.014)	-3.11 (0.000)	-0.2450 (0.000)
en-de	-0.40 (0.004)	-0.0941 (0.000)	-0.64 (0.000)	-0.1172 (0.000)
de-en	-0.42 (0.000)	-0.0182 (0.120)	-0.61 (0.000)	-0.0800 (0.000)
en-et	-0.56 (0.006)	-0.0799 (0.009)	-1.69 (0.000)	-0.2722 (0.000)
et-en	-0.84 (0.000)	-0.1163 (0.000)	-1.96 (0.000)	-0.2514 (0.000)
en-fi	-0.28 (0.024)	-0.0588 (0.003)	-0.50 (0.003)	-0.1143 (0.000)
fi-en	-0.65 (0.000)	-0.1189 (0.000)	-0.46 (0.004)	-0.0978 (0.000)
en-ko	-0.64 (0.079)	-0.0503 (0.128)	-1.99 (0.000)	-0.0939 (0.024)
ko-en	-0.94 (0.029)	-0.1617 (0.000)	-2.05 (0.004)	-0.1050 (0.028)
en-zh	-0.19 (0.182)	-0.0322 (0.126)	-1.43 (0.000)	-0.1301 (0.000)
zh-en	-0.76 (0.051)	-0.0338 (0.155)	-1.61 (0.000)	-0.1637 (0.000)

Table 5.6: Score difference and their p-values, resulting from the significance tests between the IBM2rl and HMM models. Insignificant differences are marked with gray, significantly better scores of the IBM2rl model – with green and significantly worse scores – with blue

the models with the highest scores so far (IBM4 and HMM), we include it into our implementation of the diagonal-oriented model 2 as an alternative. We thus use three versions of the model 2 – the original absolute position-based (abbreviated as `IBM2`), the diagonal-oriented model 2 (abbreviated as `IBM2(r)`) and a lexicalized version of the latter (abbreviated as `IBM2(r-l)`); this enables evaluation of the different ways of modeling distortion.

The score comparison is given in figure 5.5 for phrase-based and 5.6 – for parsing-based translation. It is clearly seen that the scores of the original model 2 are the lowest. The reason is most probably the weakness of the absolute position-based modeling of distortion. We thus exclude the original model 2 from further comparisons.

The scores of the diagonal-oriented model 2 variants on the other hand compare to the scores of the HMM-based model differently depending on the decoder and the language pair. The differences and significance test results are given in table 5.5 for the non-lexicalized version and in table 5.6 – for the lexicalized version of model 2 and the HMM-based model.

In most cases the scores of the `IBM2(r)` and `IBM2(r-l)` are significantly worse than the score of the HMM model; some exceptions include English-Czech and Czech-English parsing-based translation with `IBM2(r)` and English-Czech, English-Korean, English-Chinese and Chinese-English phrase-based translation

with `IBM2(r-l)`, which all have scores, insignificantly different from the HMM model.

Looking closer at the significant score differences, we can separate the small ones from the bigger ones. Phrase-based translation results with the `IBM2(r-l)` model are all below 1 BLEU point and 0.2 NIST points. The same alignment model for parsing-based translation causes the worst score difference for the language pairs with Czech and Estonian in them (for both directions); also language pairs with Korean and Chinese have high BLEU score differences, but moderate NIST score differences. In case of `IBM2(r)`, the worst score differences are for both phrase-based and parsing-based translation from and into Estonian, as well as parsing-based translation from and into Korean, measured with BLEU. All the other score differences are around 1 BLEU point and 0.1 NIST point.

Due to differences in the implementations of the alignment models it was not possible to evaluate the amount of time that can be saved by replacing the HMM-based model with one of the IBM2 variants – mainly since the HMM-based model implementation is multi-threaded, while our IBM2 implementations are single-threaded. We leave the strict evaluation of training time of the alignment models for future work.

To conclude, only in some cases it is that the diagonal-oriented model 2 leads to scores, insignificantly different from the HMM-based model. Approximately half of the remaining cases has moderate score difference and the other half – considerable difference (more than 2 BLEU and 0.2 NIST points). It can thus be used as a good trade-off between translation quality and training time.

5.4 Future Work

The current implementation of the diagonal-oriented model 2 was done only for the purpose of performing the currently presented experiments. In order to enable other researchers to use it, it has to be integrated better with GIZA++, so that it can be selected as an alternative.

Another modification of GIZA++ is completely excluding model 3 from the training process. As mentioned in the result analysis, it can still work as a bottleneck of the learning, despite its number of iterations set to 0.

Restricting the modifications to GIZA++ (instead of releasing an alternative alignment tool) is necessary because they use essentially the same models, but decrease the training time in most cases the model complexity without additional cost of replacing the word alignment tool that is used as default at the moment.

Finally, it is necessary to estimate, how much time exactly is saved by replacing the more complex alignment models with the simpler ones.

CHAPTER 6

CONCLUSIONS

We have presented several series of experiments, evaluating the introduced methods of modifying the initial steps of the pipeline of the log-linear statistical machine translation framework. In the following we conclude each method and its evaluation separately.

We have described the problem of overlapping parallel corpora with the difficulties and benefits that their existence offers. We introduced a method of analyzing the overlapping parts of the corpora, which is based on comparing the correspondence of the two language parts of both corpora to each other. Our method enables handling overlapping corpora and exploiting overlaps to perform additional analysis of the corpora, checking and improving the quality of the corpora and increasing their size.

Analysis of the pair of the corpus of the University of Tartu and the Estonian-English part of the JRC-Acquis corpus reveals useful information about the corpora. Despite a significant overlap, these corpora are rather heterogeneous in nature. Unlike our assumption, the large difference in the sizes of the overlapping parts is caused by sentence pairs, present in just one of the corpora, and not by a significantly different level of sentence segmentation.

The versions of the JRC-Acquis, based on Vanilla and HunAlign alignments turn out not only to be rather homogeneous, but to include the same alignments in the majority of cases.

In comparison with the baseline methods of combining the corpora for machine translation, using our method results in higher translation scores in most cases. The score difference seems to depend directly on the number of omitted material in one or the other corpus as well on whether the corpora are homogeneous or heterogeneous.

Future work includes extending the tool to support other types of corpora besides parallel, and adding functionality of using one of the language parts of the parallel corpora to group the other non-matching parts to create new corpora without repeating the annotation or sentence alignment process.

We have introduced a method of unsupervised segmentation that uses classification of the produced segments and models linguistic phenomena, like compounding and stems and lemmas, by defining schemes of re-grouping the segments based on their classes. The method is evaluated in context of machine translation, where it is used to segment the source or the target language prior to training the translation model and translating new sentences.

Experimental evaluation reveals that the method works better than the unsupervised segmentation baseline without the re-grouping schemes. Comparison with the word-based baseline depends heavily on the size of the corpora – the method leads to increased scores on small corpora and has a negative effect on large corpora. The most probable reason is the weakness of the assumption that word segments can be treated as independent translation units, and the out-of-vocabulary rate, which is much lower in large corpora.

Future work includes applying the method to segmenting unknown words to decrease the out-of-vocabulary rate. Also in our opinion the methods of bilingual unsupervised segmentation are a potentially beneficial direction of research.

Finally, we focused on the word alignment models, considered as default – the IBM models 1 to 5 and the HMM-based model. Since these were originally evaluated with alignment error rate, which was shown to correlate weakly with translation quality, we re-evaluated the models in context of machine translation.

Results show that in the majority of cases using the simpler HMM-based model results in the same or better translation scores in comparison to the commonly used IBM model 4. A modification of the IBM model 2, introduced together with the HMM-based model, diagonal-oriented model 2, causes lower scores than the HMM model. However, since in many cases the difference is small or insignificant, it can be used instead of the HMM-based model as a good trade-off between translation quality and model complexity.

Future work consists of modifying the default implementation of the alignment models, GIZA++, to remove the bottlenecks of training the models and to enable using the diagonal-oriented model 2 as an alternative.

To conclude the dissertation in general, the approach of modifying the input to statistical machine translation learning and applying stages is a powerful means of influencing the resulting translation quality without making changes to the core functionality of the models themselves. Some of our contributions lead to stable improvement of translation scores or model complexity in all or most evaluated language pairs and translation models, while others appear to heavily depend on additional factors, which are mainly the characteristics and size of the training corpora.

BIBLIOGRAPHY

Ayan, N. F. and Dorr, B. J.: 2006, *Going beyond AER: An extensive analysis of word alignments and their impact on MT*, in Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'2006)

Badr, I., Zbib, R., and Glass, J.: 2008, *Segmentation for English-to-Arabic Statistical Machine Translation*, in Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'2008), pp 153–156, Columbus, Ohio

Banerjee, S. and Lavie, A.: 2005, *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments*, in Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp 65–72, Ann Arbor, Michigan

Baum, L. E.: 1972, *An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of a Markov Process*, Inequalities **3**, 1–8

Bojar, O., Matusov, E., and Ney, H.: 2006, *Czech-English Phrase-Based Machine Translation*, in Proceedings of the 5th International Conference on Natural Language Processing (FinTAL'2006), pp 214–224, Turku, Finland

Bojar, O. and Žabokrtský, Z.: 2009, *CzEng0.9: Large Parallel Treebank with Rich Annotation*, The Prague Bulletin of Mathematical Linguistics 92

Brown, P. F., Pietra, S. D., Pietra, V. J. D., and Mercer, R. L.: 1993, *The Mathematics of Statistical Machine Translation: Parameter Estimation*, Computational Linguistics **19(2)**, 263–311

Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M., and Zaidan, O.: 2010, *Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation*, in Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, pp 17–53, Uppsala, Sweden

Callison-Burch, C., Koehn, P., Monz, C., and Schroeder, J.: 2009, *Findings of the 2009 Workshop on Statistical Machine Translation*, in Proceedings of the Fourth Workshop on Statistical Machine Translation, pp 1–28, Athens, Greece

Callison-Burch, C., Osborne, M., and Koehn, P.: 2006, *Re-evaluation the Role of Bleu in Machine Translation Research*, in Proceedings of the European Machine Translation Conference (EAMT'2006), pp 249–256, Trento, Italy

Carpuat, M.: 2009, *Toward Using Morphology in French-English Phrase-Based SMT*, in Proceedings of the Fourth Workshop on Statistical Machine Translation, pp 150–154, Athens, Greece

Chen, S. F.: 1996, *Building Probabilistic Models for Natural Language*, Ph.D. thesis, Harvard University

Chiang, D.: 2005, *A hierarchical phrase-based model for statistical machine translation*, in Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL'2005), pp 263–270, Ann Arbor, MI, USA

Chiang, D.: 2007, *Hierarchical Phrase-Based Translation*, Computational Linguistics **33(2)**, 201–228

Chung, T. and Gildea, D.: 2009, *Unsupervised Tokenization for Machine Translation*, in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'2009), pp 718–726, Singapore

Creutz, M. and Lagus, K.: 2002, *Unsupervised Discovery of Morphemes*, in Proceedings of the ACL Workshop on Morphological and Phonological Learning, pp 21–30, Philadelphia, PA, USA

Creutz, M. and Lagus, K.: 2005, *Inducing the Morphological Lexicon of a Natural Language from Unannotated Text*, in Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'2005), Espoo, Finland

Creutz, M. and Lagus, K.: 2007, *Unsupervised models for Morpheme segmentation and morphology learning*, ACM Transactions on Speech and Language Processing 4(1)

Danielsson, P. and Ridings, D.: 1997, *Practical Presentation of a "Vanilla" Aligner*, in Proceedings of the TELRI Workshop on Alignment and Exploitation of Multilingual Texts, Ljubljana, Slovenia

de Gispert, A., Virpioja, S., Kurimo, M., and Byrne, W.: 2009, *Minimum Bayes Risk Combination of Translation Hypotheses from Alternative Morphological Decompositions*, in Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'2009), pp 73–76, Boulder, Colorado

DeNero, J. and Klein, D.: 2007, *Tailoring word alignments to syntactic machine translation*, in Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'2007), p. 17, Prague, Czech Republic

Fishel, M.: 2009, *Deeper than Words: Morph-based Alignment for Statistical Machine Translation*, in Proceedings of the Conference of the Pacific Association for Computational Linguistics (PacLing'2009), Sapporo, Japan

Fishel, M.: 2010, *Simpler is better: Re-evaluation of Default Word Alignment Models in Statistical MT*, in Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation (PACLIC'2010), p. (in print), Sendai, Japan

Fishel, M. and Kaalep, H.-J.: 2008, *Experiments on Processing Overlapping Parallel Corpora*, in Proceedings of the International Conference on Language Resources and Evaluation (LREC'2008), Marrakech, Morocco

Fishel, M. and Kaalep, H.-J.: 2010, *CorporAl: a Method and Tool for Handling Overlapping Parallel Corpora*, The Prague Bulletin of Mathematical Linguistics **94**, 67–76

Fishel, M., Kaalep, H.-J., and Muischnek, K.: 2007, *Estonian-English Statistical Machine Translations: the First Results*, in Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA'2007), pp 278–283, Tartu, Estonia

Fishel, M. and Kirik, H.: 2010, *Linguistically Motivated Unsupervised Segmentation for Machine Translation*, in Proceedings of the International Conference on Language Resources and Evaluation (LREC'2010), pp 1741–1745, Valletta, Malta

Fraser, A. and Marcu, D.: 2007, *Measuring word alignment quality for statistical machine translation*, Computational Linguistics **33(3)**, 293–303

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

Galley, M. and Manning, C. D.: 2008, *A Simple and Effective Hierarchical Phrase Reordering Model*, in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'2008), pp 848–856, Honolulu, Hawaii

Ganchev, K., Graça, J. V., and Taskar, B.: 2008, *Better alignments = better translations?*, in Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'2008), pp 986–993, Columbus, OH, USA

Goldsmith, J.: 2001, *Unsupervised learning of the morphology of a natural language*, Computational Linguistics **27(2)**, 153–198

Guzman, F., Gao, Q., and Vogel, S.: 2009, *Reassessment of the Role of Phrase Extraction in PBSMT*, in Proceedings of Machine Translation Summit XII, Ottawa, Canada

Habash, N. and Sadat, F.: 2006, *Arabic Preprocessing Schemes for Statistical Machine Translation*, in Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'2006), pp 49–52, New York City, USA

Hammarström, H.: 2009, *Unsupervised Learning of Morphology and the Languages of the World*, Ph.D. thesis, Chalmers University of Technology

He, X.: 2007, *Using Word-Dependent Transition Models in HMM-Based Word Alignment for Statistical Machine Translation*, in Proceedings of the Second Workshop on Statistical Machine Translation, pp 80–87, Prague, Czech Republic

Huang, C.-C., Chen, W.-T., and Chang, J.: 2008, *Bilingual Segmentation for Alignment and Translation*, in Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics (CICLing'2008), pp 445–453, Haifa, Israel

Jelinek, F.: 1976, *Speech Recognition by Statistical Methods*, Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) **64**, 532–556

Kaalep, H.-J. and Veskis, K.: 2007, *Comparing Parallel Corpora and Evaluating their Quality*, in Proceedings of Machine Translation Summit XI, pp 275–279, Copenhagen, Denmark

Karageorgakis, P., Potamianos, A., and Klasinas, I.: 2005, *Towards incorporating language morphology into statistical machine translation systems*, in Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding, pp 80–85, Cancun, Mexico

Kathol, A. and Zheng, J.: 2008, *Strategies for building a Farsi-English SMT system from limited resources*, in Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH'2008), pp 2731–2734, Brisbane, Australia

Kirik, H. and Fishel, M.: 2008, *Modelling Linguistic Phenomena with Unsupervised Morphology for Improving Statistical Machine Translation*, in Proceedings of the Swedish Language Technology Conference Workshop on Unsupervised Methods in NLP, Stockholm, Sweden

Kneser, R. and Ney, H.: 1995, *Improved backing-off for m-gram language modeling*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'1995), Vol. 1, pp 181–184, Detroit, MI, USA

Koehn, P.: 2005, *Europarl: A Parallel Corpus for Statistical Machine Translation*, in Proceedings of Machine Translation Summit X, pp 79–86, Phuket, Thailand

Koehn, P.: 2010, *Statistical Machine Translation*, Cambridge University Press

Koehn, P. and Hoang, H.: 2007, *Factored Translation Models*, in Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'2007), pp 868–876, Prague, Czech Republic

Koehn, P., Hoang, H., Birch, A., Callison-Burch, 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 45th Annual Meeting of the Association for Computational Linguistics (ACL'2007), pp 177–180, Prague, Czech Republic

Koehn, P. and Knight, K.: 2003, *Empirical Methods for Compound Splitting*, in Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'2003), pp 187–194, Budapest, Hungary

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

Koster, J.: 2005, *Is linguistics a natural science?*, Organizing grammar: studies in honor of Henk van Riemsdijk p. 350

Kurimo, M., Virpioja, S., Turunen, V., and Lagus, K.: 2010, *Morpho Challenge 2005-2010: Evaluations and Results*, in Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, pp 87–95, Uppsala, Sweden

Lambert, P., Ma, Y., Ozdowska, S., and Way, A.: 2009, *Tracking Relevant Alignment Characteristics for Machine Translation*, in Proceedings of Machine Translation Summit XII, pp 268–275, Ottawa, Canada

Lee, Y.-S.: 2004, *Morphological Analysis for Statistical Machine Translation*, in Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'2004), pp 57–60, Boston, MA, USA

Li, Z., Callison-Burch, C., Dyer, C., Khudanpur, S., Schwartz, L., Thornton, W., Weese, J., and Zaidan, O.: 2009, *Joshua: An Open Source Toolkit for Parsing-Based Machine Translation*, in Proceedings of the Fourth Workshop on Statistical Machine Translation, pp 135–139, Athens, Greece

Liang, P., Taskar, B., and Klein, D.: 2006, *Alignment by agreement*, in Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'2006), pp 104–111, New York, USA

Liu, F., Jin, Q., Zhao, J., and Xu, B.: 2004, *Bilingual chunk alignment based on interactional matching and probabilistic latent semantic indexing*, in Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP'2004), pp 416–425, Haiman Island, China

Lopez, A.: 2008, *Statistical Machine Translation*, ACM Computing Surveys **40(3)**, 1–49

Lopez, A. and Resnik, P.: 2006, *Word-based alignment, phrase-based translation: what's the link?*, in Proceedings of 7th Biennieal Conference of the Association for Machine Translation in the Americas (AMTA'2006), pp 90–99

Ma, Y. and Way, A.: 2009, *Bilingually Motivated Domain-Adapted Word Segmentation for Statistical Machine Translation*, in Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL'2009), pp 549–557, Athens, Greece

Marcu, D. and Wong, D.: 2002, *A Phrase-Based, Joint Probability Model for Statistical Machine Translation*, in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'2002), pp 133–139, Philadelphia, PA, USA

Mermer, C. and Akın, A. A.: 2010, *Unsupervised Search for the Optimal Segmentation for Statistical Machine Translation*, in Proceedings of the ACL 2010 Student Research Workshop, pp 31–36, Uppsala, Sweden

Minkov, E., Toutanova, K., and Suzuki, H.: 2007, *Generating Complex Morphology for Machine Translation*, in Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'2007), pp 128–135, Prague, Czech Republic

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

Nguyen, T., Vogel, S., and Smith, N. A.: 2010, *Nonparametric Word Segmentation for Machine Translation*, in Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), pp 815–823, Beijing, China

Nießen, S. and Ney, H.: 2004, *Statistical machine translation with scarce resources using morpho-syntactic information*, Computational Linguistics **30(2)**, 181–204

NIST: 2002, Automatic evaluation of machine translation quality using n-gram co-occurrence statistics, Technical report

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'2003), pp 160–167, Sapporo, Japan

Och, F. J. and Ney, H.: 2000, *A comparison of alignment models for statistical machine translation*, in Proceedings of the 18th International Conference on Computational Linguistics (COLING'2000), pp 1086–1090, Saarbrücken, Germany

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'2002), pp 295–302, Philadelphia, PA, USA

Och, F. J. and Ney, H.: 2003, *A Systematic Comparison of Various Statistical Alignment Models*, Computational Linguistics **29(1)**, 19–51

Och, F. J. and Ney, H.: 2004, *The Alignment Template Approach to Statistical Machine Translation*, Computational Linguistics **30(4)**, 417–449

Oflazer, K. and Durgar El-Kahlout, I.: 2007, *Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation*, in Proceedings of the Second Workshop on Statistical Machine Translation, pp 25–32, Prague, Czech Republic

Papieni, K., Roukos, S., Ward, T., and Zhu, W.-J.: 2001, *BLEU: a method for automatic evaluation of machine translation*, in Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'2001), pp 311–318, Philadelphia, PA, USA

Popovic, M. and Ney, H.: 2004, *Towards the use of word stems and suffixes for statistical machine translation*, in Proceedings of the International Conference on Language Resources and Evaluation (LREC'2004), pp 1585–1588, Lisbon, Portugal

Popović, M., Stein, D., and Ney, H.: 2006, *Statistical Machine Translation of German Compound Words*, in Proceedings of the 5th International Conference on Natural Language Processing (FinTAL'2006), pp 616–624, Turku, Finland

Resnik, P. and Smith, N. A.: 2003, *The Web as a Parallel Corpus*, Computational Linguistics **29(3)**, 349–380

Riezler, S. and Maxwell, J. T.: 2005, *On Some Pitfalls in Automatic Evaluation and Significance Testing for MT*, in Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp 57–64, Ann Arbor, MI, USA

Sereewattana, S.: 2003, *Unsupervised segmentation for statistical machine translation*, M.Sc. thesis, University of Edinburgh

Shibatani, M. and Bynon, T.: 1999, *Approaches to language typology*, Oxford University Press

Snyder, B. and Barzilay, R.: 2008, *Unsupervised Multilingual Learning for Morphological Segmentation*, in Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'2008), pp 737–745, Columbus, Ohio

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., and Varga, D.: 2006, *The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages*, in Proceedings of the International Conference on Language Resources and Evaluation (LREC'2006), pp 2142–2147, Genoa, Italy

Stolcke, A.: 2002, *SRILM – an extensible language modeling toolkit*, in Proceedings of the International Conference on Spoken Language Processing (ICSLP'2002), Vol. 2, pp 901–904, Denver, CO, USA

Stymne, S. and Holmqvist, M.: 2008, *Processing of Swedish compounds for phrase-based statistical machine translation*, in Proceedings of the European Machine Translation Conference (EAMT'2008), pp 180–189, Hamburg, Germany

Stymne, S., Holmqvist, M., and Ahrenberg, L.: 2008, *Effects of Morphological Analysis in Translation between German and English*, in Proceedings of the Third Workshop on Statistical Machine Translation, pp 135–138, Columbus, Ohio

Talbot, D. and Osborne, M.: 2006, *Modelling Lexical Redundancy for Machine Translation*, in Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'2006), pp 969–976, Sydney, Australia

Tiedemann, J.: 2004, *Word to word alignment strategies*, in Proceedings of the 20th International Conference on Computational Linguistics (COLING'2004), pp 212–218, Geneva, Switzerland

Tiedemann, J.: 2009, *News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces*, in Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'2009), pp 237–248, Borovets, Bulgaria

Ueffing, N. and Ney, H.: 2003, *Using POS Information for Statistical Machine Translation into Morphologically Rich Languages*, in Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics (EACL'2003), pp 347–354, Budapest, Hungary

Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., and Nagy, V.: 2005, *Parallel Corpora for Medium Density Languages*, in Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'2005), pp 590–596, Borovets, Bulgaria

Vilar, D., Popovic, M., and Ney, H.: 2006, *AER: Do we need to "improve" our alignments*, in Proceedings of the International Workshop on Spoken Language Translation (IWSLT'2006), pp 205–212

Virpioja, S., Väyrynen, J., Mansikkaniemi, A., and Kurimo, M.: 2010, *Applying Morphological Decompositions to Statistical Machine Translation*, in Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, pp 195–200, Uppsala, Sweden

Virpioja, S., Väyrynen, J. J., Creutz, M., and Sadeniemi, M.: 2007, *Morphology-Aware Statistical Machine Translation Based on Morphs Induced in an Unsupervised Manner*, in Proceedings of Machine Translation Summit XI, pp 491–498, Copenhagen, Denmark

Viterbi, A.: 1967, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE transactions on Information Theory **13(2)**, 260–269

Vogel, S., Ney, H., and Tillmann, C.: 1996, *HMM-based word alignment in statistical translation*, in Proceedings of the 16th International Conference on Computational Linguistics (COLING'1996), pp 836–841, Copenhagen, Denmark

Wang, W., Zhou, M., Huang, J.-X., and Huang, C.: 2002, *Structure alignment using bilingual chunking*, in Proceedings of the 19th International Conference on Computational Linguistics (COLING'2002), pp 1–7, Taipei, Taiwan

Zollmann, A., Venugopal, A., and Vogel, S.: 2006, *Bridging the Inflection Morphology Gap for Arabic Statistical Machine Translation*, in Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'2006), pp 201–204, New York City, USA

Čmejrek, M., Cuřín, J., Havelka, J., Hajič, J., and Kuboň, V.: 2004, *Prague Czech-English Dependecy Treebank: Syntactically Annotated Resources for Machine Translation*, in Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'2004), Lisbon, Portugal

Kokkuvõte (Summary in Estonian)

Statistilise masintõlke optimeerimine sisendi modifitseerimise teel

Statistilise masintõlke ülesanne on modelleerida tõlkimist ühest loomulikust keelest teise, rakendades statistilise õppimise printsiipe. Kasutades treenimishulka (suurt hulka tõlkenäiteid), suudab statistiline masintõlge tuletada üldistatud tõlkemudeli, mille abil on võimalik tõlkida uusi, näidetes mittesisalduvaid laused. Selline lähenemine masintõlkele võimaldab luua uusi tõlkemudeleid, omamata siht- või lähtekeele kohta mingeid teadmisi, ning seega lubab kiiresti ja odavalt arendada uusi tõlkesüsteeme.

Käesoleva dissertatsiooni tulemused põhinevad mitmetel autori poolt läbiviidud eksperimentidel, mille eesmärk on muuta erinevaid tõlkemudeli õppimise samme, selleks et parandada tulemuste erinevaid aspekte. Kõik katsetatud muudatused puudutavad ainult õppeprotsessi algfaasi, mistõttu on väitekirjas kirjeldatud muudatused sõltumatud tõlkemudeli täpsemast tüübist ja implementatsioonist.

Kõiki pakutud ideid hinnatakse log-lineaarse masintõlke raamistikus, kasutades fraasipõhiseid ja parsimispõhiseid tõlkemudeleid. Igas eksperimendis sooritatakse evalveerimine mitmel keelepaaril ja mitmel tõlkimissuunal; seejuures iga eksperiment sisaldab kindlasti eesti-inglise keelepaari ning muud keelepaarid valitakse vastavalt eksperimendi iseloomule. Tõlkekvaliteeti hinnatakse automaatmeetrikate BLEU ja NIST abil.

Dissertatsiooni panused võib jagada kolmeks põhiosaks. Esimene osa on pühendatud kattuvatele paralleelkorpustele – sellistele paralleelkorpustele, mis on loodud kas täielikult või osaliselt samade allikate põhjal. Selliste korpuste kombineerimine lihtsa konkateneerimise teel moonutaks algset andmejaotust. Samas on selliste korpuste kombineerimine keerukas, kuna nende ühisosa ei tarvitse langeda korpuste vahel kokku, s.t. võib sisaldada teises korpuses puuduvaid või erinevalt segmenteerituid lausepaare, ning ka teksti väikesi erinevusi, nagu trükivigade parandusi või erinevalt kodeeritud erisümboleid.

Meie tutvustame uudset meetodit kattuvate paralleelkorpuste töötlemiseks, mis on võimeline arvestama erinevaid segmenteerimise tasemeid ja tekstide väikesi erinevusi. Meetod seisneb kahe korpuse vastavates keeltes osade umbkaudses

võrdlemises ning seejärel võrdluste vastavuse leidmises. Genereeritud vastavust saab kasutada korpuste analüüsiks, nende kvaliteedi uurimiseks ja parandamiseks ning korpuste kombinatsioonide loomiseks. Esimese põhiosa eksperimendid näitavad, et meie meetodi abil loodud kombinatsioonid annavad võrreldes alusmeetodiga üldjuhul parema tõlkekvaliteedi. Saavutatava tõlkekvaliteedi erinevus sõltub enamasti korpuste heterogeensusest ning pigem kombineeritud korpuse suurusest kui tema kvaliteedist.

Edaspidine töö selles valdkonnas seisneb loodud meetodi laiendamises, et see toetaks ka teistsuguste korpuste töötlemist peale paralleelkorpuste.

Dissertatsiooni teine põhiosa käsitleb tõlkimist aglutinatiivsete keelte vahel. Sellistel juhtudel on olnud sagedane lahendus enne tõlkemudeli treenimist ja rakendamist segmenteerida sõnavorme iseseisvateks morfeemideks, kasutades segmentatsiooni leidmiseks kas morfoloogilist analüüsi või juhendamata segmenteerimist. Selles vallas pakume välja segmenteerimise meetodi, mis ühendab endas mõlema lähenemise printsiipe ning modelleerib selliseid morfoloogilisi nähtusi nagu liitsõnad, tüved ja lõpud, kasutades juhendamata segmenteerimist; vastavalt modelleeritavale nähtusele liidetakse osa segmente üheks.

Teise põhiosa eksperimentaalne evalveerimine näitab, et meie pakutud meetod töötab masintõlke kontekstis palju paremini kui lihtsal juhendamata segmenteerimisel põhinev alusmudel. Võrreldes sõnapõhise alusmudeliga annab meie meetod oluliselt kõrgemaid tõlkehindeid väikeste treenimiskorpuste puhul; suuremate korpuste puhul on sõnapõhise alusmudeli väljundi kvaliteet aga kõrgem.

Edaspidises töös on meil plaanis rakendada pakutud segmenteerimise meetodit tundmatute sõnade tõlkimiseks. See võimaldaks kombineerida sõnapõhiste tõlkemudelite stabiilsust ning samal ajal ka oluliselt vähendada tundmatute sõnade arvu. Teine arengusuund on juhendamata masinõppe meetodid, mis tuletavad sõnade segmenteerimist, kasutades paralleelkorpuse mõlemat osa praeguse ühe osa asemel. Selliste meetodite tulemuseks on keele morfoloogia alamjuhtum, mis on optimaalne antud tõlkeülesande seisukohast.

Doktoritöö kolmas põhiosa tegeleb tõlkekvaliteedi tõstmise asemel mudelite keerukuse vähendamisega. Nii fraasi- kui parsimispõhised tõlkemudelid kasutavad õppimise etapis sõnajoondust, mille ülesandeks on leida vastavusi lausepaari sõnade vahel; sealjuures on reeglina just sõnajoonduse genereerimine üks pikemaid etappe tõlkemudelite treenimise protsessis.

Vaikimisi kasutatavad sõnajoondusmudelid on IBM mudelid 1 kuni 5 ning Markovi peitmudelil (HMM) põhinev mudel. Mudeli treenimisel alustatakse kõige lihtsama mudeliga ning seejärel kasutatakse selle leitud parameetreid, et algväärtustada järgmise, eelnevast keerukama mudeli parameetreid. Kõige levinum järjekord on IBM mudel 1, IBM mudel 3, HMM-põhine mudel ning viimasena IBM mudel 4. Meie töö sisuks on nende vaikimisi kasutatavate mudelite vajalikkuse vaidlustamine, eesmärgiga kontrollida, kas neid mudeleid saab asendada nende lihtsamate ja vähem aeganõudvate variantidega.

Tulemused näitavad, et HMM-põhine mudel toob kaasa sama või isegi parema kvaliteediga tõlkeid, võrreldes IBM mudeliga 4, vaatamata sellele, et IBM mudel 4 on arvutuslikult palju keerukam. Lisaks eelnevale leidsime, et kuigi IBM mudeli 2 ühel modifikatsioonil põhinevad tõlkemudelid annavad küll madalama kvaliteediga väljundi, siis enamikel juhtudel pole kvaliteedi vahe suur ja seega pakub IBM mudeli 2 modifikatsioon hea kompromissi tõlkekvaliteedi ja sõnajoondusmudeli ressursinõudlikkuse vahel.

Edaspidine töö selles valdkonnas seisneb väljapakutud sõnajoondusmudelite modifikatsioonide realiseerimises laialt levinud tarkvarapaketi GIZA++ projektis, et teha antud töös väljapakutud võimaluste kasutamine ja katsetamine võimalikuks ka teistele selle valdkonna uurijatele.

Töö tulemustest järeldub, et statistilise masintõlke sisendi modifitseerimise teel on võimalik oluliselt mõjutada tõlkekvaliteeti, ilma et oleks vaja muuta tõlkemudelite funktsionaalsust. Suur osa meie väljatöötatud modifikatsioonidest põhjustasid stabiilset paranemist tõlkekvaliteedis või mudelite arvutuslikus keerukuses. Ülejäänud modifikatsioonide puhul osutus, et nende mõju tõlkekvaliteedile sõltub otseselt keelepaarist ning treenimiskorpuse suurusest ja iseloomust.

Curriculum Vitae

Mark Fišel

Born	February 11, 1983, Tallinn
Citizenship	Estonia
Marital Status	married, 1 son
Address	Institute of Computer Science, Liivi 2, 50409 Tartu
Contacts	(+372) 555 89 310, `fishel@ut.ee`

Education

1991–2001	Tallinn High School of Humanities, secondary education
2001–2002	Tallinn University of Technology, Computer Science
2002–2006	University of Tartu, BSc, Computer Science
2006–2009	Nordic Graduate School of Language Technology, graduate student
start. 2006	University of Tartu, PhD student, Computer Science

Languages

Russian	native
English	fluent
Estonian	fluent
French	fair
German	fair
Latvian	fair
Yiddish	fair

Working Experience

2003–2005	Webmedia Ltd., programmer
2006	Institute of the Estonian Language, programmer
start. 2006	University of Tartu, Institute of Estonian and General Linguistics, specialist
start. 2008	University of Tartu, Institute of Computer Science, programmer

Elulookirjeldus

Mark Fišel

Sündinud	11. veebruar 1983, Tallinn
Kodakondsus	Eesti
Perekonnaseis	abielus, 1 poeg
Aadress	Arvutiteaduse instituut, Liivi 2, 50409 Tartu
Kontakt	(+372) 555 89 310, `fishel@ut.ee`

Haridus

1991–2001	Tallinna humanitaargümnaasium, keskharidus
2001–2002	Tallinna Tehnikaülikool, informaatika
2002–2006	Tartu Ülikool, BSc, informaatika
2006–2009	Põhjamaade keeletehnoloogia kraadiõppekooli doktorant
al. 2006	Tartu Ülikool, doktorant, informaatika

Keelteoskus

Vene	emakeel
Inglise	valdan vabalt
Eesti	valdan vabalt
Prantsuse	suhtlustasemel
Saksa	suhtlustasemel
Läti	suhtlustasemel
Jidiš	suhtlustasemel

Töökogemus

2003–2005	Webmedia Eesti AS, programmeerija
2006	Eesti keele instituut, programmeerija
al. 2006	Tartu Ülikool, eesti- ja üldkeeleteaduse instituut, spetsialist
al. 2008	Tartu Ülikool, arvutiteaduse instituut, programmeerija

DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

1. **Mati Heinloo.** The design of nonhomogeneous spherical vessels, cylindrical tubes and circular discs. Tartu, 1991, 23 p.
2. **Boris Komrakov.** Primitive actions and the Sophus Lie problem. Tartu, 1991, 14 p.
3. **Jaak Heinloo.** Phenomenological (continuum) theory of turbulence. Tartu, 1992, 47 p.
4. **Ants Tauts.** Infinite formulae in intuitionistic logic of higher order. Tartu, 1992, 15 p.
5. **Tarmo Soomere.** Kinetic theory of Rossby waves. Tartu, 1992, 32 p.
6. **Jüri Majak.** Optimization of plastic axisymmetric plates and shells in the case of Von Mises yield condition. Tartu, 1992, 32 p.
7. **Ants Aasma.** Matrix transformations of summability and absolute summability fields of matrix methods. Tartu, 1993, 32 p.
8. **Helle Hein.** Optimization of plastic axisymmetric plates and shells with piece-wise constant thickness. Tartu, 1993, 28 p.
9. **Toomas Kiho.** Study of optimality of iterated Lavrentiev method and its generalizations. Tartu, 1994, 23 p.
10. **Arne Kokk.** Joint spectral theory and extension of non-trivial multiplicative linear functionals. Tartu, 1995, 165 p.
11. **Toomas Lepikult.** Automated calculation of dynamically loaded rigid-plastic structures. Tartu, 1995, 93 p, (in Russian).
12. **Sander Hannus.** Parametrical optimization of the plastic cylindrical shells by taking into account geometrical and physical nonlinearities. Tartu, 1995, 74 p, (in Russian).
13. **Sergei Tupailo.** Hilbert's epsilon-symbol in predicative subsystems of analysis. Tartu, 1996, 134 p.
14. **Enno Saks.** Analysis and optimization of elastic-plastic shafts in torsion. Tartu, 1996, 96 p.
15. **Valdis Laan.** Pullbacks and flatness properties of acts. Tartu, 1999, 90 p.
16. **Märt Põldvere.** Subspaces of Banach spaces having Phelps' uniqueness property. Tartu, 1999, 74 p.
17. **Jelena Ausekle.** Compactness of operators in Lorentz and Orlicz sequence spaces. Tartu, 1999, 72 p.
18. **Krista Fischer.** Structural mean models for analyzing the effect of compliance in clinical trials. Tartu, 1999, 124 p.

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
20. **Jüri Lember.** Consistency of empirical k-centres. Tartu, 1999, 148 p.
21. **Ella Puman.** Optimization of plastic conical shells. Tartu, 2000, 102 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
25. **Maria Zeltser.** Investigation of double sequence spaces by soft and hard analitical methods. Tartu, 2001, 154 p.
26. **Ernst Tungel.** Optimization of plastic spherical shells. Tartu, 2001, 90 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 p.
28. **Rainis Haller.** *M(r,s)*-inequalities. Tartu, 2002, 78 p.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
30. **Eno Tõnisson.** Solving of expession manipulation exercises in computer algebra systems. Tartu, 2002, 92 p.
31. **Mart Abel.** Structure of Gelfand-Mazur algebras. Tartu, 2003. 94 p.
32. **Vladimir Kuchmei.** Affine completeness of some ockham algebras. Tartu, 2003. 100 p.
33. **Olga Dunajeva.** Asymptotic matrix methods in statistical inference problems. Tartu 2003. 78 p.
34. **Mare Tarang.** Stability of the spline collocation method for volterra integro-differential equations. Tartu 2004. 90 p.
35. **Tatjana Nahtman.** Permutation invariance and reparameterizations in linear models. Tartu 2004. 91 p.
36. **Märt Möls.** Linear mixed models with equivalent predictors. Tartu 2004. 70 p.
37. **Kristiina Hakk.** Approximation methods for weakly singular integral equations with discontinuous coefficients. Tartu 2004, 137 p.
38. **Meelis Käärik.** Fitting sets to probability distributions. Tartu 2005, 90 p.
39. **Inga Parts.** Piecewise polynomial collocation methods for solving weakly singular integro-differential equations. Tartu 2005, 140 p.
40. **Natalia Saealle.** Convergence and summability with speed of functional series. Tartu 2005, 91 p.
41. **Tanel Kaart.** The reliability of linear mixed models in genetic studies. Tartu 2006, 124 p.

42. **Kadre Torn.** Shear and bending response of inelastic structures to dynamic load. Tartu 2006, 142 p.
43. **Kristel Mikkor.** Uniform factorisation for compact subsets of Banach spaces of operators. Tartu 2006, 72 p.
44. **Darja Saveljeva.** Quadratic and cubic spline collocation for Volterra integral equations. Tartu 2006, 117 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
46. **Annely Mürk.** Optimization of inelastic plates with cracks. Tartu 2006. 137 p.
47. **Annemai Raidjõe.** Sequence spaces defined by modulus functions and superposition operators. Tartu 2006, 97 p.
48. **Olga Panova.** Real Gelfand-Mazur algebras. Tartu 2006, 82 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
50. **Margus Pihlak.** Approximation of multivariate distribution functions. Tartu 2007, 82 p.
51. **Ene Käärik.** Handling dropouts in repeated measurements using copulas. Tartu 2007, 99 p.
52. **Artur Sepp.** Affine models in mathematical finance: an analytical approach. Tartu 2007, 147 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
54. **Kaja Sõstra.** Restriction estimator for domains. Tartu 2007, 104 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
57. **Evely Leetma.** Solution of smoothing problems with obstacles. Tartu 2009, 81 p.
58. **Ants Kaasik.** Estimating ruin probabilities in the Cramér-Lundberg model with heavy-tailed claims. Tartu 2009, 139 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
60. **Indrek Zolk.** The commuting bounded approximation property of Banach spaces. Tartu 2010, 107 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.

63. **Marek Kolk.** Piecewise Polynomial Collocation for Volterra Integral Equations with Singularities. Tartu 2010, 134 p.
64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
65. **Larissa Roots.** Free vibrations of stepped cylindrical shells containing cracks. Tartu 2010, 94 p.