Linköping Studies in Science and Technology

Thesis No. 1421

Compound Processing for Phrase-Based Statistical Machine Translation

by

Sara Stymne

Submitted to Linköping Institute of Technology at Linköping University in partial fulfilment of the requirements for degree of Licentiate of Philosophy

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköping 2009

ISBN: 978-91-7393-501-2 ISSN: 0280–7971
Printed in Linköping, Sweden, 2009
by LiU-Tryck

Compound Processing for Phrase-Based Statistical Machine Translation

by

Sara Stymne

December 2009
ISBN 978-91-7393-501-2
Linköping Studies in Science and Technology
Thesis No. 1421
ISSN 0280–7971
LiU–Tek–Lic–2009:29

ABSTRACT

In this thesis I explore how compound processing can be used to improve phrase-based statistical machine translation (PBSMT) between English and German/Swedish. Both German and Swedish generally use closed compounds, which are written as one word without spaces or other indicators of word boundaries. Compounding is both common and productive, which makes it problematic for PBSMT, mainly due to sparse data problems.

The adopted strategy for compound processing is to split compounds into their component parts before training and translation. For translation into Swedish and German the parts are merged after translation. I investigate the effect of different splitting algorithms for translation between English and German, and of different merging algorithms for German. I also apply these methods to a different language pair, English–Swedish. Overall the studies show that compound processing is useful, especially for translation from English into German or Swedish. But there are improvements for translation into English as well, such as a reduction of unknown words.

I show that for translation between English and German different splitting algorithms work best for different translation directions. I also design and evaluate a novel merging algorithm based on part-of-speech matching, which outperforms previous methods for compound merging, showing the need for information that is carried through the translation process, rather than only external knowledge sources such as word lists. Most of the methods for compound processing were originally developed for German. I show that these methods can be applied to Swedish as well, with similar results.

This work has been supported by the Swedish National Graduate School of Language Technology (GSLT) and Santa Anna IT Research Institute.

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Acknowledgements

First and foremost I want to thank my main supervisor Lars Ahrenberg, for his great support of all aspects of this work, but especially for reminding me about the big picture when I got too focused on small details. I also want to thank my secondary supervisor Joakim Nivre who, despite the fact that I only started to work with him during 2009, have given me many valuable comments.

Thanks to all the members of NLPLab, past and present, for creating a nice working environment and for many good discussions at seminars as well as at coffee breaks: Lars Ahrenberg, Nils Dahlbäck, Lars Degerstedt, Jody Foo, Arne Jönsson, Maria Holmqvist, Bertil Lyberg, Jalal Maleki, Magnus Merkel, Annika Silvervarg, and Håkan Sundblad. A special thank you to Maria Holmqvist for introducing me to statistical machine translation research, for many valuable discussions, for fun travel, for co-authoring one of the papers and proofreading the others, but most of all for being a great friend, and making my everyday work a lot more pleasant.

Thanks also to everyone else in the Human Centered Systems division, including Susanna, Fabian, Johan, Amy, Magnus, Ola, Anders, Per, Jiri, and Björn, for making all the time spent there much more enjoyable. Another thank you to the technical and administrative staff at IDA, who helped this thesis come about in different ways.

Thanks to the native German speakers Joe Steinhauer and Uwe Horn for helping me with grammaticality judgements and to Jalal Maleki for proofreading. Another thank you to the anonymous reviewers of the three conference papers in this thesis, your comments were very useful! Thank you also to all the people that have discussed my work in connection with my conference presentations and to everyone who discussed my work at Xerox Research Centre Europe, where I spent the spring of 2009.

I also want to thank Santa Anna, especially Sture Hägglund, and GSLT for financial support. Also thanks to everyone involved in GSLT, both fellow students and supervisors, for creating an inspiring research environment with great courses, seminars, and many invaluable discussions.

Finally a big thank you to my family and friends, who were always there.

Contents

1 Introduction

Translation is the task of transferring an original text, written in a source language, into another language, a target language. In order to translate a sentence properly a human needs knowledge of both languages, to understand the source text, and to be able to produce a well-formed target language text. In addition, knowledge about the subject matter and the intended readers is a prerequisite for a good translation.

Machine translation (MT), automatic translation by computers, is even more of a challenge. To code all this knowledge into a machine would be very hard, and most systems that use that type of approach, rule-based systems, settle on a syntactic analysis, possibly with some semantics, but do not aim at world knowledge. Another type of approach to machine translation is the empirical approach where existing human translations are used as a knowledge source in the translation process. In this thesis the focus will be on statistical MT (SMT), where statistical models are trained automatically from parallel corpora of human translations. The paradigm adopted is phrase-based statistical machine translation (PBSMT), where the translation unit is the phrase, a contiguous sequence of words.

PBSMT has been a successful approach to MT, and it is the dominant approach in current research on MT. PBSMT systems have the advantage of being easy and fast to build as long as there is a suitable parallel corpus, which, however, is not always the case. The core methods are language independent; the models are trained in the same way regardless of which language pair that is treated. This is an advantage when training a new system, but it has the disadvantage of not using any language pair specific knowledge, which could possibly improve the translation process.

The basic PBSMT approach can be extended in various ways. One way is by adding a preprocessing step and possibly a postprocessing step. In these steps, the texts in one or both languages can be transformed, so that they become more similar. Such modules allow the inclusion of language pair specific knowledge. Another way to extend PBSMT is by factored translation, where words are represented by a vector of factors, such as surface form, lemma and part-of-speech.

Compounds in Germanic languages are normally written as one word without spaces or other indicators of word boundaries. They are productive, and novel compounds can be readily formed and understood. This makes them problematic in the context of statistical machine translation, mostly because of sparse data problems, i.e., occurrences of compounds in the translation input that are not known to the system or that have few

Swedish original	Fru Lalumiéres betänkande återspeglar flera **Natoländers** tänkande enligt vilket **snabbinsatsstyrkorna** tämligen snabbt utvecklas till en fullskalig **krigsduglig** armé.
English translation	Mrs Lalumiére's report reflects a number of **natoländers** thinking in which **snabbinsatsstyrkorna** relatively quickly turned into a full-scale **krigsduglig** army.
English reference	Mrs Lalumiére's report reflects the thinking of many **nato countries**, according to which a **rapid reaction force** would very quickly develop into a fully-fledged army **capable of warfare**.

Figure 1.1: Example of a translation from Swedish to English by a baseline SMT system

English original	However, if we wish - and we do, for we consider it absolutely essential - **sea and river ports** to be included in the system of trans-European networks and to have their own system, then we must by necessity establish a hierarchy and a classification list for this system.
Swedish translation	Men, om vi vill - och det gör vi, eftersom vi anser det absolut nödvändigt - **havet och flod hamnar** skall ingå i systemet för transeuropeiska nät och få sitt eget system, då måste vi med nödvändighet upprätta en hierarki och en klassificering för detta system.
Swedish reference	Om vi trots detta vill - vilket vi gör, eftersom vi anser att det är absolut nödvändigt - att också **havs- och flodhamnarna** skall ingå i det transeuropeiska transportnätet och därmed kunna bilda ett system, måste vi införa en hierarki och en gradering.

Figure 1.2: Example of a translation from English to Swedish by a baseline SMT system

occurrences. This can give rise to problems as in Figure 1.1, where several Swedish compounds are left untranslated in the English output, or as in Figure 1.2, where a phrase that should naturally be translated as a coordinated compound in Swedish has been translated as separate words instead.[1]

To handle compounds in statistical machine translation a general split-merge strategy is adopted, where pre- and postprocessing is added to a factored PBSMT system. In the splitting phase, which is performed prior to training, compounds are split into their component parts. The translation system is then trained as usual, but now for translation between English and split German or Swedish. For translation into Swedish or German,

[1] These translations are produced by the baseline PBSMT system in paper 2 of this thesis.

the separated compound parts have to be merged into full compounds in a postprocessing step after translation.

The main research question of this thesis is whether and how PBSMT, extended with pre- and postprocessing and factors, can be improved by compound processing for German and Swedish. An additional goal is that the methods used should be applicable to other compounding languages as well. To achieve the latter goal only relatively simple tools that are available for many languages, such as part-of-speech taggers, are used. In particular, the following research questions are investigated:

- How can compound splitting be performed in order to give good results for PBSMT? Are the same splitting methods suitable for translation in both directions?
- How can compound parts be merged and what information is needed for it to be successful?
- Does the split-merge strategy work as well for Swedish as for German?

1.1 Contributions

This thesis shows how compound processing can be used to improve statistical machine translation. The main focus is on translation from English into the compounding languages German and Swedish, but also the other translation direction is investigated. The main contributions are:

- Extending the compound splitting algorithm of Koehn and Knight (2003) and investigating the consequences for PBSMT between German and English, showing that different versions of the algorithm give best results in the two translation directions.
- Introducing a novel compound merging algorithm based on part-of-speech matching that can merge unseen compounds, while reducing the risk of erroneous merges. I also show that this algorithm is preferable to previous suggestions of compound merging algorithms.
- Showing that for merging to be successful some additional knowledge source, besides simple word forms, is needed in the translation output, such as parts-of-speech or special symbols on compound parts.
- Successfully applying these splitting and merging methods to a new language, Swedish.

1.2 Outline

In chapter 1, a brief introduction to the subject area and contributions of the thesis were given. In chapter 2, I present a background, with a

focus on statistical machine translation and compound processing. Chapter 3 contains a description of the PBSMT system used in the papers, and summarizes the algorithms and results of the papers. Chapter 4 contains a discussion of the findings, a conclusion and some suggestions for future work. Finally there are three included papers:

Paper 1: Sara Stymne. 2008. German compounds in factored statistical machine translation. In Aarne Ranta and Bengt Nordström, editors, *Proceedings of GoTAL – 6th International Conference on Natural Language Processing*, pages 464–475. Gothenburg, Sweden: Springer Verlag, LNCS/LNAI.

In this paper different compound splitting methods are explored for translation between German and English.

Paper 2: Sara Stymne and Maria Holmqvist. 2008. Processing of Swedish compounds for phrase-based statistical machine translation. In *Proceedings of the 12th Annual Conference of the European Association for Machine Translation*, pages 180–189. Hamburg, Germany.

In this paper the same methods as in paper 1 are applied to a new language, Swedish, and in addition the effect of varying markup and normalization for compound parts are explored.

Paper 3: Sara Stymne. 2009. A comparison of merging strategies for translation of German compounds. In *Proceedings of the EACL 2009 Student Research Workshop*, pages 61–69. Athens, Greece.

In this paper the focus is on compound merging for translation from English into German. An evaluation of a number of merging algorithms based on different knowledge sources is performed.

These papers will be referred to as papers 1–3 throughout this thesis.

2 Background

In this chapter an overview of statistical machine translation is presented, with a focus on phrase-based SMT, factored translation, pre- and postprocessing and evaluation methods. In addition I will discuss compounds in German and Swedish, with a focus on how compound processing has been incorporated with SMT. I also describe previous work on splitting and merging compounds.

2.1 Statistical MT

Statistical machine translation is based on statistical models that are trained on a corpus of human translations, a parallel corpus. Traditional statistical MT uses words as the translation unit and is based on the noisy channel model, shown using Bayes' rule in Equation 2.1[1], where we want to find the probability of a target sentence, T, given a source sentence, S. To find the best translation, $\hat{T}$, Equation 2.1 can be re-written as 2.2, where the denominator, $P(S)$, is removed, since the probability of the source sentence is constant. $P(S|T)$, is given by a translation model and $P(T)$ is given by a language model. In addition, to find the best translation a decoder is needed, which given a source sentence, S, produces the most probable target sentence T, or possibly an n-best list of the most probable translations.

$$P(T|S) = \frac{P(S|T) \cdot P(T)}{P(S)} \tag{2.1}$$

$$\hat{T} = \arg\max_{T} P(S|T) \cdot P(T) \tag{2.2}$$

Language model

The language model accounts for the fluency of the translation, it gives a probability for a sequence of words being a likely target sentence. It is common to use n-gram based language models that build on the Markov assumption that the probability for each word can be based on the n previous words. The probability for a sentence is calculated as the product of the probability of each word, given a history of $n-1$ previous words. In a bigram model, where $n = 2$, this means that the probability for each word is only

[1] The language independent notation S for source language and T for target language is used in this thesis. This can be contrasted to the often used notation of E for English and F for French or foreign.

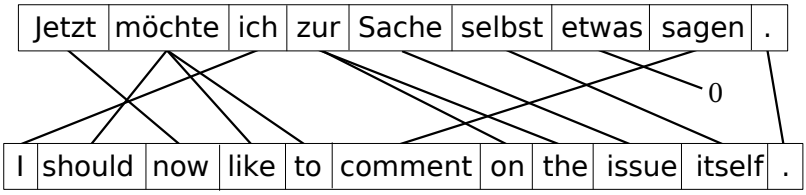

Figure 2.1: Example of a word aligned sentence

conditioned on the previous word, and the probability for the sentence *The old man sleeps.* would be calculated as in Equation 2.3, where BOS and EOS are beginning and end of sentence markers.

$$\begin{aligned} P(\text{The old man sleeps .}) \quad = \quad & P(\text{The}|\text{BOS}) \cdot P(\text{old}|\text{The}) \cdot P(\text{man}|\text{old}) \cdot \\ & P(\text{sleeps}|\text{man}) \cdot P(.|\text{sleeps}) \cdot P(\text{EOS}|.) \end{aligned} \tag{2.3}$$

These probabilities can be estimated from a mono-lingual corpus using maximum-likelihood estimation, as in Equation 2.4 for bigrams, where $C(w_{n-1})$ is the count of word w_{n-1} in a corpus (Manning and Schütze, 1999). Even if an n-gram model is trained on a large amount of data, it will suffer from data sparseness, i.e., many n-grams will have been seen few or no times at all. This is addressed by the use of smoothing techniques, where some of the probability mass of seen events are given to unseen or rare events.

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \tag{2.4}$$

Translation model

The translation model accounts for the adequacy, i.e., how faithful the translation is, of the translation. It is normally estimated from a bilingual corpus. Statistical translation models estimate the conditional probability of a target sentence given a source sentence, using word alignments. In a word aligned text, words that correspond to each other are linked, as shown in Figure 2.1. Some words have no correspondences in the other language, such as *etwas (something)*, which then receives a so called null link. The translation model can be calculated as the sum over all possible alignments, as in Equation 2.5.

$$P(S|T) = \sum_{A} P(S, A|T) \tag{2.5}$$

IBM researchers (Brown et al., 1993) developed a series of five increasingly complex models that estimate translation models and word alignments from sentence-aligned text, called the IBM models. The first model only takes into account the translation of words into other words. In models 2-5 distortion is taken into account as well and in models 3-5 fertility is also added. Distortion is a measure of how target words are reordered, compared to the source. Fertility is a measure of how many target words a single source word is translated into.

The IBM models do not directly estimate the probability in Equation 2.5. A somewhat simplified equation for models 3–5 is shown in Equation 2.6, where i is a position of the target sentence t with length l, j is a position in the source sentence s with length m, a_j is the position of the target word that word j is aligned to, and ϕ_i is the fertility of word i (Elming, 2008). The equation has three parts, the probability n of how many source words a target word translates into, the probability tr, that a source word form translates into a target word form, and the distortion probability d, the probability that a word form appears in a source sentence position, given the link to a target sentence position, and the length of the sentences.

$$P(S, A|T) = \prod_{i=1}^{l} n(\phi_i|t_i) \prod_{j=1}^{m} tr(s_j|t_{a_j}) \prod_{j=1}^{m} d(j|a_j, m, l) \tag{2.6}$$

To estimate these probabilities the expectation-maximization (EM) algorithm (Dempster et al., 1977) is used. The EM algorithm is an iterative method with two steps. In the expectation step alignment frequencies are estimated based on the current model parameters. In the maximization step the model parameters are reestimated based on the alignment frequencies. The EM algorithm is only guaranteed to reach a local maximum, which makes it sensitive to the initial estimation of the model parameters. Therefore, the models are often run in sequence, where the result of the lower models is used to initialize the next model. IBM model 2 is often replaced by a HMM-based model described by Vogel et al. (1996). All these models are assymetric and create one-to-many alignments, i.e., one word in the source text can be aligned to many target words, but each target word can only be aligned to one source word.

Extensions of word-based SMT

In later years the basic word-based models have been extended in a number of ways. Maybe the most common way is phrase-based SMT where not only single words, but phrases, sequences of words, are used as translation units, which will be described in section 2.1.1. Shallow syntax has been included into PBSMT using so called factored translation, which will be described in section 2.1.2. Another possibility is to apply transformations of the corpus

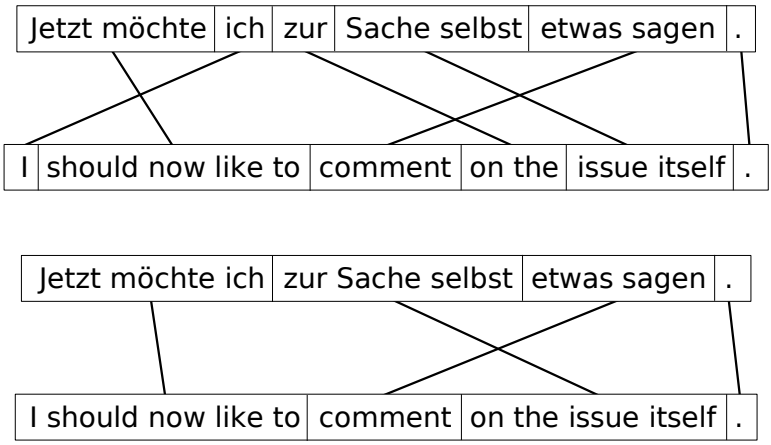

Figure 2.2: Examples of phrase-alignments with different granularity

in a pre- and/or postprocessing step based on some syntactic knowledge, which will be described in section 2.1.3.

Another group of methods that is not investigated in this thesis is hierarchical or syntactic models. In these models syntactic differences can be modeled, which go beyond the power of PBSMT. Syntax can be used either on the source side (Liu et al., 2006), the target side (Yamada and Knight, 2002), or on both sides (Zhang et al., 2007a). Chiang (2005) presented a model where a synchronous context free grammar was induced automatically from plain parallel data. While several of these approaches have shown significant improvements over phrase-based models, their search procedures are more complex, and some methods do not scale well to large training corpora.

2.1.1 Phrase-based SMT

In phrase-based SMT, the unit of translation is not a single word but a phrase. A phrase in this context is a sequence of words, not necessarily a linguistically motivated phrase. Figure 2.2 shows two examples of a phrase-aligned sentence, with different granularity of the phrases.

In PBSMT a log-linear model is commonly used, where the probability $P(T|S)$ is modeled by a set of M feature functions $h_m(T, S)$, where each feature function has a weight λ_m. The best sentence, $\hat{T}$, is computed as in Equation 2.7, where Z_s is a normalization constant. The feature functions include the language model and the translation model.

$$\begin{aligned}\hat{T} &= \arg\max_T P(T|S) \\ &= \arg\max_T \frac{1}{Z_s} exp\left(\sum_{m=1}^{M} \lambda_m h_m(T,S)\right)\end{aligned} \tag{2.7}$$

The language model is normally the same for phrase-based as for word-based translation. The main difference from word-based models is in the translation model, which now includes probabilities for translating phrases, not only single words. An advantage of log-linear models is that it is easy to add other feature functions than just the language and translation models. It is common for instance to add more advanced distortion models, and word and phrase penalties, that can control the length of the output sentence and the tendency to choose long or short phrases.

Translation model

The translation model contains probabilities for phrase translations. A common way to construct a translation model for PBSMT, described in Koehn et al. (2003), is to start with assymetric one-to-many word alignments in both directions, extracted based on the IBM models, which are then symmetrized into many-to-many alignments. From this alignment consistent phrases are extracted and scored. There are other possibilities, such as to estimate phrase probabilities directly from the corpus, not via word alignments (Marcu and Wong, 2002), which has, however, been shown to perform worse than word-alignment-based methods.

Symmetrization normally starts with the intersection of the two unidirectional alignments, and proceeds by adding some links from the union. Och and Ney (2000) described a refined symmetrization method, where they add alignment points from the union if they align at least one unaligned word, and are horizontal or vertical neighbours of an alignment point, or if they connect previously unaligned words. Koehn et al. (2005) described an alternative to this method, grow-diag-final-and, where diagonal neighbours are also allowed, and where unaligned points are added in a final step if they connect two previously unaligned words.

From a symmetrized alignment, Koehn et al. (2003) created a phrase alignment by collecting all phrase pairs that are consistent with the word alignment, that is, the words in a phrase pair can only be aligned with words in the same pair, not to words outside the phrase pair. The probabilities were estimated by relative frequencies, as in Equation 2.8, where $(\bar{s}, \bar{t})$ is a phrase correspondence, an alignment between two phrases.

$$\phi(\bar{s}|\bar{t}) = \frac{count(\bar{s},\bar{t})}{\sum_{\bar{s}} count(\bar{s},\bar{t})} \tag{2.8}$$

Koehn et al. (2003) suggested using lexical weighting besides phrase probabilities. The lexical weighting is a probability that is based on the probabilities of the word alignments between individual words in a phrase pair. Both for phrase probabilities and lexical weighting, it is common to use probabilities for both translation directions, i.e., not only $P(s|t)$, but also $P(t|s)$.

Distortion models

In PBSMT a large part of the local reordering is taken care of within phrase pairs. The phrase pairs can capture local reorderings that were seen in the training data, as in (1) where the German subject follows the verb after an adverbial. These reorderings are only local and cannot be generalized, so there is still a need to model distortion in phrase-based models.

(1) Gestern erlebten wir die Verhaftung ...
Yesterday experienced we the arrest ...
Yesterday, we experienced the arrest ...

It is common to use a distortion penalty, a flat penalty that punishes any deviation from the source order of phrases. The distortion penalty simply adds a factor δ^n for movements over n words. The distortion penalty only takes the position of phrases into account, not the words in them. In addition it is common to impose a constraint, a distortion limit, on the maximum distance a phrase can move. This default distortion model is weak; it discourages distortion, but allows some distortion to take place if it has support from the language model.

A number of alternative distortion models, with a higher degree of discrimination of orderings have been suggested (e.g., Koehn et al., 2005; Al-Onaizan and Papineni, 2006; Kuhn et al., 2006). Koehn et al. (2005) described a lexicalised reordering model, that for each phrase learns how likely it is to follow the previous phrase (monotone), swap places with the previous phrase (swap) or not be connected to the previous phrase (discontinuous). Probabilities are estimated for the three possibile orientations: $P(orientation|S,T)$. This probability can be conditioned on both the source and the target, or only on the source, and the orientation can be based on either the previous or the next phrase. These probabilities can be estimated from an aligned corpus using a smoothed maximum likelihood estimation (Koehn, 2009).

Decoding

The task of finding the translation option that maximizes the log-linear model (Equation 2.7) is exponential on the length of the input sentence. Thus heuristic search techniques like best-first search or stack decoding are normally used to estimate the best translation. The main idea is to use

a priority queue, where partial hypothesis are stored together with their scores, and where the current best hypothesis is expanded at each step. This priority queue can be pruned to a specific size to reduce time and memory complexity at the cost of risking removing partial hypotheses that would be useful in the end.

One example of a search algorithm used for PBSMT is beam search, which is used in the Moses decoder (Koehn et al., 2007). In this algorithm the target sentence is built from left to right, by expanding any source word phrase. The translation hypotheses are stored in beams, where each beam covers a particular number of source words. Each beam can be pruned independently, based on either histogram pruning, where a limit is set on the maximum number of hypothesis in each beam, or by threshold pruning, where hypotheses are cut based on how much worse than the best hypothesis in the beam they are. The hypotheses are scored based on their feature function values for the expanded part, and an estimate of the future cost of expanding the hypothesis fully, based on the translation cost and a simplified language model cost (Koehn, 2009).

Weight optimization

The weights, λ_m, of the log-linear model (Equation 2.7) should reflect the importance given to each of the models. The weights can be optimized on an evaluation metric against a development corpus (see section 2.1.4, for a description of some common metrics). This process is often called tuning. A procedure for performing such optimization is minimum error-rate training (MERT; Och, 2003). It works by translating a set of sentences using some weights, giving an n-best list of translation hypotheses. The feature weights are then recalculated, to produce a good ordering of the n-best list with respect to the translation metric scores. The translation step is repeated with the new weights. These steps are iterated until no new translation hypotheses are found in the translation step.

2.1.2 Factored SMT

In the models discussed so far, each token in the source text is represented by its surface form. In a factored model (Koehn and Hoang, 2007) each token is represented as a vector of features, which can include linguistically motivated features such as lemmas, part-of-speech tags and morphology, as illustrated in Figure 2.3.

In factored PBSMT an additional type of model, a generation model, can be used. The generation model is only used on the target side, to generate surface form from other features, such as lemma and morphology. It can be trained on mono-lingual data. The full translation process is decomposed into one or several translation steps and zero, one, or several generation steps, which is called a decoding path. Factors can also be used in lexicalized distortion models.

the|the|DET|DEF boy|boy|N|SING plays|play|V|3-PRES .|.|PUNC|–
pojken|pojke|N|SG-DEF-UTR leker|leka|V|PRES .|.|PUNC|–

Figure 2.3: An example of an English and Swedish sentence represented with factors for surface form, lemma, part-of-speech and morphology.

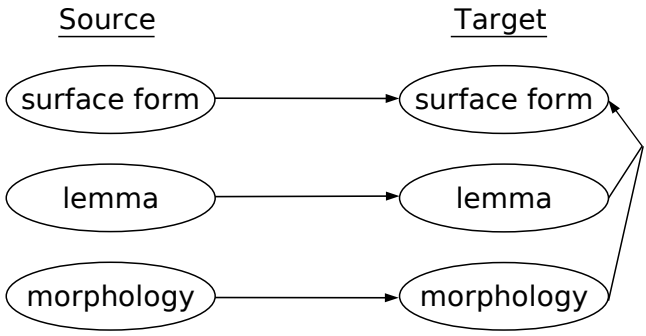

Figure 2.4: Example setup for factored translation

Another feature of the factored translation framework is that it is possible to have multiple alternative decoding paths (Birch et al., 2007). This makes it possible to combine a standard translation model from surface form to surface form, with more complex models including generation steps. Figure 2.4 shows an example of such a setup for factored translation, where there are two decoding paths, from surface form to surface form, and a more complex path with two translation models and one generation model.

Factored translation has been used for a number of language pairs in order to target several problems with standard PBSMT. One way to use factors is to have several factors in the target language, and use other sequence models in addition to the ordinary language model. This can improve word order and agreement. Improvements have been seen by using morphologically enriched part-of-speech tags as an extra output factor for translation into German (Koehn et al., 2008; Stymne et al., 2008), and by using supertags for translation from Dutch to English (Birch et al., 2007). Avramidis and Koehn (2008) use source side factors to model case in translation between English and Greek. A more elaborate model for modelling case in English to Hindi translation is presented by Ramanathan et al. (2009), where lemmas, suffixes, and semantic relations are used on the source side, and a generation model is used on the target side to combine lemmas and suffixes or case markers to surface form. The setup with several decoding paths can also be used for domain adaptation by combining translation models trained on in-domain and out-of-domain corpora (Koehn and Schroeder, 2007).

2.1.3 Pre- and postprocessing for SMT

In almost all PBSMT systems, some pre- and postprocessing is performed. Typically the training data and translation input are tokenized and lowercased. In this case, postprocessing steps are needed where the translation output is detokenized and recased. These steps are commonly performed for most language pairs.

Pre- and postprocessing have been applied to many other phenomena, however, which will be discussed in this section. Examples of pre- and postprocessing that involves compound processing will be described in Section 2.2.3.

Preprocessing

Preprocessing of the bilingual corpora and of the translation input is a strategy that is common for many language specific phenomena. In the preprocessing step the source language can be transformed to become more similar to the target language in some respect. This has often been done to target word order differences between languages, but also for phenomena such as morphology in morphologically complex languages such as Arabic, and German phrasal verbs.

For translation from a morphologically complex language like Arabic to English, the Arabic side has been segmented into morphs in a preprocessing step, to look more like English (El Isbihani et al., 2006; Habash and Sadat, 2006). Nießen and Ney (2000) described work where they performed a number of transformations on the German source side for translation into English. One of the transformations was to join separated verb prefixes, such as *fahre . . . los/losfahren (to leave)* to the verb, since these constructions are usually translated with a single verb in English.

Preprocessing has also been used to transform the word order of the source language. The transformations can be handwritten rules targeting known syntactic differences (Collins et al., 2005; Li et al., 2009), or they can be learnt automatically (Xia and McCord, 2004; Habash, 2007). In these studies the reordering decision was taken deterministically on the source side. This decision can be delayed to decoding time by presenting several reordering options to the decoder, either as a lattice (Zhang et al., 2007b; Niehues and Kolss, 2009), or as an n-best list (Li et al., 2007). Reordering rules can be based on different levels of linguistic annotation, such as part-of-speech (Niehues and Kolss, 2009), chunks (Zhang et al., 2007b) or parse trees (Habash, 2007).

Postprocessing

If preprocessing is performed on the target language prior to training, a postprocessing step of the translation output, where it is transformed back

to standard target language is needed. This has not been investigated as much as preprocessing, but has been applied for instance to morphology.

Virpioja et al. (2007) split words into morphemes, prior to training, for translation between Finnish, Swedish and Danish. They marked all split modifier parts, with a special symbol. In the postprocessing step, every word that was marked with a symbol was merged with the next word. The translation results measured by automatic metrics were worse when splitting and merging was used, than without morphological splitting. However, an error analysis of the result showed other advantages, such as a reduction of untranslated words. No analysis of the merging itself took place. This strategy does have the advantage of being able to merge novel word forms, but has a drawback in that it can merge parts into nonwords if the parts are misplaced in the translation output.

Another study of postprocessing of morphs is El-Kahlout and Oflazer (2006), where translation from English into Turkish was explored. Prior to training, morphs were split and the modifier parts of each word were marked with a symbol and affixes were normalized to base form. In the merging phase, surface forms were generated following morphographemic rules. When the parts were just merged, based on symbols, it gave rise to many illegal forms, and translation results were bad. The reason for this was that the parts were translated out of order. To overcome this to some extent, parts were only merged if the resulting word was accepted by a morphologic analyser, ignoring other, redundant or wrong, morphemes. This constraint improved translation, but it was still worse than the baseline without morphological processing. Grouping some of the split morphemes prior to training, i.e., having a lower number of total splits, improved the system above the baseline.

Another approach for treating morphology is to generate the correct morphological form in translation output where only lemmas are generated (Minkov et al., 2007; Fraser, 2009). Postprocessing has also been used to target word-order phenomena by reordering the translation output based on dependency trees (Na et al., 2009).

There are also postprocessing techniques that do not require any preprocessing. In reranking of n-best lists (Och et al., 2004; Shen et al., 2004) no transformations are performed, but a choice is made between the n best translations produced by the decoder, based on more knowledge than is available in the translation process itself. A postprocessing approach which targets unknown words was suggested by Paul et al. (2009), who applied a transliteration component to words which were unknown to the SMT system.

2.1.4 Evaluation of MT

Evaluation of translations is difficult, since there is not one correct answer, but many possible translations that can convey the meaning of a source text

in an adequate way.

Evaluation can be either human or automatic. In human evaluation translation output is normally judged in some way by human judges, who preferably should be native speakers of the target language. In automatic evaluation the translation hypothesis is generally compared with one or several human reference translations of the same source text.

Human evaluation

One way for humans to evaluate translation output is to judge them on some scales for adequacy and fluency. This, however, has been shown to be hard, with a low annotator agreement, by e.g., Callison-Burch et al. (2007), who suggested ranking the translations from different systems either on sentence or constituent level instead. Other evaluation schemes that have been proposed are for instance assessment of acceptability (Callison-Burch et al., 2008) or usability (Offersgaard et al., 2008). Another possibility is to measure the time or the number of keystrokes or mouse clicks needed by humans to post-edit machine translation output (Jäppinen and Kulikov, 1991).

Another type of human evaluation is to perform an error analysis of the translation output, in addition to the system-wide evaluation. Error analyses can be large scale categorizations of all types of errors that occur. Such a classification is suggested by Vilar et al. (2006), who used it to evaluate Spanish and English translations. The same classification has been used in other studies, e.g., by Avramidis and Koehn (2008) for Greek. Error analysis can also target specific phenomena such as compound translation or noun-phrase agreement (Stymne et al., 2008), Korean verbal heads (Li et al., 2009), or case markers (Ramanathan et al., 2009).

Human evaluation is very time consuming and humans often have a low agreement with other humans (Callison-Burch et al., 2007, 2008). Thus large-scale human evaluation is performed mostly for larger evaluation campaigns, such as the Workshop of Statistical Machine Translation (see e.g., Callison-Burch et al., 2009). Another drawback of human evaluation is that the effort that goes into evaluation is not reusable; if a system is modified, a new human evaluation is needed.

Automatic evaluation

Most of the commonly used automatic metrics work by comparing the translation output to one or more human reference translations, giving some kind of score that quantifies the closeness to it. There are a huge number of automatic metrics, but I will focus on the five metrics that are used in the papers of this thesis, Bleu, Neva, NIST, Meteor and PER, of which all are based on surface matching of words, except for Meteor where stemming and WordNet can be used as well. Other approaches to automatic metrics includes using part-of-speech (Popović and Ney, 2009), syntax (Owczarzak et al., 2007) or

deeper linguistic representations such as semantic roles and discourse representation structures (Giménez and Márquez, 2008). It is also possible to combine several metrics (Giménez and Márquez, 2008) or to use machine learning techniques (Duh, 2008).

Bleu (BiLingual Evaluation Understudy; Papineni et al., 2002) is a metric that measures the precision of n-gram overlap with one or several reference translations, and in addition takes into account the length of the translation hypothesis. Equation 2.9 shows the formula for Bleu, where N is the order of n-grams that are used, usually 4, p_n is a modified n-gram precision, where each n-gram in the reference can be matched by at most one n-gram from the hypothesis. BP is a brevity penalty, which is used to penalize too short translations. It is based on the length of the hypothesis, c, and the reference length, r. If several references are used, there are alternative ways of calculating the reference length, using the closest, average or shortest reference length. Bleu can only be used to give accurate system wide scores, since the geometric mean formulation means it will be zero if there are no overlapping 4-grams, which is often the case in single sentences.

$$Bleu = BP \cdot exp\left(\sum_{n=1}^{N} \frac{1}{n} \log p_n\right) \tag{2.9}$$
$$BP = \begin{cases} 1 & \text{if} \quad c > r \\ e^{(1-r/c)} & \text{if} \quad c \leq r \end{cases}$$

Bleu was the first automatic evaluation metric that was shown to correlate well with human judgements. It has become a de-facto standard for machine translation evaluation, even though later studies have shown that other metrics often have a higher correlation to human judgements (e.g., Callison-Burch et al., 2008).

Neva (N-gram EVAluation; Forsbom, 2003) is a reformulation of Bleu, which allows per-sentence scores, by using the arithmetic mean, and not counting n-grams of a higher order than the sentence length. Equation 2.10 shows the equation for Neva, where the notation and the brevity penalty, BP, is the same as for Bleu and N_{max} is normally 4.

$$NEVA = BP \cdot \sum_{n=1}^{N} \frac{1}{n} p_n \tag{2.10}$$
$$N = \begin{cases} N_{max} & \text{if} \quad c \geq N_{max} \\ c & \text{if} \quad c < N_{max} \end{cases}$$

NIST (Doddington, 2002) was developed to target some of the flaws in Bleu. It is also based on n-gram precision and includes a brevity penalty. However, it does not give equal weight to all n-grams, but less frequent n-grams, which should be more informative, have a higher weight. It also

has a different brevity penalty. The formula for NIST is shown in Equation 2.11, where $C(w_i \dots w_n)$ is the count of the n-gram $w_i \dots w_n$ in the reference translation(s), L_{sys} is the length of the system output and $\bar{L}_{ref}$ is the average length of the references, β is a constant that is set to make the brevity penalty 0.5 when the word ratio between the system output and the reference is 2/3, and the order of n-grams, N, is normally set to 5.

$$NIST = \sum_{n=1}^{N} \left\{ \frac{\sum_{\substack{\text{all } w_1 \dots w_n \\ \text{that co-occur}}} \text{Info}(w_1 \dots w_n)}{\sum_{\substack{\text{all } w_1 \dots w_n \\ \text{in sysoutput}}} (1)} \right\} \cdot BP \tag{2.11}$$

$$BP = exp\left\{ \beta \log^2 \left[\min\left(\frac{L_{sys}}{\bar{L}_{ref}}, 1 \right) \right] \right\}$$

$$\text{Info}(w_1 \dots w_n) = \log_2 \left(\frac{C(w_1 \dots w_{n-1})}{C(w_1 \dots w_n)} \right)$$

Meteor (Metric for Evaluation of Translation with Explicit ORdering; Banerjee and Lavie, 2005) is different from the above metrics in that it includes recall, not only precision, and only considers unigrams. Fluency is captured by a penalty based on the number of contiguous chunks formed by the matched words. The matching of words is flexible where the matching is performed in stages, starting with surface form and allowing additional matching steps for stems, and for WordNet synonyms. Equation 2.12 shows the formula for Meteor, where P is unigram precision and R is unigram recall based on several matching stages, and α, β, γ are weights. In the original version the weights were instantiated as $\alpha = 0.9, \beta = 3, \gamma = 0.5$. In subsequent versions of Meteor these weights have been optimized against human judgments, both on adequacy and fluency (Lavie and Agarwal, 2007) and on ranking of systems (Agarwal and Lavie, 2008). The original Meteor version can be used for any target language using only surface form matching, but WordNet is only available for English, and the stemmer works only for a restricted number of languages. The optimized versions of Meteor are trained for English, German, French and Spanish.

$$\text{Meteor} = F_{\text{mean}} \cdot (1 - \text{Penalty}) \tag{2.12}$$

$$F_{\text{mean}} = \frac{P \cdot R}{\alpha \cdot P + (1 - \alpha) \cdot R}$$

$$\text{Penalty} = \gamma \cdot \left(\frac{\#\text{chunks}}{\#\text{unigrams\_matched}} \right)^{\beta}$$

PER (position independent word error rate; Tillmann et al., 1997) is one of many different error rates, that are used to calculate the distance of a translation suggestion to a reference translation. The matching is based on the Levenshtein distance (Levenshtein, 1966), the number of insertions,

deletions and substitutions needed to transform the hypothesis into the reference. WER (word error rate), is the Levenshtein distance normalized by the reference length. PER is similar to WER, but does not take word order into account. This amounts to comparing the two sentences as bags of words, computing the difference between them, and normalizing by the reference length. One formulation of PER is shown in Equation 2.13 where T_t is the system translation and T_r is the reference sentence (Vogel et al., 2000). Since PER is an error-rate, a lower score is better, and 0 means an identical translation to the reference except for word order.

$$PER = \frac{max(|T_t|, |T_r|) - |T_t \cap T_r|}{|T_r|} \tag{2.13}$$

The main advantage of automatic metrics is that they are cheap and fast to apply, which allows quick testing during system development. They are, however, less informative than human analysis and it is often hard to see exactly what a gain in a metric actually means. Most automatic metrics, including Bleu, are unfair when comparing systems that use different MT architectures, tending to bias in favour of SMT. They are, however, considered useful for incremental development of the same system (Callison-Burch et al., 2006). In each paper of this thesis I use several metrics, to try to give a broader picture of possible improvements, since the different metrics to some extent measure different aspects of translation quality.

2.2 Compounds

Compounds are words that are created by combining at least two free morphemes. German and Swedish, as well as many other languages, for instance Albanian, Arabic, Bulgarian, Dutch, Farsi, Finnish, and Norwegian, generally use so-called closed compounds. Closed compounds are written as single words without spaces or other word boundaries. This can be contrasted to English, where open compounds are generally used, i.e., compound parts are normally written as separate words with a space between them.

In this section I will give an introduction to compounds in German and Swedish, and discuss computational processing of compounds in the context of machine translation.

2.2.1 Compounds in German and Swedish

In both German and Swedish, compounding is very common and productive; new compounds can be readily formed and understood. This is confirmed in a number of corpus studies. In German, compounds have been shown to make up 5-7% of tokens and 43-47% of types in news text (Schiller, 2005; Baroni et al., 2002). If function words are removed, an even higher number of the tokens are compounds; in both Swedish and German 10% of the content

words in a news text have been found to be compounds (Hedlund, 2002). That compounding is productive means that it is likely that a high number of compounds have a very low frequency in texts. Baroni et al. (2002) found that 83% of the compounds in a large German news corpus occur less than five times. In Swedish, compounds are the most common type of hapax words, i.e., words that occur only once in a text (Carlberger et al., 2005).

Some examples of compounds are shown in (2) for German and in (3) for Swedish.[2] Compounds can be binary, i.e., made up of two parts (2a,3a), or have more parts (2b,3b).[3] There are also coordinated compound constructions (2c,3c). In a few cases compounds are written with a hyphen (2d,3d), often when one of the parts is a proper name or an abbreviation.

(2) a. Parlamentsgebäude (*parliament building*)
Parlament+Gebäude (*parliament building*)

b. Menschrechtsverletzungen (*breaches of human rights*)
Mensch+Recht+Verletzungen (*human law breaches*)

c. Struktur- und Kohäsionsfond (*structural and cohesion fund*)
Struktur- und Kohäsion+Fond (*structure and cohesion fund*)

d. EU-Mitgliedstaaten (*EU member states*)
EU-Mitglied+Staaten (*EU member states*)

e. Lehrplan (*curriculum*)
Lehre+Plan (*lesson plan*)

(3) a. medlemsländer (*member states*)
medlemsländer (*member countries*)

b. andrabehandlingsrekommendation (*recommendation for second reading*)
andra+behandling+rekommendation (*second treatment recommendation*)

c. hamn- och lotsavgifter (*port and pilotage dues*)
hamn- och lots+avgifter (*port- and pilot fees*)

d. Tobin-skatt (*Tobin tax*)
Tobin-skatt (*Tobin tax*)

e. klargöra (*clarify*)
klar+göra (*clear make*)

Compounds in one language do not necessarily correspond to a compound in another language. German and Swedish compounds can for instance have English translations that are open compounds (2a,3a), other constructions,

[2] A plus sign, +, will sometimes be used to show the boundary in compounds. The plus sign is not part of the orthography.

[3] Even if a compound have several parts, it can be analysed as a nested binary structure, for (2b) *((Mensch+recht)+verletzungen)*. In PBSMT the representation of words is flat, there is no hierarchy, so this will not be taken into account.

possibly with inserted function words and reordering (2b,3b), or single words (2e,3e).

Compounds are sometimes divided into two types: determinative and copulative (Thorell, 1981). In determinative compounds the last part is the head of the compound, and the other parts are some kind of modifiers of the head, as in (2a), where a *parliament building* is a building used by a parliament. In copulative compounds the parts are coordinated and all parts have the same importance, as in the Swedish *blågul* (*blue and yellow*). Determinative compounds are more common than copulative compounds. In both classes the compound has the same part-of-speech as the last part, and also the same derivational pattern. I will refer to the last part of the compound as the head, and the other parts as modifiers, even for copulative compounds.

Another distinction can be made between occasional and lexicalized compounds (Hedlund, 2002). Occasional compounds can be formed readily, and their meaning is always compositional, i.e., they can be directly understood based on the semantics of the parts, as in (2a). When compounds are used often, they become lexicalized. Lexicalized compounds can be compositional, but it can also happen that their semantics change into a more specific meaning, as in (4).[4] In this type of compound, there is still a relationship between the semantics of the parts and the full compound. There are also compounds that are opaque, where the semantics of the compound cannot be derived from that of its parts, as in (5). Opaque compounds are always lexicalized.

(4) DE Hochhaus *(skyscraper)*
Hoch+Haus *(high house)*
SV höghus *(skyscraper)*
hög+hus *(high house)*

(5) DE Schneebesen *(egg whisk)*
Schnee+Besen *(snow broom)*
SV jordgubbe *(strawberry)*
jord+gubbe *(earth man)*

In German and Swedish both full compounds and their parts can have many different parts-of-speech. Productive compounds can be nouns, adjectives, verbs, and adverbs. There are compounds with other parts-of-speech, such as the German preposition *gegen+über* (*opposite*), but they are not productive. Modifier parts can belong to a larger class of parts-of-speech than the full compounds, also including prepositions, numerals, pronouns and interjections. Table 2.1 gives some examples of possible combinations. Noun compounds are the most common compounds in both languages, with noun+noun compounds being the most common combination, which have

[4] The abbreviation DE is used to indicate German examples and SV to indicate Swedish examples.

Table 2.1: Some part-of-speech combinations in compounds

Type	Examples	
Pron+N	DE	ichform (*first person*) ich+Form (*I form*)
Num+N	SV	femkamp (*pentathlon*) fem+kamp (*five struggle*)
V+V	DE	kennenlernen (*get to know*) kennen+lernen (*know learn*)
Particle+V	SV	utbreda (*spread out*) ut+breda (*out spread*)
N+Adj	DE	zahlreich (*plentiful*) Zahl+reich (*number rich*)
PN+Adj	SV	finlandssvensk (*Finno-Swedish*) Finland+svensk (*Finland Swedish*)
Adj+Adv	DE	grösstenteils (*in most instances*) grössten+teils (*largest partly*)
N+Adv	SV	jättesällan (*very rarely*) jätte+sällan (*giant rarely*)

been found to constitute 62% of the compounds in German news text (Baroni et al., 2002).

2.2.2 Compound morphology

Compound modifiers can have a different morphological form than the base form of the part as a stand-alone word. The head of the compound, on the other hand, can occur in any paradigmatic form, and does not show any changes specific to compounds. The special modifier forms can differ from the base form in that letters are added and/or removed from it. This change has often been called connecting element (De: *Fugenelement*, Sv:*Fogeelement*).[5] This term is criticized by Langer (1998), who argues that modifier forms should be regarded as word forms on the same level as other word forms. His view is shared by Heid et al. (2002) who assume that nouns have three types of stems: simplex, derivational and compounding stems. Langer (1998) suggests the terms compound suffix (De: *Kompositionssuffix*) for the letter changes and compound form (De: *Kompositionsform*) for the combination of the modifier and the compound suffix. I will use these terms.

Langer (1998) divides compound suffixes into four types:

Null operations – the compound form is identical to the base form.

Additions – one or several letters are added to the base form.

Deletions – one or several letters are removed from the base form.

[5] There is no consistent terminology for the morphological changes in modifier parts. Other terms used include linking element, linking suffix, linker, filler, and juncture morpheme.

Table 2.2: Types of compound forms with examples

Type	Examples	
Null operation	DE *0*	umweltfreundlich (*environmentally-friendly*) Umwelt+freundlich (*environment friendly*)
	SV *0*	naturkatastrof (*natural disaster*) natur+katastrof (*nature disaster*)
Addition	DE *+es*	Jahreswechsel (*turn of the year*) Jahr+Wechsel (*year change*)
	SV *+s*	kvalitetstecken (*quality mark*) kvalitet+tecken(*quality sign*)
Deletion	DE *-e*	Lymphreaktion (*lymphatic response*) Lymphe+Reaktion (*lymph response*)
	SV *-a*	flickskola (*girls' school*) flicka+skola (*girl school*)
Combination	DE *-on/+en*	Stadienexperte(*stadium expert*) Stadion+Experte (*stadium expert*)
	SV *-e/+s*	arbetsolycka (*industrial accident*) arbete+olycka (*work accident*)
Umlaut	DE *"+er*	Völkermord(*genocide*) Volk+Mord(*people murder*)
	SV *"-er/+ra*	brödrakärlek (*brotherly love*) broder+kärlek (*brother love*)

Umlaut – either of the above types is combined with umlaut.

I will use a fifth term, **combinations**, where a deletion is combined with an addition. Table 2.2 shows examples of the different types. Umlaut is very uncommon in Swedish, and is not productive. Another view of deletions, presented for instance in Goldsmith and Reutter (1998) and Hellberg (1978), is that the form without the deleted suffix is the stem, so that in the German example of deletion in Table 2.2, the stem would be *Lymph*, not *Lymphe*. This would also mean that combinations would be simple additions. A consequence of this view is that this type of base form will not coincide with words as they are found in a corpus.

Compound forms can coincide with paradigmatic forms, such as genitive and plural in German. Examples of this can be seen in Table 2.2 where *Jahres* is also the genitive form of *Jahr* and *Stadien* is also the plural form of *Stadion*. An alternative analysis would be to analyse these forms as paradigmatic forms rather than as compound forms. Langer (1998) argues against this, since for German nouns, plural and singular forms in compound modifiers do not always correspond to plural and singular semantics. In Swedish the base form tends to be used in modifier parts, rather than inflected forms, as plurals (Thorell, 1981). Compound forms do coincide with genitive in Swedish as well, however. Like Langer (1998) and Rackow et al. (1992) I will adopt the analysis that treats the base forms of words as the default form, and any changes to this in modifier parts as compound

forms, even if they coincide with paradigmatic forms.

Many compound parts have different forms in different compounds, exemplified in (6). Most compound parts tend to have only one or very few possible compound forms, with the null operations being the most common. Which compound form a part should have in a particular compound is very hard to predict. There are no rules, but many tendencies, which means that it is hard to formalize them in an automatic system.

(6) DE *0* Kindphase (*child-caring period*)
Kind+Phase (*child phase*)
+s Kindstod (*cot death*)
Kind+Tod (*child death*)
+es Kindesschutz (*child protection*)
Kind+Schutz (*child protection*)
+er Kinderarbeit (*child labour*)
Kind+Arbeit (*child work*)
SV *0* arvprins (*hereditary prince*)
arv+prins (*heritage prince*)
+s arvsmassa (*genetic stock*)
arv+massa (*heritage mass*)
+e arvegods (*heritage*)
arv+gods (*heritage goods*)

Goldsmith and Reutter (1998) mentioned several factors that influence the choice of compound suffixes for German, namely gender, word-length, phonology, diachrony, and dialectal variety. Kürschner (2003) groups factors that influence choice of compound form for German and Danish into the main categories: semantics, flexion, etymology, derivational patterns and phonology. For Swedish, Thorell (1981) used categories based on declension type. However, even within these categories there are no strict rules, but mainly tendencies of patterns based on factors such as phonology, intelligibility, stylistic level and dialectal influencies. The compound suffix also varies with the number of parts in a compound, for instance, the middle part in a ternary compound is more likely to have an s-addition than the same part in a binary compound for Swedish.

There have been some attempts to create lists of the possible compound forms for different word forms. Hellberg (1978) contains the possible compound suffixes for a number of Swedish nouns. Heid et al. (2002) and Goldsmith and Reutter (1998) both describe methods for automatically collecting an inventory of compound forms for specific nouns based on a raw German corpus. The approach of Heid et al. (2002) requires manual verification.

In some cases concatenating two words would lead to the occurrence of three identical consecutive consonants. In Swedish, there is a spelling rule that does not allow this, and three identical consonants are reduced to two, as in (7^{SV}). I will call this spelling rule the *3-consonant rule*. This spelling

rule was also used for some German compounds before 1996, when it was changed by a spelling reform, so that nowadays three identical consecutive consonants are never reduced to two at compound boundaries in German (Institut für Deutsche Sprache, 1998), as shown in (7[DE]).

(7) SV tullagstiftning *customs legislation*
tull+lagstiftning *customs legislation*

DE Zelllinie *cell line*
Zell+Linie *cell line*

2.2.3 Integrating compound processing and SMT

Compound treatment has been addressed for translation between German and English by several authors. The most common architecture for translation from German is to split compounds in a preprocessing step prior to training and translation, using some automatic method, for instance in Nießen and Ney (2000); Koehn and Knight (2003); Popović et al. (2006); Holmqvist et al. (2007); Stymne et al. (2008); Koehn et al. (2008) for SMT and by Brown (2002) for example-based MT. The German compounds are split into their component parts in a preprocessing step and the translation model is then trained between modified German and English. At translation time, the German source text is also run through a compound splitter.

In the studies cited above, only one splitting option is given as input to the decoder, which can be problematic in case the splitting is wrong, or if any of the parts are unknown. In Dyer et al. (2009) several splitting options were given to the decoder in the form of a lattice. It is, however, not possible to use lattices during training, and in order to solve this, they doubled the training corpus, keeping one part without splits and in the other part they used the best splitting option for each word. Experiments showed that this method is successful for translation from German and Hungarian into English.

For translation into German, Popović et al. (2006) split compounds during training and after translation merged compound parts back into full compounds. They also tried a model where they merged English compounds prior to training instead of splitting German compounds.

Compound splitting has also been used to improve word alignment by splitting compounds prior to word alignment (Popović et al., 2006). After the word alignment step, compounds were merged again, and the alignments were adjusted, before training the phrase-based models. This procedure improved translations compared to the baseline without compound processing in both translation directions, and gave similar results as using splitting and merging in the phrase-based translation model.

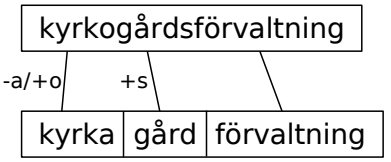

Figure 2.5: Example of how a compound can be split (*church yard administration / cemetery administration*)

2.2.4 Compound splitting

Compound splitting is the task of splitting compounds into their component parts. It has also been called decompounding. Figure 2.5 shows an example of this, also showing the compound suffixes used.

A complication for automatic compound splitting is that compounds can be ambiguous. In some cases, as in (8), several options can result in semantically likely interpretations. In other cases, as in (9), there clearly is one semantically likely interpretation, and others that would be ruled out by a human. For automatic methods, however, these cases can be problematic as well. Ambiguities are often a result of the use of different compound suffixes, such as a possible addition of *+e/+es* in (9$^{\text{DE}}$) or the 3-consonant spelling rule in (9$^{\text{SV}}$). There is also a risk of splitting noncompounds into two parts that happen to constitute two individual words, as in (10).

(8) DE Staubecken (*reservoir* or *dust corner*)
Stau+Becken (*holdup pond*), Staub+Ecken(*dust corner*)

SV bildrulle (*bad driver* or *roll of film*)
bil+drulle (*car maniac*), bild+rulle (*film roll*)

(9) DE Jahrestag (*anniversary*)
Jahr+Tag (*year day*), ?Jahr+Stag (*year hemp rope*)

SV stopplikt (*obligation to stop*)
stopp+plikt (*stop duty*), ?stop+plikt (*stoup duty*),
?stopp+likt (*stop alike*)

(10) SV vante (*glove*)
∗van+te (*accustomed tea*)

DE konsularisch (*consular*)
∗Konsul+arisch (*consul Aryan*)

Compound splitting is addressed in many papers, both as a separate task (Schiller, 2005) and targeted for applications such as information retrieval (Holz and Biemann, 2008), speech recognition (Larson et al., 2000), grammar checking (Sjöbergh and Kann, 2004), text clustering (Rosell, 2003), lex-

icon acquisition (Kokkinakis, 2001), word prediction (Baroni et al., 2002), and machine translation (Koehn and Knight, 2003).

Alfonseca et al. (2008b) summarizes the main strategy generally used for compound splitting in the following steps:

1. For each word, split it in every possible way

2. Calculate a score for each possible splitting option using some weighting function

3. Choose the highest scoring splitting option (which could mean choosing not to split at all, if that has the highest score, or if there are no other splitting options)

The first step is often performed using some kind of word list, and allowing all splitting options where all of the parts are known words. The word list could either be a dictionary, or it could be compiled from a corpus, which tends to give better coverage, especially for specific domains. It is also possible to use special word lists of known compound parts (Sjöbergh and Kann, 2004). In addition to word lists, special attention needs to be given to compound forms, changes to the form of compound parts, and spelling changes (see Section 2.2.2). It is hard to predict where these forms will appear, so a common strategy is to allow them on all modifier parts (Koehn and Knight, 2003). It is possible to constrain the set of splitting options further by imposing different types of constraints, such as limiting the minimum length of compound parts or using part-of-speech constraints (Koehn and Knight, 2003).

There have been many suggestions of how to rank and score the candidate splitting options. For German, Schiller (2005) used a weighted finite state transducer to choose the most likely split based on probabilities of parts being compound modifiers, and preferring a small number of splits. Holz and Biemann (2008) filtered splitting options based on corpus frequencies and the length of parts. Brown (2002) identified German compounds based on the existence of cognates in another language, English. Rackow et al. (1992) described a recursive procedure, where they deterministically choose parts from left to right, based on dictionary lookup. Larson et al. (2000) used a corpus, to calculate how many words that share possible prefixes and suffixes, and split at points where both the suffix and prefix are common.

For Swedish, Brodda (1979) used a rule-based method, based on the observation that consonant combinations at splitting points, such as the sequence *lkk* in (11), are often not found in noncompounds. Another approach based on consonant clusters is described in Kokkinakis (2001). Sjöbergh and Kann (2004) tries a number of features for scoring, including semantic context, component corpus frequencies, syntactic context, part-of-speech, and character n-grams. Their most successful system combines character n-grams with part-of-speech and a couple of ad hoc rules.

(11) mjölkko *(dairy cow)*
mjölk+ko *(milk cow)*

Another strategy is to use supervised machine learning to train a classifier, based on a corpus annotated with compounds. Alfonseca et al. (2008b) trained an SVM classifier, with features including corpus frequencies, mutual information, and anchor point statistics from webpages. Friberg (2007) used memory based learning, and features based on character n-grams.

In most studies of compound splitting, splitting is investigated only for one language, often German. Alfonseca et al. (2008a), however, discuss using the same method for more than one language. They also find that it is possible to use training material from a language other than the one that splitting is performed on, sometimes with better results than for training on the same language. However, many other methods are also largely language independent. The method of Koehn and Knight (2003) is only specific for German with regard to the compound suffixes used. The method of Sjöbergh and Kann (2004), using for instance character n-grams, for Swedish, could probably be applied to other languages with good results, as they pointed out.

Compound splitting for PBSMT

In this thesis I base compound splitting on an empirical compound splitting algorithm developed for statistical machine translation by Koehn and Knight (2003), which I will describe in more detail. The algorithm was developed for German, but is mostly language independent.

Possible splitting options were identified by splitting every word into parts that are known from a monolingual corpus. The known words are restricted to at least three characters in length and the addition of *+s* or *+es* was allowed to occur at all split points. If at least one splitting option was found for a word, they chose the best split (which can be not to split), using three different scoring methods:

- Eager:
  A simple baseline method, where the split with the highest number of parts were chosen. If several best splits were possible, ties were resolved by the frequency-based method below.

- Frequency-based:
  This method used the frequencies of words in the monolingual corpus. The best splitting option, $\hat{S}$, is the option with the highest geometric mean of its n parts p_i of all possible splitting options, S:

$$\hat{S} = \arg\max_{S} \left(\prod_{p_i \in S} count(p_i) \right)^{\frac{1}{n}}$$

- Alignment-based:
  Since the goal of splitting was to improve translation into English, compounds were split in such a way that their parts were aligned to separate English words in a bilingual automatically aligned corpus.

In addition they experimented with constraints based on part-of-speech, by restricting words from the monolingual corpus to content words: nouns, adverbs, adjectives and verbs. They found that the more complex alignment-based method, which was good on a gold standard evaluation, did not improve either word-based or phrase-based SMT. The frequency-based method was best in both cases, and the eager method was good for PBSMT. They did not try the part-of-speech restrictions in combination with either of the two best methods. The evaluation was performed only on NP/PPs, where the number of compounds is higher than in full texts, which makes the results difficult to compare to other studies.

Evaluation of compound splitting

There have been two major approaches to evaluate compound splitting, either direct evaluation by comparing the results to a manually prepared gold standard, or indirect evaluation by evaluating its effects on a task, such as information retrieval, speech recognition or machine translation.

Two types of gold standards have been suggested for compound splitting evaluation. Most common are annotations where all words that are considered compounds are identified. Since compounds are less frequent than noncompounds, weighted texts, with a higher frequency of compounds than normal are often used (Alfonseca et al., 2008a). What should be considered a compound can be hard to distinguish, with borderline cases such as phrasal verbs. The choices made in creating gold standards are, however, often not discussed, which makes a comparison between the results against different gold standards hard.

For this type of gold standard, agreement between different human judges were calculated by Alfonseca et al. (2008a). They reported agreement numbers for compound classification agreement (CCA), i.e., if a word is classified as a compound or not, and for decompounding agreement (DA), i.e., if the judges agree on how to split a compound. In addition they gave kappa scores for CCA. They gave results for five Germanic languages, and for Finnish, and had a high kappa agreement on CCA for all languages. The DA scores were lower than CCA, but still over 81% for all languages. This indicates that compound splitting is relatively simple for humans.

Koehn and Knight (2003) suggested a different type of gold standard, targeted at machine translation, which they call one-to-one correspondence with English, since English is their target language for translation. In this type of gold standard only those compounds are annotated, where each part corresponds to a distinct English content word. As an example, the words in (12) are in one-to-one correspondence with English despite reordering of

content words and insertion of function words, whereas (13) are not in one-to-one correspondence with English since their two parts correspond to one English content word.

(12) DE Medienfreiheit *(freedom of the media)*
Medien+Freiheit *(media freedom)*
SV unionsfördraget *(Treaty of the Union)*
union+fördraget *(union treaty)*

(13) DE Zeitraum *(period)*
Zeit+Raum *(time area)*
SV ändringsförslag *(amendment)*
ändring+förslag*(change suggestion)*

Koehn and Knight (2003) defined a number of categories and metrics, that they used for the evaluation against their gold standard:

correct split: words that were correctly split

correct not: words that should not be split and were not

wrong not: words that should be split but were not

wrong faulty: words that should be split but that were split incorrectly

wrong split: words that should not be split but were

precision: $\frac{\text{(correct split)}}{\text{(correct split+wrong faulty+wrong split)}}$

recall: $\frac{\text{(correct split)}}{\text{(correct split+wrong faulty+wrong not)}}$

accuracy: $\frac{\text{(correct)}}{\text{(correct+wrong)}}$

These categories and metrics are also used by other researchers, e.g., Alfonseca et al. (2008a). But other definitions of these metrics have been used as well, for instance, Sjöbergh and Kann (2004) reports accuracy on a set of ambiguous compounds and Holz and Biemann (2008) computes recall and accuracy on each individual split, not on full words.

Koehn and Knight (2003) discuss the correlation between evaluation on a gold standard compared to the performance on a machine translation task. They find that for phrase-based SMT, splitting methods that perform poorly on the gold standard can give good results on the translation task. Part of the explanation for this is that during phrase-alignment the granularity of the splits is decided, since the statistical methods can effectively rejoin split parts in a phrase pair. The type of errors made by the algorithm can thus be more important than the recall and precision figures.

2.2.5 Compound merging

Compound merging is the task of combining split compound parts into full compounds. It is generally performed when compound splitting has been performed in some previous processing step. The task has also been called recompounding or compound recombination.

Merging of previously split compounds for machine translation is much less explored than compound splitting, partly since translation into English is much more common than translation into a language with closed compounds. Compound merging has also been performed for speech recognition and there are related problems, such as the identification of erroneously split compounds in spell/grammar checkers.

Popović et al. (2006) merged compound parts in a postprocessing step after translation into German. The split parts were not normalized, and did not have any type of markup. They used a method based on word lists. Two lists were extracted from the original German training corpus, one of compound parts, and one of full compounds. For every word in the generated output, they checked if it was a possible compound part, and if it was, it was merged with the next word if it resulted in a compound. There is no evaluation of the merging as a separate process, but using it in combination with splitting resulted in improved translation results. Some limitations of the method are that it cannot merge unseen compounds, and that it does not handle coordinated compound parts. Only binary compounds were merged, but in principle the same method could be used for compounds with more than two parts.

Popović et al. (2006) also tried to merge English compounds prior to training, which they call joining, as an alternative to splitting German compounds. For this they try two methods:

- POS-based joining: English words corresponding to compounds are usually nouns, therefore each consecutive sequence of English nouns was merged into one word.

- Alignment-based joining: Several English words aligned to one German word were considered possible compound parts, and were merged into one word.

Both these methods resulted in an improvement over a baseline without compound processing, but were worse than using splitting and merging of German compounds.

Fraser (2009) merged split German compounds after translation from English, by applying a second PBSMT system trained on German with split compounds and normal German. Again, this method cannot merge novel compounds. The compound merging component is not evaluated in isolation, but in combination with other morphological processing. The combination had a lower Bleu score than his baseline system.

Koehn et al. (2008) discussed treatment of hyphenated compounds for translation into German by splitting at hyphens and treating the hyphen as a separate token, marked by a symbol, that was merged with the surrounding words after translation. The impact on translation results was small.

Compound merging has also been performed for speech recognition. An example of this is Berton et al. (1996) who extended the word graphs output by a German speech recognizer with possible compounds, by combining edges of words during a lexical search. The final hypotheses were then identified from the graph using dynamic programming techniques. Compound merging for speech recognition is a somewhat different problem than for machine translation, however, since the order of parts is not an issue, as compared to PBSMT, where there is no guarantee that the order of the parts in the translation output is correct.

Another somewhat related problem to compound merging, is that of detection of erroneously split compounds in human text, that is faced by grammar checkers. Writing compounds with spaces between parts, as separate words, is a common writing error in Swedish and German. Carlberger et al. (2005) described a system for Swedish that used hand-written rules to identify, among other errors, erroneously split compounds. The rules used part-of-speech and morphological features. On a classified gold standard of writing errors they had a recall of 46% and a precision of 39%, for identifying split compounds, indicating that it is a hard problem to find split compounds in free, unmarked text.

3 Resources, algorithms and results

In this chapter I give a summary of the work described in paper 1–3. I describe the external resources used, and give a more detailed description of the machine translation system setup, than the space in the papers permitted. I also summarize the extensions to the splitting algorithm of Koehn and Knight (2003) that I have investigated, present choices made concerning markup, normalization and part-of-speech of compound parts, and present the main compound merging algorithm proposed in this thesis. Finally I summarize the results of the three papers.

3.1 External tools and resources

A number of external tools and resources were used in this work. The training and running of the MT system used the Moses toolkit (Koehn et al., 2007). In addition language models were trained using the SRILM toolkit (Stolcke, 2002) and word alignments were created using GIZA++ (Och and Ney, 2003). In the preprocessing step part-of-speech taggers are used; for German and English I used TreeTagger (Schmid, 1994) and for Swedish I used the Granska tagger (Carlberger and Kann, 1999). The corpus used in all experiments was the Europarl corpus (Koehn, 2005).

Moses (Koehn et al., 2007) is a toolkit for phrase-based SMT that contains a decoder. In addition Moses contains scripts for creating translation and lexicalised reordering models, and for tuning feature weights. It has support for integration with a number of language model toolkits. Moses allows factored translation (see section 2.1.2). It has support for using factors in the translation and distortion models, in additional language models, and in generation steps on the target side.

SRILM (Stolcke, 2002) is a toolkit for building and applying language models. The toolkit implements several smoothing methods, including the two methods used in the experiments: modified Kneser-Ney (Chen and Goodman, 1999) and Witten-Bell (*Method C* in Witten and Bell, 1991).

GIZA++ (Och and Ney, 2003) is a word-alignment tool that implements IBM model 1-4 (Brown et al., 1993), an HMM-based model that can replace model 2 (Vogel et al., 1996) and parameter smoothing. It produces unidirectional one-to-many alignments between two languages. In the experiments GIZA++ runs 5 iterations each of model 1 and the HMM model, and 3 iterations each of model 3 and 4. All word alignment is performed on surface

forms.

To be able to use part-of-speech as a factor the training texts have to be tagged. For English and German TreeTagger (Schmid, 1994) is used, and for Swedish, the Granska tagger (Carlberger and Kann, 1999). Both taggers are trained using statistical methods; TreeTagger is a probabilistic tagger based on decision trees and the Granska tagger is based on a hidden Markov model. Both taggers give both part-of-speech and lemma for each word. The lemmas are used in the compound splitting algorithm. The Granska tagger also produces morphological analyses, with information such as gender and number for nouns and tense for verbs. The morphology is not used in any of the papers. The Granska tagger is developed for grammar checking, and makes a few tokenisation choices that are not suitable for translation, so the output from it is processed in order to separate time expressions and coordinated compounds.

All experiments are performed on the Europarl corpus[1] (Koehn, 2005), which contains transcriptions of the proceedings of the European Parliament in eleven languages, including English, German and Swedish. Europarl is sentence aligned using the algorithm by Gale and Church (1993). The full Europarl is over 1,000,000 sentences per language pair, but in order to reduce training times of the PBSMT system, I used a smaller partition of Europarl for training. In paper 1 I used 439,513 sentences and in paper 2 and 3 I used 701,157 sentences.

3.2 MT system

In all papers a factored phrase-based SMT system is used. It is trained in the same way in all experiments, except for the amount of training data and the compound processing strategies. The main architecture is illustrated in figure 3.1.

Factored translation is used with one source side factor, surface form, which is translated into two target side factors, surface form and part-of-speech. The part-of-speech output factor is used to improve word order by the use of a part-of-speech sequence model, and as a knowledge source for compound merging. In paper 3 it is also used to uppercase the first letter of German nouns.

A log-linear model is trained using the following feature functions (see Section 2.1.1 for a more thorough description of the methods):

Translation models: contains phrase probabilities and lexical weighting for both translation directions, giving a total of four features:

- phrase translation probability $\varphi(s|t)$
- lexical weighting $lex(s|t)$
- reverse phrase translation probability $\varphi(t|s)$

[1] Version 3 of Europarl was used, released on September 28, 2007.

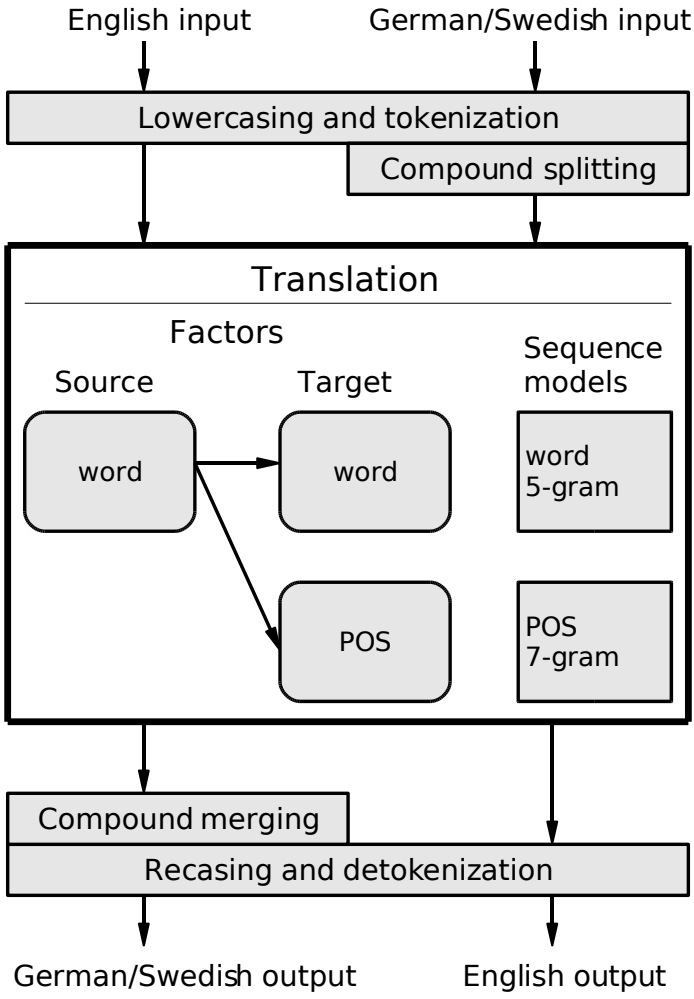

Figure 3.1: The MT system architecture

- reverse lexical weighting $lex(t|s)$

The translation model is trained using the method described in Koehn et al. (2003), where unidirectional word alignments are created by GIZA++ (Och and Ney, 2003) in both directions, which are then symmetrized by the grow-diag-final-and method (Koehn et al., 2005). From this many-to-many alignment, consistent phrases of up to length 7 are extracted.

Distortion models: Two distortion models are used, the standard distance based distortion model, and a lexicalized reordering model (Koehn et al., 2005). The lexicalized reordering model is conditioned on both languages, and has six features, for the three orientations monotone, swap and discontinuous, conditioned on the next or previous phrase.

Sequence models: two sequence models trained on the target side of the bilingual corpus are used:

- A 5-gram language model on surface form, trained using interpolated modified Kneser-Ney smoothing (Chen and Goodman, 1999).
- A 7-gram sequence model on part-of-speech, trained using interpolated Witten-Bell smoothing[2] (Witten and Bell, 1991).

Word penalty: A count of the number of words in the output sentence. This feature is useful to control the length of the output sentence.

Phrase penalty: A count of the number of phrases in the output sentence. This feature controls the tendency to choose longer or shorter phrases.

There are also a number of limitations, in order to make the search problem easier:

- The maximum length of the phrases in the translation model is 7
- The maximum distortion distance is 6
- The maximum beam size during the beam search is set to 200
- Only the 20 most probable translations for each phrase are considered.

To tune the weights, λ, of the log-linear model (see Equation 2.7 on page 9), minimum error-rate training (Och, 2003) is used, as implemented in Moses. The tuning phase is slightly modified compared to the standard algorithm in two ways. It optimizes the Neva metric (Forsbom, 2003), instead of the more commonly used Bleu metric (Papineni et al., 2002). For translation into German and Swedish, compound merging is integrated into the

[2] The more advanced Kneser-Ney smoothing cannot be used when the distribution of counts-of-counts is not strictly decreasing. This assumption is generally met by surface forms, but not by part-of-speech.

tuning phase in papers 2 and 3. This is achieved by applying a compound merging algorithm on the n-best list used in the tuning process.

In the preprocessing step the sentences are first filtered to remove sentence pairs where at least one of the sentences is longer than 40 words. Then the input is detokenized and lower-cased. The standard Moses scripts are used for this, except for an addition of a Swedish abbreviation list, which was created semi-automatically, to aid Swedish tokenization. In the postprocesing step, the reverse detokenization and recasing are performed. For the detokenization the standard Moses script is used. The recasing is performed by training another instance of Moses on the target side of the bilingual corpus, and a lower-cased version of it. In paper 3, German nouns are upper-cased based on the part-of-speech output factor, before this recasing procedure.

The system described this far constitutes the baseline system. In the test systems with compound processing, compounds are split in the preprocessing step. For translation from German and Swedish, compounds are also split in the translation input. For translation into German and Swedish, compounds are merged in the postprocessing step after translation.

3.3 Compound splitting algorithm

The algorithm I use for splitting is based on Koehn and Knight (2003). I re-implemented this algorithm and extended it in a number of ways, introducing more variations, particularly for constraining the splitting options considered. The choices that can be made are:

- Scoring method:
  - Eager: The maximum number of parts is chosen, as in Koehn and Knight (2003), except that ties are broken by preferring the option with the shortest first part(s).
  - Geometric mean of part frequencies: The same as the frequency-based method in Koehn and Knight (2003)
  - Arithmetic mean of part frequencies: The best splitting option, $\hat{S}$, is the option with the highest arithmetic mean of its n parts p_i of all possible splitting options, S:

$$\hat{S} = \arg\max_{S} \left(\frac{1}{n} \cdot \sum_{p_i \in S} count(p_i) \right)$$

    The arithmetic mean always gives an equal or higher value than the geometric mean for positive values, and will thus give a higher number of splits than using the geometric mean.

- Minimum length of words and parts: The minimum length of words to be split and of compound parts can be changed. The main reason for

this is that compounds tend to be long. It also blocks many common errors where short derivational affixes coincide with separate words, such as the German *Ei (egg)* in words like *Schweinerei (rascality)*.

- Number of parts per compound: can be unrestricted, maximum two and maximum two for all parts-of-speech except nouns. The reason for these choices is that noun compounds tend to have many parts to a much higher degree than other types of compounds, and that compounds with several parts are relatively unusual compared to binary compounds.

- Compound suffixes: Three types of compound forms can be handled, additions, deletions and combinations, but umlauts are not handled. The compound suffixes to be used can be specified in a file, which allows easy adaption for new languages. All specified compound suffixes are allowed at all splitting points. This construction also allows hyphens to be treated as a compound suffix, on the same level as for instance the addition of *+s*, which was done in paper 3.

- Constraints based on part-of-speech:

  - Restrict the last part to have the same part-of-speech as the full compound. This can block many erroneous splits, since the last part is the compound head, and always has the same part-of-speech as the full compound.

  - Restrict the words that are to be split to have a certain part-of-speech. Compounds belong to a small number of parts-of-speech, so this could stop making erroneous splits, such as splitting prepositions. In addition it could be desirable not to split proper nouns, since the parts often do not contribute to the semantics of the full word, as in the Swedish surname *Alm+kvist*, whose parts mean *elm* and *branch*.

  - Restrict the words from the monolingual corpus that are to be used for frequency calculations. The modifier parts in a compound also tend to belong to certain parts-of-speech. This class is bigger than the parts-of-speech that can be full compounds; it is for instance possible to have prepositions as compound parts. In this case proper nouns can be useful, to allow compounds such as the Swedish noun *Atlant+kust (Atlantic coast)*.

- Use of lemma: for the frequency calculations from the monolingual corpus, lemmas extracted by the taggers can be used besides using only surface form. The motivation for this is that most modifier parts are in base form, and also that the possible compound suffixes are defined based on the assumption that the modifier parts are in base form.

The alignment-based scoring method of Koehn and Knight (2003) were not reimplemented since it did not result in good results on the translation task in their study.

During the splitting process information is collected that can later be used at merging time. Two different frequency lists are created: one contains all identified compounds and one contains normalized forms of all compound parts, combined with all possible compound forms of that part.

Paper 1 contains a comparison of a number of different settings in the splitting algorithm for German. In paper 2 and 3, one setting was used for splitting, since the focus was on other aspects of compound processing.

3.4 Markup, normalization and part-of-speech

There are several things to consider after compound splitting, which concerns how compound parts should be treated in the translation process. I have considered three aspects: normalization of compound forms, markup of compound parts, and part-of-speech for compound parts.

I have used the assumption that the last part of the compound is the head of the compound, that is, it conveys the main meaning of the compound, and it has the same part-of-speech as the full compound. The other parts, the modifier parts, modifies the meaning of the compound head in some way and need not have the same part of speech as the full compound. Compound suffixes cannot occur for the head, only for the modifier parts.

Compounds are not always compositional, some compound parts have meanings that are different from their standard meaning as stand-alone-words. An example where the meaning is not compositional is shown in (14). A more common case is shown in (15), where one of the parts is ambiguous, and only one of the interpretations will give the correct interpretation for the compound. Compositionality is an import factor for how compound parts should be treated in the translation process.

(14) DE Grundrechte (*basic rights*)
Grund+Rechte (*foundation rights*)
SV huvudprincip (*major principle*)
huvud+princip (*head principle*)

(15) DE Küchenmesser (*kitchen knife*)
Küche+Messer (*kitchen knife/gauger*)
SV affärsbeslut (*business decision*)
affär+beslut (*shop/business decision*)

Markup

I have used three different markup schemes, that I call unmarked, marked and sepmarked. In the unmarked scheme no markup is used, all compound parts are treated as ordinary words. In the marked scheme the modifier

parts are marked with a symbol. In this way the modifier parts are separated from normal words, which is useful for noncompositional parts. The head is not marked, since it is assumed to have a compositional meaning. In the sepmarked scheme there is no marking of the parts, instead an additional token is added between compound parts. For the first part of coordinated compounds, another symbol is used. Examples of the three schemes are shown in (16).[3]

(16) DE Staats- und Regierungschef (*Head of State and Government*)
unmarked: staat und regierung chef
marked: staats-# und regierungs# chef
sepmarked: staat @-@ und regierung @#@ chef

Normalization

Modifier parts can have different compound forms, as shown in Table 2.2 on page 22. These can be left as they are after splitting or they can be normalized to a canonical form. If they are normalized the parts will coincide with words that are not used in compounds, which is good for compositional compounds, but can be problematic for noncompositional compounds. In (16) normalization has been performed in the unmarked and sepmarked scheme, with the consequence that the compound suffix *+s* has been removed. In the marked scheme no normalization is performed, since the parts will not coincide with other words anyway, because of the markup.

Part-of-speech

Part-of-speech is used as an output factor in the translation systems, which means that all tokens need to be marked with a part-of-speech tag. When compounds are split there is thus a need to choose which part-of-speech tags to assign to the compound parts. For the head I always use the same tag as for the full compound. For modifier parts I try two variants: either adding a special part-of-speech tag based on that of the head, or using the part-of-speech tag that was found in the corpus for that word. This is illustrated in (17), where the modifier *bitter*, is marked as the adjective it is in the sepmarked scheme, but as a part of a noun compound (*N-PART*), in the other schemes. The special compound part-of-speech, where parts are marked after the head, can be used to restrict which parts that should be merged after translation.

(17) SV bittermandel|N (*bitter almond*)
unmarked: bitter|N-PART mandel|N
marked: bitter#|N-PART mandel|N
sepmarked: bitter|ADJ @#@|COMP mandel|N

[3] All examples of translation input and output are lower-cased, since lowercasing is performed before and recasing is performed after the compound processing.

Combinations

All together there are 12 possible combinations of markup, normalization and part-of-speech tags. All of these have, however, not been used, only one combination with each type of markup has been explored:

- Unmarked, normalized, special POS-tags
- Marked, non-normalized, special POS-tags
- Sepmarked, normalized, ordinary POS-tags

Examples of the three schemes can be seen in (18).

(18) DE Tageszeitung|N (*daily newspaper*)
unmarked: tag|N-PART zeitung|N
marked: tages#|N-PART zeitung|N
sepmarked: tag|N @#@|COMP zeitung|N

In the marked case, compound parts are separated from normal words by a symbol, so there is no need to normalize them, or to use ordinary part-of-speech, since they would not coincide with other words anyway. In the unmarked and sepmarked case, parts are not marked, and are normalized to coincide with other words. Special POS-tags are used for the unmarked system, in order to separate compound parts from other words in some way. In the sepmarked system, normal POS-tags are used, since it is possible to identify compound parts based on the symbol token. It is possible that other combinations could be useful as well, but this has not been explored in this thesis.

3.5 Compound merging algorithm

For translation into a language with closed compounds, some kind of merging strategy is needed after the translation step if compounds were split during training. The merging step has two main tasks: to identify which words that should be merged into compounds, which is complicated by the fact that the translation process is not guaranteed to produce translations where compound parts are kept together, and to choose the correct form of the compound parts.

Table 3.1 shows examples of possible merging scenarios, and the result after the merging process. There are two main scenarios, either the parts are placed in an order where they lead to a likely good compound, or they are placed in an incorrect order, in which case they should not be merged. Even if the parts are placed in an order which seems good according to the part-of-speech sequence, merging them can lead to a nonexistent word, as in the last example in Table 3.1.

Table 3.1: Merging scenarios (with German examples)

Type	Example input	Result
Correct		
Binary marked	zwischen#|a-part staatliche|adj	zwischenstaatliche
Binary unmarked	forschung|n-part rat|n	Forschungsrat
Binary sepmarked	gesicht|n @#@|comp punkt|n	Gesichtspunkt
Ternary	mit#|n-part glied#|n-part staaten|n	Mitgliedstaaten
Coordinated	polizei-#|n-part und|kon zoll#|n-part behörden|n	Polizei- und Zollbehörden
Erroneous		
Mis-matching POS	schiffs#|n-part in|appr	Schiffs in
Bad compound	bio#|n-part nabe#|n-part fällen|n	Bionabefällen

The main merging algorithm suggested in this thesis is based on part-of-speech matching, and will be called the POS-matching algorithm. This algorithm is applicable for the two markup schemes that have special POS-tags, the marked and unmarked scheme. For the sepmarked scheme, an alternative method based on symbols was used. In addition a method based on word lists is explored in paper 3.

POS-matching algorithm

The POS-matching algorithm uses the fact that it is possible to have several output factors beside surface form in a factored translation system. It merges parts that are marked with the special part-of-speech tag used for compound parts, if the next part-of-speech is matching. As described in section 3.4, the part-of-speech of a compound modifier part is based on the part-of-speech of its head word, so a word is considered matching, if the next word is a compound part of the same type, or a head with a matching part-of-speech. In addition, if the next part does not match, the part could be part of a coordinated compound, which is checked by seeing if the next word is a conjunction,[4] in which case a hyphen is added to the part.

If a compound part is followed by anything other than a matching part-of-speech or a conjunction it is most likely misplaced after the translation process. These compound parts are left as they are in the translation output, which is often fine, since only compound parts that occur as separate words in a corpus are split, which means that the parts often work as stand-alone words.

[4] In paper 1 all conjunctions were allowed. However, an error analysis showed that this lead to some errors, so in paper 2 the allowed conjunctions for Swedish, were restricted to *och (and), eller (or), respektive (respectively), samt (and), som (as well as)* and in paper 3, for German, to only *und (and)*.

When two matching compound parts are merged, the process is iterated to see if the next word is either a matching compound part, head or conjunction. This allows compounds with an arbitrary number of parts to be merged, and coordinated compounds with a first part with several components.

For compound parts that were normalized in the training data, i.e., the special compound forms were changed into base forms, the reverse process, reverse normalization, is needed in order to recreate the correct form for the specific compound. In this process the two frequency lists of compounds and compound forms that were created at split time are used. To find the correct form of a word I first try all combinations of forms of each compound part and check if the result is a word that is known from the corpus. If any known words are found I choose the most frequent one. Else, the parts are added from left to right choosing the most frequent possible combination at each merging point, or if no known combination exists, choosing the most frequent compound form for each part. For Swedish, it is also necessary to take the 3-consonant rule into account, by removing a consonant if a merge results in three identical consecutive consonants.

In summary, the merging algorithm has the following steps:

- Step through each word+POS pair from left to right[5]
  - If a compound-POS, X-PART, is found:
    - Remove markup of the part if present
    - Store the compound part
    - While the next POS is a matching part, X-PART:
      - Remove markup of the part if present
      - Store the compound part
    - If the next POS is a matching head, X:
      - Store the compound head
    - If at least two parts have been found (either several modifiers or a head):
      - Perform reverse normalization on the stored parts if parts are normalized
      - Merge all parts
      - For Swedish: remove a consonant if any of the merges resulted in three identical consecutive consonant
    - If the next POS is a conjunction and no head was found:
      - Add a hyphen at the end of the compound part

The POS-merging algorithm can handle all merging scenarios in Table 3.1 except the last case, where the part-of-speech tags are matching, but it nevertheless produces an erroneous compound. It can, however, not be

[5]The words that are processed in the inner if-clause are skipped in the outer loop.

used for the sepmarked markup scheme, where no special compound parts-of-speech are available.

Alternative merging algorithms

In addition to the POS-matching algorithm I have implemented two other algorithms based on previous research. Paper 3 contains a comparison of some varieties of these algorithms and the POS-matching algorithm.

The symbol-based method is based on work on morphology merging (El-Kahlout and Oflazer, 2006; Virpioja et al., 2007). It merges words that are marked with a symbol with the next word in the marked scheme. For the unmarked scheme, it is based on the part-of-speech tags, without matching. In the sepmarked scheme, when a standard symbol is found, the words on both sides of it are merged. If the symbol for coordinated compounds is found, a hyphen is added to the word before it. In the unmarked and sepmarked schemes, reverse normalization takes place as well. This algorithm has the disadvantage, compared to the POS-matching algorithm, that it is more likely to merge words into noncompounds, since no matching check is carried out.

I also implemented a method based on word lists, inspired by compound merging in Popović et al. (2006). This method is based only on external knowledge sources, namely frequency word lists compiled at split time. Three types of lists were used, lists of compound parts, of compounds and of words. If a compound part is encountered, it is checked if merging it with the next word results in either another compound part, or a compound or word. This is performed recursively, to allow compounds with several parts. Again, reverse normalization is performed when needed. In this scheme no novel compounds can be formed, and it does not handle coordinated compounds. It does not merge words into noncompounds, but there is another risk, that of merging words that should be separate in a specific context, but that happen to form a valid compound when combined, such as those in (19).

(19) DE beider (*both*)
bei der (*at the*)
SV sjukdom (*disease*)
sjuk dom (*absurd judgement*)

In paper 3, the algorithms described above, based on POS-matching, symbols, and word lists, were also extended by combining them, or adding some constraints to them. The word list based method was varied either by only merging words into compounds, or by merging them into all known words from the corpus. It was also combined with the symbol method. Both the symbol and word list method were constrained by only allowing content word part-of-speech on the head word, which blocks some erroneous merges such as that in (19$^{\text{DE}}$). The POS-matching algorithm was implemented both with and without treatment of coordinated compounds.

3.6 Result summary

In this section I summarize the results of the three papers.

3.6.1 Paper 1

Sara Stymne: German Compounds in Factored Statistical Machine Translation
In paper 1 I explored different splitting algorithms for translation between German and English. In this study I only used the marked markup scheme. I varied the splitting algorithms on different aspects such as limiting the minimum length of parts, the number of allowed compound suffixes, and the number of parts per compound.

A gold standard evaluation was performed on one-to-one correspondence with English, which showed a lot of variation between the algorithms. The recall, for instance, varied between 24.9%–76.9%. As in previous studies, however, the results on this gold standard evaluation were not a good indicator of the usefulness of a splitting strategy for PBSMT.

In both translation directions splitting improved the translation results on a majority of the three metrics used. The improvements were larger for translation into German than for translation into English. For translation into English there was a large reduction of the number of unknown words, which is clearly positive. Some marked compound parts were unknown though, showing a drawback of the marked scheme. I also found that different algorithms performed best in the two different translation directions. Generally a larger number of splits was better when translating into German, and a smaller number of splits better when translating into English.

3.6.2 Paper 2

Sara Stymne and Maria Holmqvist: Processing of Swedish Compounds for Phrase-Based Statistical Machine Translation
In paper 2 we applied the split-merge strategy to a new language, Swedish. The study showed that the methods for compound splitting that were originally developed for German worked well for Swedish as well, with similar results. For compound splitting we needed to collect an inventory of compound suffixes. And for both splitting and merging, we had to take the 3-consonant spelling rule into account. We investigated two different ways of handling markup and normalization for compound parts: the marked and unmarked schemes.

A gold standard evaluation of compound splitting was performed in addition to the evaluation of splitting and merging in a PBSMT system. Two gold standards were created, one with all compounds, and one for one-to-one correspondence with English. We found that the precision was higher on the gold standard with all compounds, but that recall was higher on the

one-to-one test set. This is good since it shows that most of the splits performed actually splits real compounds, but the algorithm is better at finding compounds in one-to-one correspondence than other compounds.

The translation results were similar to the German experiments, with small improvements, especially for translation into Swedish. For translation into English the results were somewhat inconsistent across metrics, but again a reduction in the number of unknown words was seen. An error analysis of compound translation was performed, which showed a small improvement for the systems with split compounds. On the Swedish side no merging errors were found in this sample for the marked system, and only two reverse normalization errors were found in the unmarked system. Overall, there was no clear difference between the results with the two different markup schemes.

3.6.3 Paper 3

Sara Stymne: A Comparison of Merging Strategies for Translation of German Compounds

Paper 3 is focused on merging for translation into German. I explored different knowledge sources for merging, based on different combinations of the use of parts-of-speech, symbols and word lists. In this paper I explored three markup schemes: marked, unmarked and sepmarked. I also investigated the influence of an extra sequence model on parts-of-speech tags both for the baseline system and for the systems with splitting.

Automatic evaluation of the translation showed inconsistent results compared to the baseline. It did show, however, that there were big differences between the different merging algorithms, with the POS-matching and symbol methods consistently performing better than the word list based methods across both markup schemes and metrics.

An error analysis of the POS-matching merging algorithm showed that it produced a high percentage of correct compounds. Even though the symbol methods performed on par with the POS-matching algorithm on the automatic metrics, the error analysis showed that POS-matching does reduce the errors compared to using only symbols. Overall, the evaluation showed that for merging to be successful, some translation internal knowledge source is needed in the translation output. Using only unmarked output and a word list gave bad results.

For the baseline system, the use of a part-of-speech sequence model improved results as measured by Bleu, but not on PER, indicating that the usefulness of this model for the baseline is mainly to improve word order. For the systems with splitting, however, the results were improved both on Bleu and PER for all markup schemes and merging methods. The error analysis of compound merging confirmed this, by showing a reduction of erroneously placed compound parts when the extra sequence model is used.

4 Discussion

In the discussion I revisit the translation examples presented in the introduction, and discuss how compound processing affected them. I also describe how the methods presented have been further evaluated by participation in a shared task. Then I go on to discuss the findings of the papers before providing directions for future work, and a conclusion.

4.1 Translation examples

In the introduction I showed two problematic translation examples, with problems due to compounds, in Figures 1.1 and 1.2. These are repeated in Figures 4.1 and 4.2, where the output of the unmarked systems with compound treatment from paper 2 is also shown. This system will be called *comp-proc* in the following discussion. The compound translations are improved, but there are also other problems both with the comp-proc and baseline translations. Other phenomena than compounds have been affected by the compound processing, such as word choice and word order.

In Figure 4.1, there are three untranslated Swedish compounds in the baseline system. In the comp-proc system, the situation is improved with one good and one acceptable translation, and only one untranslated compound. The untranslated compound was not split since its last part *länders* (*countries'*), has not been seen in the genitive form in the monolingual training corpus.

There are also other changes, especially with regard to word choice. One example of this is *enligt vilket*, which is translated as *according to which* in the comp-proc system and in the reference, but as *in which*, without a verb in the baseline. In both system translations the modal verb *would* is missing, and both alternative wordings fail to express it in some other way.

In Figure 4.2, the coordinated compound that was problematic in the baseline system has been translated as a coordinated compound in the comp-proc system. There is a problem with word choice, however, since *sea* has been translated into *sjö*, which normally means *lake* but in compounds often have a meaning closer to *shipping*. This makes it a good translation in many compounds, but less fortunate in this particular case. In the baseline translation each word in the compound is translated separately, which makes it hard to understand, especially since the first part *havet* (*the sea*) is definite, and the head, *hamnar (ports)* happen to coincide with a present tense verb, *end up*.

Swedish original	Fru Lalumiéres betänkande återspeglar flera **Natoländers** tänkande enligt vilket **snabbinsatsstyrkorna** tämligen snabbt utvecklas till en fullskalig **krigsduglig** armé.
English comp-proc translation	Mrs Lalumiére's report reflects a number of **natoländers** thinking, according to which the **rapid reaction forces** relatively quickly develop into a full-scale **war operational** army.
English baseline translation	Mrs Lalumiére's report reflects a number of **natoländers** thinking in which **snabbinsatsstyrkorna** relatively quickly turned into a full-scale **krigsduglig** army.
English reference	Mrs Lalumiére's report reflects the thinking of many **nato countries**, according to which a **rapid reaction force** would very quickly develop into a fully-fledged army **capable of warfare**.

Figure 4.1: Example of a translation from Swedish to English by a baseline SMT system and with compound treatment (comp-proc)

English original	However, if we wish - and we do, for we consider it absolutely essential - **sea and river ports** to be included in the system of trans-European networks and to have their own system, then we must by necessity establish a hierarchy and a classification list for this system.
Swedish comp-proc translation	Men om vi vill - och det gör vi, för vi anser det absolut nödvändigt - **sjö- och flodhamnar** tas med i de transeuropeiska näten och har sina egna system, då måste vi upprätta en hierarki av nödvändighet och en klassificering listan för detta system.
Swedish baseline translation	Men, om vi vill - och det gör vi, eftersom vi anser det absolut nödvändigt - **havet och flod hamnar** skall ingå i systemet för transeuropeiska nät och få sitt eget system, då måste vi med nödvändighet upprätta en hierarki och en klassificering för detta system.
Swedish reference	Om vi trots detta vill - vilket vi gör, eftersom vi anser att det är absolut nödvändigt - att också **havs- och flodhamnarna** skall ingå i det transeuropeiska transportnätet och därmed kunna bilda ett system, måste vi införa en hierarki och en gradering.

Figure 4.2: Example of a translation from English to Swedish by a baseline SMT system and with compound treatment (comp-proc)

Again there are also other changes. There is a problem with word order in the comp-proc system where *nödvändighet* (*necessity*) has been misplaced, which changes the semantics of the sentence. There is also a split compound in the comp-proc system *klassificering listan* (*classification list*), which was not merged, since the first part was not marked as a compound part in the translation output. This concept is translated as the noncompound *klassificering* (*classification*) both in the baseline and the reference.

4.2 Shared task results

In addition to the three papers included in this thesis the suggested methods for compound treatment have been tested by using them as part of shared task contributions made by the MT research group at Linköping University (Stymne et al., 2008; Holmqvist et al., 2009) to the WMT[1] workshops of 2008 and 2009. Our system is called the *liu* system.

In the translation task the participants submit translations of the same test set, that are evaluated by a large scale human evaluation and by a high number of automatic metrics. Training material was supplied, which in 2008 consisted of two multilingual corpora, the large Europarl corpus and a smaller news corpus. In 2009 there were also large monolingual news corpora. In 2008 the evaluation was on both Europarl and news, but in 2009 it was only on news. There are several European language pairs in the shared task, but *liu* participated only in the English–German and German–English language pairs.

In the *liu* submissions we used a factored PBSMT system with compound processing techniques like those described in this thesis, where compounds were split before training, and merged after translation into German using the POS-matching algorithm.

The 2008 *liu* system also used factored translation with an additional sequence model based on part-of-speech tags, extended with morphology for German. The compound processing and the morphology treatment were not evaluated in isolation. We focused on the Europarl task and did not use the news corpus for training the system.

In 2008 the *liu* system was among the top scoring systems both based on human evaluation and on automatic metrics on the Europarl domain, but was less competitive on the news domain, to which it was not adapted (Callison-Burch et al., 2008). In addition we performed an error analysis of compound translation in the *liu* system, similar to that in paper 2, which showed an improvement compared to a baseline system.

In the 2009 task we extended the 2008 system by improved word alignment and domain adaptation to the news task. On the official human evaluation, sentence ranking, the *liu* system was among the best in the restricted condition, with systems that used no other resources than those provided for the

[1] The Workshop on Statistical Machine Translation, see `http://statmt.org/wmt08/` and `http://statmt.org/wmt09/`

workshop, but not as good as most of the unrestricted systems (Callison-Burch et al., 2009).

4.3 Findings

In this section I discuss the findings of this thesis with a focus on aspects concerning metrics, splitting, merging, and markup.

4.3.1 The use of automatic metrics

In many cases the different automatic metrics gave different results. In some cases it is quite clear what this difference means, as when comparing PER to other metrics, since PER is position independent and does not take word order into account as the other metrics do. In other cases it is hard to interpret exactly what the differences mean, but if many metrics point in the same direction, it is clearly a stronger indication that there is a real improvement.

The results raise the question of how useful the used automatic metrics are for this kind of task. In some cases, where there are clear improvements for humans, this could be punished by automatic metrics, as in producing *war operational* in Figure 4.1, instead of keeping it as a single untranslated word, which could raise the brevity penalty of Bleu, NIST and Neva. It is also the case that nearly perfect compounds, only missing a compound suffix, can be produced, which would not be recognized by metrics, but that are fully understandable for humans. Thus, it would be useful to find other ways of measuring the translation quality.

4.3.2 Compound splitting

The compound splitting algorithm that was originally developed for German was useful for Swedish as well, with similar results. The only language specific part of the splitting algorithm is the compound form setup and the 3-consonant rule. These can be easily configured in the splitting program. As long as there is an inventory of compound suffixes, the algorithm can be used for any language. It can be used without compound form treatment, but that is likely to reduce recall.

For German I found that different versions of the compound splitting algorithm performed better in the two translation directions. This was not explored thoroughly for Swedish, but it is likely that the same would hold there, which was indicated by a small pilot study.

Paper 1 indicated that there is no clear connection between the performance of a splitting algorithm on a gold standard, using metrics such as precision and recall, and translation results. It is, however, likely that the type of errors made would influence the results. An erroneous split that

results in words with incorrect semantics is clearly bad. But splits which result in parts that are suffixes rather than free morphemes do not necessarily hurt translation, since such parts are likely to form phrases in translation. I thus think that it is important to look at the types of errors at gold standard evaluation, not only on the results on metrics such as precision and recall.

A simple inspection of the split compounds shows that allowing all compound suffixes on every part does lead to errors. A simple way to remove some of these errors would be to filter the allowed compound suffixes based on part-of-speech, since nouns tend to have a much higher number of allowable suffixes than other parts-of-speech. A more restrictive way would be to collect lists of possible compound forms for different words, for instance based on the methods in Heid et al. (2002), but there is a risk of missing novel compounds with that approach.

One advantage of compound splitting, is that it reduces the vocabulary drastically, by around 55% for German and 45% for Swedish. Despite this reduction, the number of types is still higher for Swedish and German, than for English. This reduction of the vocabulary is in itself likely to be helpful for overall improvements of PBSMT, since vocabulary size is one explanatory factor of the hardness of PBSMT (Birch et al., 2008).

4.3.3 Compound merging

The novel POS-matching merging algorithm described in this thesis, gave good results for translation. It has the advantage over previous algorithms in that it can produce novel compounds, while reducing the risk of performing erroneous merges. It is based on a knowledge source that is internal to the translation process, part-of-speech tags. Using internal knowledge sources was better than using only external knowledge sources, such as different word lists. Using the other internal knowledge source, symbols, also gave good results, even though it did produce more erroneous merges.

The POS-matching strategy requires a decoder that can produce output factors, such as Moses. In paper 3, it was also shown that the extra sequence model on part-of-speech that can be used in a factored system was useful in improving the placement of compound parts in the translation part. If a decoder without factors is to be used, however, it would be possible to extend the symbol setup from just one symbol, to a set of more elaborate symbols that contain part-of-speech information, to allow some POS-matching. In the current marked scheme the symbols are not on heads, which would be needed for a matching scheme. This could be overcome by marking all possible heads as well. Another option is to tag the PBSMT output, which would, however, be problematic, since taggers are not trained on texts with split compounds.

4.3.4 Markup choices

Three different markup schemes were explored in the thesis. There were no clear differences between them. Especially between the marked and unmarked markup schemes the differences were small, both for Swedish and German. The sepmarked scheme was only used for German, and had results that differed from the other schemes, with worse results on the Bleu metric, but better on PER.

The special part-of-speech tag for compound forms is necessary for the POS-matching strategy, which generally gave superior merging results. The sepmarked strategy has the drawback of not having these tags, disabling the POS-matching algorithm for that scheme. It would be possible to use the special tags with the sepmarked scheme, which might improve the results for that markup scheme.

Words that did not have a matching head were left as single words in the translation output. This is fine for normalized words, which coincide with other words, but it can be problematic for nonnormalized words. It would be useful to normalize such words, but it would have a minor impact on translation results, since these parts are rare using the best methods, and many parts have compound forms that are identical to their base forms.

4.4 Future work

There are a number of possible directions for future work, based on the findings in this thesis. Below I outline and discuss some directions.

It would be useful to perform a thorough error analysis of the translation output. Such an analysis would give insights such as those presented in Section 4.1, but be more generally applicable since they would be based on a larger sample than a single sentence. Even though automatic metrics give some picture of improvements, they are hard to interpret, and an error analysis would help to give a fuller picture of the advantages and disadvantages of compound processing.

In this thesis all studies are performed on translation between German or Swedish with closed compounds, and English with open compounds. It would be interesting to perform similar studies between two languages with closed compounds, such as German and Swedish, where compound splitting would be needed for both languages. It is possible that the methods would be more successful in this case, since it is likely that the structure of the languages with regard to compound formation is more similar. I also believe that the presented methods could be applied to other compounding languages with good results.

The splitting algorithm suggested in this thesis is relatively simple, and does not perform very well on gold standard evaluation. Even though both previous research (Koehn and Knight, 2003) and paper 1 indicated that splitting quality on a gold standard does not affect translation quality to a

large extent, it is possible that this would be different if a splitting strategy that is much better is used. An error analysis, as suggested above, could also investigate which types of splits are problematic, which could be useful for improving splitting targeted at PBSMT. Improved splitting would also influence the merging process, since there hopefully would be fewer erroneous parts.

Compounds constitute one difference between German/Swedish and English, but there are many other differences, such as, verb placement, case on German nouns, and definiteness for Swedish nouns. In general I believe that it would be useful to identify differences between languages, which could be treated in a similar manner to compound processing, in a preprocessing step, with a possible matching postprocessing step if the target language is preprocessed. This strategy is less language independent than pure PBSMT, but this thesis indicates that PBSMT has much to gain from using language pair specific knowledge.

4.5 Conclusion

In this thesis I have shown that compound processing is useful for translation from and into the two compounding languages German and Swedish. Overall, compound processing gives some improvements, although results are somewhat inconsistent across translation directions and metrics. Generally the improvements are larger for translation into Swedish and German than into English. For translation into English there is a large reduction of untranslated words though, which clearly is an improvement.

I have extended an existing compound splitting method designed for machine translation, and shown that for translation between German and English, different splitting options tend to work better in the different translation directions. I also support earlier results indicating that there is no clear correlation between gold standard evaluation of compound splitting and of machine translation results.

Previous to the work presented in this thesis, there had not been much research on how to merge compounds after translation into a compounding language. I designed a part-of-speech matching algorithm for compound merging, and showed that it worked better than other suggested methods for translation into German. In particular I showed that using some kind of internal knowledge source such as part-of-speech or symbols, is superior to merging methods that only use external word and compound lists.

The compound processing methods were developed for translation from and into German. I have shown that theses methods work equally well for another language, Swedish, with only a few modifications for the different setup of compound suffixes in Swedish.

References

Abhaya Agarwal and Alon Lavie. 2008. METEOR, M-BLEU and M-TER: Evaluation metrics for high-correlation with human rankings of machine translation output. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 115–118. Columbus, Ohio, USA.

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 529–536. Sydney, Australia.

Enrique Alfonseca, Slaven Bilac, and Stefan Pharies. 2008a. Decompounding query keywords from compounding languages. In *Proceedings ofthe 46th Annual Meeting of the ACL: Human Language Technologies, Short papers*, pages 253–256. Columbus, Ohio, USA.

Enrique Alfonseca, Slaven Bilac, and Stefan Pharies. 2008b. German decompounding in a difficult corpus. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING)*, pages 128–139. Haifa, Israel.

Eleftherios Avramidis and Philipp Koehn. 2008. Enriching morphologically poor languages for statistical machine translation. In *Proceedings of the 46th Annual Meeting of the ACL: Human Language Technologies*, pages 763–770. Columbus, Ohio, USA.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43th Annual Meeting of the ACL*. Ann Arbor, Michigan, USA.

Marco Baroni, Johannes Matiasek, and Harald Trost. 2002. Predicting the components of German nominal compounds. In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI)*, pages 470–474. Amsterdam, the Netherlands.

André Berton, Pablo Fetter, and Peter Regel-Brietzmann. 1996. Compound words in large-vocabulary German speech recognition systems. In *Proceedings of the Fourth International Conference on Spoken Language Processing*, pages 1165–1168. Philadelphia, Pennsylvania, USA.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2007. CCG supertags in factored statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 9–16. Prague, Czech Republic.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting success in machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 745–754. Honolulu, Hawaii, USA.

Benny Brodda. 1979. Något om de svenska ordens fonotax och morfotax: Iakttagelse med utgångspunkt från experiment med automatisk morfologisk analys. In *PILUS nr 38*. Inst. för lingvistik, Stockholms universitet, Sweden.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2), pages 263–311.

Ralf D. Brown. 2002. Corpus-driven splitting of compound words. In *Proceedings of the Ninth International Conference of Theoretical and Methodological Issues in Machine Translation*, pages 12–21. Keihanna, Japan.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (Meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158. Prague, Czech Republic.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106. Columbus, Ohio, USA.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28. Athens, Greece.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *Proceedings of the 11th Conference of the EACL*, pages 249–256. Trento, Italy.

Johan Carlberger, Rickard Domeij, Viggo Kann, and Ola Knutsson. 2005. The development and performance of a grammar checker for Swedish: A language engineering perspective. In Ola Knutsson. 2005. *Developing and Evaluating Language Tools for Writers and Learners of Swedish*. Ph.D. thesis, Royal Institute of Technology (KTH), Stockholm, Sweden.

Johan Carlberger and Viggo Kann. 1999. Implementing an efficient part-of-speech tagger. *Software Practice and Experience*, 29, pages 815–832.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4), pages 359–393.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 263–270. Ann Arbor, Michigan, USA.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 531–540. Ann Arbor, Michigan, USA.

Arthur E. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1), pages 1–38.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurence statistics. In *Proceedings of the Second International Conference on Human Language Technology*, pages 228–231. San Diego, California, USA.

Kevin Duh. 2008. Ranking vs. regression in machine translation evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 191–194. Columbus, Ohio, USA.

Chris Dyer, Hendra Setiawan, Yuval Marton, and Philip Resnik. 2009. The University of Maryland statistical machine translation system for the Fourth Workshop on Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 145–149. Athens, Greece.

Anas El Isbihani, Shahram Khadivi, Oliver Bender, and Hermann Ney. 2006. Morpho-syntactic Arabic preprocessing for Arabic to English statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 15–22. New York City, New York, USA.

İlknur Durgar El-Kahlout and Kemal Oflazer. 2006. Initial explorations in English to Turkish statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 7–14. New York City, New York, USA.

Jakob Elming. 2008. *Syntactic Reordering in Statistical Machine Translation*. Ph.D. thesis, Copenhagen Business School, Denmark.

Eva Forsbom. 2003. Training a super model look-alike: Featuring edit distance, n-gram occurrence, and one reference translation. In *Proceedings of*

the Workshop on Machine Translation Evaluation: Towards Systemizing MT Evaluation, pages 29–36. New Orleans, Louisiana, USA.

Alexander Fraser. 2009. Experiments in morphosyntactic processing for translating to and from German. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 115–119. Athens, Greece.

Karin Friberg. 2007. Decomposing Swedish compounds using memory-based learning. In *Proceedings of the 16th Nordic Conference on Computational Linguistics (Nodalida)*, pages 224–230. Tartu, Estonia.

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

Jesús Giménez and Lluìs Márquez. 2008. A smorgasbord of features for automatic MT evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 195–198. Columbus, Ohio, USA.

John Goldsmith and Tom Reutter. 1998. Automatic collection and analysis of German compounds. In *Proceedings of the Coling-ACL Workshop on the Computational Treatment of Nominals Workshop*, pages 61–69. Montreal, Quebec, Canada.

Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. In *Proceedings of MT Summit XI*, pages 215–222. Copenhagen, Denmark.

Nizar Habash and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the 2006 Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 49–52. New York City, New York, USA.

Turid Hedlund. 2002. Compounds in dictionary-based cross-language information retrieval. *Information Research*, 7(2). Available at `http://InformationR.net/ir/7-2/paper128.html` (visited November 9, 2009).

Ulrich Heid, Bettina Säuberlich, and Arne Fitschen. 2002. Using descriptive generalizations in the acquisition of lexical data for a word formation analyser. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02)*. Las Palmas, Gran Canaria.

Staffan Hellberg. 1978. *The Morphology of Present-Day Swedish*. Number 13 in Data linguistica. Stockholm, Sweden: Almqvist & Wiksell.

Maria Holmqvist, Sara Stymne, and Lars Ahrenberg. 2007. Getting to know Moses: Initial experiments on German-English factored translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 181–184. Prague, Czech Republic.

Maria Holmqvist, Sara Stymne, Jody Foo, and Lars Ahrenberg. 2009. Improving alignment for SMT by reordering and augmenting the training corpus. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 120–124. Athens, Greece.

Florian Holz and Chris Biemann. 2008. Unsupervised and knowledge-free learning of compound splits and periphrases. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING)*, pages 117–127. Haifa, Israel.

Institut für Deutsche Sprache. 1998. Rechtschreibreform (Aktualisierte Ausgabe). IDS Sprachreport, Extra-Ausgabe Dezember 1998. Mannheim, Germany.

Harri Jäppinen and Leo Kulikov. 1991. Evaluation of machine translation systems: A system developer's viewpoint. In *Proceedings of the Evaluators' Forum*, pages 143–156. Les Rasses, Switzerland.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit X*, pages 79–86. Phuket, Thailand.

Philipp Koehn. 2009. *Moses, a Beam-Search Decoder for Factored Phrase-Based Statistical Machine Translation Models, User Manual and Code Guide*. University of Edinburgh. Software Manual.

Philipp Koehn, Abhishek Arun, and Hieu Hoang. 2008. Towards better machine translation quality for the German-English language pairs. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 139–142. Columbus, Ohio, USA.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*. Pittsburgh, Pennsylvania, USA.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural language Processing and Computational Natural Language Learning*, pages 868–876. Prague, Czech Republic.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL, demonstration session*, pages 177–180. Prague, Czech Republic.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the 10th Conference of the EACL*, pages 187–193. Budapest, Hungary.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the NAACL*, pages 48–54. Edmonton, Alberta, Canada.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227. Prague, Czech Republic.

Dimitros Kokkinakis. 2001. *A Framework for the Acquisition of Lexical Knowledge: Description and Applications*. Ph.D. thesis, Göteborg University, Sweden.

Roland Kuhn, Denis Yuen, Michel Simard, Patrick Paul, George Foster, Eric Joanis, and Howard Johnson. 2006. Segment choice models: Feature-rich models for global distortion in statistical machine translation. In *Proceedings of the 2006 Human Language Technology Conference of the NAACL*, pages 25–32. New York City, New York, USA.

Sebastian Kürschner. 2003. *Von Volk-s-musik und Sport-O-geist im Lemming-0-land – Af folk-e-musik og sport-s-ånd i lemming-e-landet: Fugenelemente im Deutschen und Dänischen – eine kontrastive Studie zu einem Grenzfall der Morphologie*. Master's thesis, Albert-Ludwigs-Universität, Freiburg, Germany.

Stefan Langer. 1998. Zur Morphologie und Semantik von Nominalkomposita. In *Tagungsband der 4. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS)*, pages 83–97. Bonn, Germany.

Martha Larson, Daniel Willett, Joachim Köhler, and Gerhard Rigoll. 2000. Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches. In *Proceedings of the Sixth International Conference on Spoken Language Processing*, volume 3, pages 945–948. Beijing, China.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Prague, Czech Republic.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8), pages 707–710.

Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL*, pages 720–727. Prague, Czech Republic.

Jin-Ji Li, Jungi Kim, Dong-Il Kim, and Jong-Hyeok Lee. 2009. Chinese syntactic reordering for adequate generation of Korean verbal phrases in Chinese-to-Korean SMT. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 190–196. Athens, Greece.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 609–616. Sydney, Australia.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 133–139. Pennsylvania, Pennsylvania, USA.

Einat Minkov, Kristina Toutanova, and Hisami Suzuki. 2007. Generating complex morphology for machine translation. In *Proceedings of the 45th Annual Meeting of the ACL*, pages 128–135. Prague, Czech Republic.

Hwidong Na, Jin-Ji Li, Jungi Kim, and Jong-Hyeok Lee. 2009. Improving fluency by reordering target constituents using MST parser in English-to-Japanese phrase-based SMT. In *Proceedings of MT Summit XII*, pages 276–283. Ottawa, Ontario, Canada.

Jan Niehues and Muntsin Kolss. 2009. A POS-based model for long-range reorderings in SMT. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 206–214. Athens, Greece.

Sonja Nießen and Hermann Ney. 2000. Improving SMT quality with morpho-syntactic analysis. In *Proceedings of the 2003 Human Language Technology Conference of the NAACL*, pages 1081–1085. Saarbrücken, Germany.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 42nd Annual Meeting of the ACL*, pages 160–167. Sapporo, Japan.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of the 2004 Human Language Technology Conference of the NAACL*, pages 161–168. Boston, Massachusetts, USA.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the ACL*, pages 440–447. Hong Kong.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1), pages 19–51.

Lene Offersgaard, Claus Povlsen, Lisbeth Almsten, and Bente Maegaard. 2008. Domain specific MT in use. In *Proceedings of the 12th Annual Conference of the European Association for Machine Translation*, pages 150–159. Hamburg, Germany.

Karolina Owczarzak, Josef van Genabith, and Andy Way. 2007. Dependency-based automatic evaluation for machine translation. In *Proceedings of the Workshop on Syntax and Structure in Statistical Translation*, pages 80–87. Rochester, New York, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 311–318. Philadelphia, Pennsylvania, USA.

Michael Paul, Andrew Finch, and Eiichiro Sumita. 2009. NICT@WMT09: Model adaptation and transliteration for Spanish-English SMT. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 105–109. Athens, Greece.

Maja Popović and Hermann Ney. 2009. Syntax-oriented evaluation measures for machine translation output. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 29–32. Athens, Greece.

Maja Popović, Daniel Stein, and Hermann Ney. 2006. Statistical machine translation of German compound words. In *Proceedings of FinTAL – 5th International Conference on Natural Language Processing*, pages 616–624. Turku, Finland: Springer Verlag, LNCS.

Ulrike Rackow, Ido Dagan, and Ulrike Schwall. 1992. Automatic translation of noun compounds. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 1249–1253. Nantes, France.

Ananthakrishnan Ramanathan, Hansraj Choudhary, Avishek Ghosh, and Pushpak Bhattacharyya. 2009. Case markers and morphology: Addressing the crux of the fluency problem in English-Hindi SMT. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 800–808. Suntec, Singapore.

Magnus Rosell. 2003. Improving clustering of Swedish newspaper articles using stemming and compound splitting. In *Proceedings of the 14th Nordic Conference on Computational Linguistics (Nodalida)*. Reykjavik, Iceland.

Anne Schiller. 2005. German compound analysis with wfsc. In *Proceedings of Finite State Methods and Natural Language Processing*, pages 239–246. Helsinki, Finland: Springer Verlag, LNCS.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49. Manchester, UK.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the 2004 Human Language Technology Conference of the NAACL*, pages 177–184. Boston, Massachusetts, USA.

Jonas Sjöbergh and Viggo Kann. 2004. Finding the correct interpretation of Swedish compounds, a statistical approach. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 901–904. Denver, Colorado, USA.

Sara Stymne. 2008. German compounds in factored statistical machine translation. In Aarne Ranta and Bengt Nordström, editors, *Proceedings of GoTAL – 6th International Conference on Natural Language Processing*, pages 464–475. Gothenburg, Sweden: Springer Verlag, LNCS/LNAI.

Sara Stymne. 2009. A comparison of merging strategies for translation of German compounds. In *Proceedings of the EACL 2009 Student Research Workshop*, pages 61–69. Athens, Greece.

Sara Stymne and Maria Holmqvist. 2008. Processing of Swedish compounds for phrase-based statistical machine translation. In *Proceedings of the 12th Annual Conference of the European Association for Machine Translation*, pages 180–189. Hamburg, Germany.

Sara Stymne, Maria Holmqvist, and Lars Ahrenberg. 2008. Effects of morphological analysis in translation between German and English. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 135–138. Columbus, Ohio, USA.

Olof Thorell. 1981. *Svensk ordbildningslära*. Stockholm, Sweden: Esselte Studium.

Christoph Tillmann, Stephan Vogel, Hermann Ney, Alex Zubiaga, and Hassan Sawaf. 1997. Accelerated DP based search for statistical translation. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2667–2670. Rhodes, Greece.

David Vilar, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. 2006. Error analysis of machine translation output. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*, pages 697–702. Genoa, Italy.

Sami Virpioja, Jaako J. Väyrynen, Mathias Creutz, and Markus Sadeniemi. 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In *Proceedings of MT Summit XI*, pages 491–498. Copenhagen, Denmark.

Stephan Vogel, Hermann Ney, and Christoph Tillman. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 836–841. Copenhagen, Denmark.

Stephan Vogel, Sonja Nießen, and Hermann Ney. 2000. Automatic extrapolation of human assessment of translation quality. In *Proceedings of the Workshop on the Evaluation of Machine Translation at LREC'2000*, pages 35–39. Athens, Greece.

Ian H. Witten and Timothy C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4), pages 1085–1094.

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 508–514. Geneva, Switzerland.

Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical MT. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 303–310. Philadelphia, Pennsylvania, USA.

Min Zhang, Hongfei Jiang, Ai Ti Aw, Jun Sun, Sheng Li, and Chew Lim Tan. 2007a. A tree-to-tree alignment-based model for statistical machine translation. In *Proceedings of MT Summit XI*, pages 535–542. Copenhagen, Denmark.

Yuqi Zhang, Richard Zens, and Hermann Ney. 2007b. Improved chunk-level reordering for statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 21–28. Trento, Italy.