A Comparison of Merging Strategies for Translation of German Compounds

Sara Stymne
Department of Computer and Information Science
Linköping University, Sweden
sarst@ida.liu.se

Abstract

In this article, compound processing for translation into German in a factored statistical MT system is investigated. Compounds are handled by splitting them prior to training, and merging the parts after translation. I have explored eight merging strategies using different combinations of external knowledge sources, such as word lists, and internal sources that are carried through the translation process, such as symbols or parts-of-speech. I show that for merging to be successful, some internal knowledge source is needed. I also show that an extra sequence model for part-of-speech is useful in order to improve the order of compound parts in the output. The best merging results are achieved by a matching scheme for part-of-speech tags.

1 Introduction

In German, as in many other languages, compounds are normally written as single words without spaces or other word boundaries. Compounds can be binary, i.e., made up of two parts (1a), or have more parts (1b). There are also coordinated compound constructions (1c). In a few cases compounds are written with a hyphen (1d), often when one of the parts is a proper name or an abbreviation.

(1) a. Regierungskonferenz
intergovernmental conference

b. Fremdsprachenkenntnisse
knowledge of foreign languages

c. See- und Binnenhäfen
sea and inland ports

d. Kosovo-Konflikt
Kosovo conflict

e. Völkermord
genocide

German compounds can have English translations that are compounds, written as separate words (1a), other constructions, possibly with inserted function words and reordering (1b), or single words (1e). Compound parts sometimes have special compound forms, formed by addition or truncations of letters, by *umlaut* or by a combination of these, as in (1a), where the letter *-s* is added to the first part, *Regierung*. For an overview of German compound forms, see Langer (1998).

Compounds are productive and common in German and other Germanic languages, which makes them problematic for many applications including statistical machine translation. For translation into a compounding language, fewer compounds than in normal texts are often produced, which can be due to the fact that the desired compounds are missing in the training data, or that they have not been aligned correctly. Where a compound is the idiomatic word choice in the translation, a MT system can instead produce separate words, genitive or other alternative constructions, or only translate one part of the compound.

The most common way to integrate compound processing into statistical machine translation is to split compounds prior to training and translation. Splitting of compounds has received a lot of focus in the literature, both for machine translation, and targeted at other applications such as information retrieval or speech recognition.

When translating into a compounding language there is a need to merge the split compounds after translation. In order to do this we have 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.

Proceedings of the EACL 2009 Student Research Workshop, pages 61–69,
Athens, Greece, 2 April 2009. ©2009 Association for Computational Linguistics

In this article I explore the effects of merging in a factored phrase-based statistical machine translation system. The system uses part-of-speech as an output factor. This factor is used as a knowledge source for merging and to improve word order by using a part-of-speech (POS) sequence model.

There are different knowledge sources for merging. Some are external, such as frequency lists of words, compounds, and compound parts, that could be compiled at split-time. It is also possible to have internal knowledge sources, that are carried through the translation process, such as symbols on compound parts, or part-of-speech tags. Choices made at split-time influence which internal knowledge sources are available at merge-time. I will explore and compare three markup schemes for compound parts, and eight merging algorithms that use different combinations of knowledge sources.

2 Related Work

Splitting German compounds into their parts prior to translation has been suggested by many researchers. Koehn and Knight (2003) presented an empirical splitting algorithm that is used to improve translation from German to English. They split all words in all possible places, and considered a splitting option valid if all the parts are existing words in a monolingual corpus. They allowed the addition of *-s* or *-es* at all splitting points. If there were several valid splitting options they chose one based on the number of splits, the geometric mean of part frequencies or based on alignment data. Stymne (2008) extended this algorithm in a number of ways, for instance by allowing more compound forms. She found that for translation into German, it was better to use the arithmetic mean of part frequencies than the geometric mean. Using the mean of frequencies can result in no split, if the compound is more frequent than its parts.

Merging has been much less explored than splitting since it is common only to discuss translation from compounding languages. However, Popović et al. (2006) used merging for translation into German. They did not mark compound parts in any way, so the merging is based on two word lists, with compound parts and full compounds found at split-time. All words in the translation output that were possible compound parts were merged with the next word if it resulted in a known compound. They only discussed merging of binary compounds. The drawback of this method is that novel compounds cannot be merged. Nevertheless, this strategy led to improved translation measured by three automatic metrics.

In a study of translation between English and Swedish, Stymne and Holmqvist (2008) suggested a merging algorithm based on part-of-speech, which can be used in a factored translation system with part-of-speech as an output factor. Compound parts had special part-of-speech tags based on the head of the compound, and merging was performed if that part-of-speech tag matched that of the following word. When compound forms had been normalized the correct compound form was found by using frequency lists of parts and words compiled at split-time. This method can merge unseen compounds, and the tendency to merge too much is reduced by the restriction that POS-tags need to match. In addition coordinated compounds were handled by the algorithm. This strategy resulted in improved scores on automatic metrics, which were confirmed by an error analysis.

Koehn et al. (2008) discussed treatment of hyphened compounds in translation into German by splitting at hyphens and treat the hyphen as a separate token, marked by a symbol. The impact on translation results was small.

There are also other ways of using compound processing to improve SMT into German. Popović et al. (2006) suggested using compound splitting to improve alignment, or to merge English compounds prior to training.

Some work has discussed merging of not only compounds, but of all morphs. Virpioja et al. (2007) merged translation output that was split into morphs for Finnish, Swedish and Danish. They marked split parts with a symbol, and merged every word in the output which had this symbol with the next word. If morphs were misplaced in the translation output, they were merged anyway, possibly creating non-existent words. This system was worse than the baseline on Bleu (Papineni et al., 2002), but an error analysis showed some improvements.

El-Kahlout and Oflazer (2006), discuss merging of morphs in Turkish. They also mark morphs with a symbol, and in addition normalize affixes to standard form. In the merging phase, surface forms were generated following morphographemic rules. They found that morphs were often translated out of order, and that merging based purely on symbols gave bad results. To reduce this risk, they constrained splitting to allow only morphologically correct splits, and by grouping some morphemes. This lead to less ordering problems in the translation output and gave improvements over the baseline.

Compound recombination have also been applied to German speech recognition, e.g. by (Berton et al., 1996), who performed a lexical search to extend the word graph that is output by the speech recogniser.

3 Compound Processing

German compounds are split in the training data and prior to translation. After translation, the parts are merged to form full compounds. The knowledge sources available to the merging process depend on which information is carried through the translation process.

The splitting algorithm of Stymne (2008) will be used throughout this study. It is slightly modified such that only the 10 most common compound forms from a corpus study of Langer (1998) are allowed, and the hyphen in hyphened compounds is treated as a compound form, analogous to adding for instance the letter *s* to a part.

The annotation of compound parts influences the merging process. Choices have to be made concerning the form, markup and part-of-speech of compound parts. For the form two options have been considered, keeping the original compound form, or normalizing it so that it coincides with a normal word. Three types of marking have been investigated, no marking at all (*unmarked*), a marking symbol that is concatenated to all parts but the last (*marked*), or using a separate symbol between parts (*sepmarked*). The sepmarked scheme has different symbols for parts of coordinated compounds than for other compounds. Parts are normalized in the unmarked and sepmarked schemes, but left in their compound form in the marked scheme, since the symbol separates them from ordinary words in any case.

There is also the issue of which part-of-speech tag to use for compound parts. The last part of the compound, the head, always has the same part-of-speech tag as the full compound. Two schemes are explored for the other parts. For the marked and unmarked system, a part-of-speech tag that is derived from that of the last part of the word is used. For the sepmarked scheme the most common part-of-speech tag of the part from the tagged monolingual corpus is used.

In summary, the three markup schemes use the following combinations, exemplified by the result of splitting the word *begrüßenswert* (*welcome*, literally *worth to welcome*)

- **Unmarked:** no symbol, normalization, special POS-tags
  begrüßen ADJ-PART *wert* ADJ
- **Marked:** symbol on parts, no normalization, special POS-tags
  begrüßens# ADJ-PART *wert* ADJ
- **Sepmarked:** symbol as separate token, normalization, ordinary POS-tags
  begrüßen VV @#@ COMP *wert* ADJ

3.1 Merging

There is no guarantee that compound parts appear in a correct context in the translation output. This fact complicates merging, since there is a general choice between only merging those words that we know are compounds, and merging all occurrences of compound parts, which will merge unseen compounds, but probably also merge parts that do not form well-formed compounds. There is also the issue of parts possibly being part of coordinated compounds.

The internal knowledge sources that can be used for merging depends on the markup scheme used. The available internal sources are markup symbols, part-of-speech tags, and the special tags for compound parts. The external resources are frequency lists of words, compounds and parts, possibly with normalization, compiled at split-time.

For the unmarked and sepmarked scheme, reverse normalization, i.e., mapping normalized compound parts into correct compound forms, has to be applied in connection with merging. As in Stymne and Holmqvist (2008), all combinations of compound forms that are known for each part are looked up in the word frequency list, and the most frequent combination is chosen. If there are no known combinations, the parts are combined from left to right, at each step choosing the most frequent combination.

Three main types of merging algorithms are investigated in this study. The first group, inspired

Name	Description
word-list	Merges all tokens that have been seen as compound parts with the next part if it results in a known word, from the training corpus
word-list + head-pos	As word-list, but only merges words where the last part is a noun, adjective or verb
compound-list	As word-list, but for known compounds from split-time, not for all known words
symbol	Merges all tokens that are marked with the next token
symbol + head-pos	As symbol, but only merges words where the last part is a noun, adjective or verb
symbol + word-list	A mix of symbol and word-list, where marked compounds are merged, if it results in a known word
POS-match	Merges all tokens with a compound part-of-speech tag, if the tag match the tag of the next token
POS-match + coord	As POS-match, but also adds a hyphen to parts that are followed by the conjunction *und* (*and*)

Table 1: Merging algorithms

by Popović et al. (2006), is based only on external knowledge sources, frequency lists of words or compounds, and of parts, compiled at split-time. Novel compounds cannot be merged by these algorithms. The second group uses symbols to guide merging, inspired by work on morphology merging (Virpioja et al., 2007). In the unmarked scheme where compound parts are not marked with symbols, the special POS-tags are used to identify parts instead[1]. The third group is based on special part-of-speech tags for compounds (Stymne and Holmqvist, 2008), and merging is performed if the part-of-speech tags match. This group of algorithms cannot be applied to the sepmarked scheme.

In addition a restriction that the head of the compound should have a compounding part-of-speech, that is, a noun, adjective, or verb, and a rule to handle coordinated compounds are used. By using these additions and combinations of the main algorithms, a total of eight algorithms are explored, as summarized in Table 1. For all algorithms, compounds can have an arbitrary number of parts.

If there is a marked compound part that cannot be combined with the next word, in any of the algorithms, the markup is removed, and the part is left as a single word. For the sepmarked system, coordinated compounds are handled as part of the symbol algorithms, by using the special markup symbol that indicates them.

3.2 Merging Performance

To give an idea of the potential of the merging algorithms, they are evaluated on the split test reference corpus, using the unmarked scheme. The corpus has 55580 words, of which 4472 are identified as compounds by the splitting algorithm. Of these 4160 are known from the corpus, 245 are novel, and 67 are coordinated. For the methods based on symbols or part-of-speech, this merging task is trivial, except for reverse normalization, since all parts are correctly ordered.

Table 2 shows the number of errors. The POS-match algorithm with treatment of coordination makes 55 errors, 4 of which are due to coordinated compounds that does not use *und* as the conjunction. The other errors are due to errors in the reverse normalization of novel compounds, which has an accuracy of 79% on this text. The POS-match and symbol algorithms make additional errors on coordinated compounds. The head-pos restriction blocks compounds with an adverb as head, which gave better results on translation data, but increased the errors on this evaluation. The word list method both merges many words that are not compounds, and do not merge any novel compounds. Using a list of compounds instead of words reduces the errors slightly.

4 System Description

The translation system used is a factored phrase-based translation system. In a factored translation model other factors than surface form can be used, such as lemma or part-of-speech (Koehn and Hoang, 2007). In the current system part-of-speech is used only as an output factor in the target language. Besides the standard language model a sequence model on part-of-speech is used, which can be expected to lead to better word order in the translation output. There are no input factors, so no tagging has to be performed prior to translation, only the training corpus needs to be tagged. In addition, the computational overhead is small. One possible benefit gained by using part-of-speech as an output factor is that ordering, both in general, and of compound parts, can be improved. This hypothesis is tested by trying two system setups, with and without the part-of-speech sequence model. In addition part-of-speech is used for postprocess-

[1] For the marked scheme using POS-tags to identify compound parts is equivalent to using symbols.

wlist	wlist+head-pos	clist	symbol	symbol+head-pos	symbol+wlist	POS-match	POS-match+coord
2393	1656	2257	118	205	330	118	55

Table 2: Number of merging errors on the split reference corpus

		Tokens	Types
English	baseline	15158429	63692
German	baseline	14356051	184215
	marked	15674728	93746
	unmarked	15674728	81806
	sepmarked	17007929	81808

Table 3: Type and token counts for the 701157 sentence training corpus

ing, both for uppercasing German nouns and as a knowledge source for compound merging.

The tools used are the Moses toolkit (Koehn et al., 2007) for decoding and training, GIZA++ for word alignment (Och and Ney, 2003), and SRILM (Stolcke, 2002) for language models. A 5-gram model is used for surface form, and a 7-gram model is used for part-of-speech. To tune feature weights minimum error rate training is used (Och, 2003), optimized against the Neva metric (Forsbom, 2003). Compound splitting is performed on the training corpus, prior to training. Merging is performed after translation, both for test, and incorporated into the tuning step.

4.1 Corpus

The system is trained and tested on the Europarl corpus (Koehn, 2005). The training corpus is filtered to remove sentences longer than 40 words and with a length ratio of more than 1 to 7. The filtered training corpus contains 701157 sentences. 500 sentences are used for tuning and 2000 sentences for testing[2]. The German side of the training corpus is part-of-speech tagged using TreeTagger (Schmid, 1994).

The German corpus has nearly three times as many types, i.e., unique tokens, as the English corpus despite having a somewhat lower token count, as shown for the training corpus in Table 3. Compound splitting drastically reduces the number of types, to around half or less, even though it is still larger than for English. Marking on parts gives 15% more types than no marking.

5 Evaluation

Two types of evaluation are performed. The influence of the different merging algorithms on the overall translation quality is evaluated, using two automatic metrics. In addition the performance of the merging algorithms are analysed in some more detail. In both cases the effect of the POS sequence model is also discussed. Even when the POS sequence model is not used, part-of-speech is carried through the translation process, so that it can be used in the merging step.

5.1 Evaluation of Translation

Translations are evaluated on two automatic metrics: Bleu (Papineni et al., 2002) and PER, position independent error-rate (Tillmann et al., 1997). Case-sensitive versions of the metrics are used. PER does not consider word order, it evaluates the translation as a bag-of-word, and thus the systems without part-of-speech sequence models can be expected to do well on PER. Note that PER is an error-rate, so lower scores are better, whereas higher scores are better for Bleu.

These metrics have disadvantages, for instance because the same weight is given to all tokens, both to complex compounds, and to function words such as *und* (*and*). Bleu has been criticized, see e.g. (Callison-Burch et al., 2006; Chiang et al., 2008).

Table 4 and 5 shows the translation results using the different merging algorithms. For the systems with POS sequence models the baseline performs slightly better on Bleu, than the best systems with merging. Without the POS sequence model, however, merging often leads to improvements, by up to 0.48 Bleu points. For all systems it is advantageous to use the POS sequence model.

For the baseline, the PER scores are higher for the system without a POS sequence model, which, compared to the Bleu scores, confirms the fact that word order is improved by the sequence model. The systems with merging are better than the baseline with the POS sequence model. In all cases, however, the systems with merging performs worse when not using a POS sequence model, indicating that the part-of-speech

[2] The test set is *test2007* from the ACL 2008 Workshop on Statistical Machine Translation, `http://www.statmt.org/wmt08/shared-task.html`

	with POS-model			without POS-model		
	unmarked	sepmarked	marked	unmarked	sepmarked	marked
word-list	17.93	17.66	18.92	17.70	17.29	18.69
word-list + head-pos	19.34	19.07	19.60	19.13	18.63	19.38
compound-list	18.94	17.77	18.13	18.56	17.40	17.86
symbol	20.02	19.57	20.03	**19.66**	19.14	**19.79**
symbol + head-pos	20.02	19.55	20.01	**19.75**	19.12	**19.78**
symbol + word-list	20.03	19.72	20.02	**19.76**	19.29	**19.79**
POS-match	20.12	–	20.03	**19.84**	–	**19.80**
POS-match + coord	20.10	–	19.97	**19.85**	–	**19.80**

Table 4: Translation results for Bleu. Baseline with POS: 20.19, without POS: 19.66. Results that are better than the baseline are marked with bold face.

	with POS-model			without POS-model		
	unmarked	sepmarked	marked	unmarked	sepmarked	marked
word-list	29.88	28.64	28.19	30.27	29.94	28.71
word-list + head-pos	27.49	**26.07**	27.26	27.78	27.22	27.84
compound-list	**26.92**	27.99	29.25	27.46	29.07	29.74
symbol	**27.21**	**26.13**	**26.95**	27.70	27.40	27.61
symbol + head-pos	**27.11**	**26.10**	**26.92**	27.34	27.35	27.54
symbol + word-list	**26.86**	**25.54**	**26.80**	27.15	26.72	27.39
POS-match	**26.99**	–	**26.93**	27.17	–	27.53
POS-match + coord	**27.10**	–	**26.93**	27.28	–	27.53

Table 5: Translation results for PER. Baseline with POS: 27.22, without POS: 26.49. Results that are better than the baseline are marked with bold face.

sequence model improves the order of compound parts.

When measured by PER, the best results when using merging are achieved by combining symbols and word lists, but when measured by Bleu, the POS-based algorithms are best. The simpler symbol-based methods, often have similar scores, and in a few cases even better. Adding treatment of coordinated compounds to the POS-match algorithm changes scores marginally in both directions. The word list based methods, however, generally give bad results. Using the head-pos restriction improves it somewhat and using a compound list instead of a word list gives different results in the different markup schemes, but is still worse than the best systems. This shows that some kind of internal knowledge source, either symbols or part-of-speech, is needed in order for merging to be successful.

On both metrics, the marked and unmarked system perform similarly. They are better than the sepmarked system on Bleu, but the sepmarked system is a lot better on PER, which is an indication of that word order is problematic in the sepmarked system, with its separate tokens to indicate compounds.

5.2 Evaluation of Merging

The results of the different merging algorithms are analysed to find the number of merges and the type and quality of the merges. In addition I investigate the effect of using a part-of-speech model on the merging process.

Table 6 shows the reduction of words[3] achieved by applying the different algorithms. The word list based method produces the highest number of merges in all cases, performing many merges where the parts are not recognized as such by the system. The number of merges is greatly reduced by the head-pos restriction. An investigation of the output of the word list based method shows that it often merges common words that incidentally form a new word, such as *bei* (*at*) and *der* (*the*) to *beider* (*both*). Another type of error is due to errors in the corpus, such as the merge of *umwelt* (*environment*) and *und* (*and*), which occurs in the corpus, but is not a correct German word. These two error types are often prohibited by the head-pos restrictions. The compound list method avoids these errors, but it does not merge compounds that were not split by the splitting algorithm, due to a high frequency, giving a very low number of splits in some cases. There are small differences between the POS-match and symbol algorithms. Not using the POS sequence model results in a higher number of merges for all systems.

A more detailed analysis was performed of the

[3] The reduction of words is higher than the number of produced compounds, since each compound can have more than two parts.

	with POS-model			without POS-model		
	unmarked	sepmarked	marked	unmarked	sepmarked	marked
word-list	5275	5422	4866	5897	5589	5231
word-list + head-pos	4161	4412	4338	4752	4601	4661
compound-list	4460	4669	3253	5116	4850	3534
symbol	4431	4712	4332	5144	4968	4702
symbol + head-pos	4323	4671	4279	4832	4899	4594
symbol + word-list	4178	4436	4198	4753	4656	4530
POS-match	4363	–	4310	4867	–	4618
POS-match + coord	4361	–	4310	4865	–	4618

Table 6: Reduction of number of words by using different merging algorithms

		with POS-model			without POS-model		
		unmarked	sepmarked	marked	unmarked	sepmarked	marked
Known		3339	3594	3375	3747	3762	3587
Novel	Good	168	176	105	104	245	93
	Bad	20	97	8	10	64	7
Coordinated	Good	43	43	42	42	37	44
	Bad	9	9	3	22	7	5
Single part	Good	6	–	5	136	–	33
	Bad	11	–	16	52	–	46
Total		3596	3919	3554	4113	4115	3815

Table 7: Analysis of merged compounds

compounds parts in the output. The result of merging them are classified into four groups: merged compounds that are known from the training corpus (2a) or that are novel (2b), parts that were not merged (2c), and parts of coordinated compounds (2d). They are classified as bad if the compound/part should have been merged with the next word, does not fit into its context, or has the wrong form.

(2) a. Naturschutzpolitik
nature protection policy

b. UN-Friedensplan
UN peace plan

c. * West- zulassen
west allow

d. Mittel- und Osteuropa
Central and Eastern Europe

For the unmarked and sepmarked systems, the classification was based on the POS-match constraint, where parts are not merged if the POS-tags do not match. POS-match cannot be used for the sepmarked scheme, which has standard POS-tags.

Table 7 shows the results of this analysis. The majority of the merged compounds are known from the training corpus for all systems. There is a marked difference between the two systems that use POS-match, and the sepmarked system that does not. The sepmarked system found the highest number of novel compounds, but also have the highest error rate for these, which shows that it is useful to match POS-tags. The other two systems find fewer novel compounds, but also make fewer mistakes. The marked system has more errors for single parts than the other systems, mainly beacuse the form of compound parts were not normalized. Very few errors are due to reverse normalization. In the unmarked system with a POS sequence model, there were only three such errors, which is better than the results on split data in Section 3.2.

Generally the percentage of bad parts or compounds is lower for the systems with a POS sequence model, which shows that the sequence model is useful for the ordering of compound parts. The number of single compound parts is also much higher for the systems without a POS sequence model. 80% of the merged compounds in the unmarked system are binary, i.e., have two parts, and the highest number of parts in a compound is 5. The pattern for the other systems is similar.

All systems produce fewer compounds than the 4472 in the German reference text. However, there might also be compounds in the output, that were not split and merged. These numbers are not directly comparable to the baseline system, and applying the POS-based splitting algorithm to translation output would not give a fair comparison.

An indication of the number of compounds in a text is the number of long words. In the reference text there are 351 words with at least 20 characters, which will be used as the limit for long words. A manual analysis showed that all these words are compounds. The baseline system produces 209 long words. The systems with merging, discussed above, all produce more long words than the baseline, but less than the reference, between 263 and 307, with the highest number in the marked system. The trend is the same for the systems without a POS sequence model, but with slightly fewer long words than for the systems with merging.

6 Discussion

The choice of merging method has a large impact on the final translation result. For merging to be successful some internal knowledge source, such as part-of-speech or symbols is needed. The pure word list based method performed the worst of all systems on both metrics in most cases, which was not surprising, considering the evaluation of the merging algorithms on split data, where it was shown that the word-list based methods merged many parts that were not compounds.

The combination of symbols and word lists gave good results on the automatic metrics. An advantage of this method is that it is applicable for translation systems that do not use factors. However, it has the drawback that it does not merge novel compounds, and finds fewer compounds than most other algorithms. The error analysis shows that many valid compounds are discarded by this algorithm. A method that both find novel compounds, and that works well is that based on POS-match. In its current form it needs a decoder that can handle factored translation models. It would, however, be possible to use more elaborate symbols with part-of-speech information, which would allow a POS-matching scheme, without the need of factors.

The error analysis of merging performance showed that merging works well, especially for the two schemes where POS-matching is possible, where the proportion of errors is low. It also showed that using a part-of-speech sequence model was useful in order to get good results, specifically since it increased the number of compound parts that were placed correctly in the translation output.

The sepmarked scheme is best on the PER metric it is worse on Bleu, and the error analysis shows that it performs worse on merging than the other systems. This could probably be improved by the use of special POS-tags and POS-matching for this scheme as well. It is hard to judge which is best of the unmarked and marked scheme. They perform similarly on the metrics, and there is no clear difference in the error analysis. The unmarked scheme does produce a somewhat higher number of novel compounds, though. A disadvantage of the marked scheme is that the compound form is kept for single parts. A solution for this could be to normalize parts in this scheme as well, which could improve performance, since reverse normalization performance is good on translation data.

The systems with splitting and merging have more long words than the baseline, which indicates that they are more successful in creating compounds. However, they still have fewer long words than the reference text, indicating the need of more work on producing compounds.

7 Conclusion and Future Work

In this study I have shown that the strategy used for merging German compound parts in translation output influences translation results to a large extent. For merging to be successful, it needs some internal knowledge source, carried through the translation process, such as symbols or part-of- speech. The overall best results were achieved by using matching for part-of-speech.

One factor that affects merging, which was not explored in this work, is the quality of splitting. If splitting produces less erroneously split compounds than the current method, it is possible that merging also can produce better results, even though it was not clear from the error analysis that bad splits were a problem. A number of more accurate splitting strategies have been suggested for different tasks, see e.g. Alfonseca et al. (2008), that could be explored in combination with merging for machine translation.

I have compared the performance of different merging strategies in one language, German. It would be interesting to investigate these methods for other compounding languages as well. I also want to explore translation between two compounding languages, where splitting and merging would be performed on both languages, not only on one language as in this study.

References

Enrique Alfonseca, Slaven Bilac, and Stefan Pharies. 2008. Decompounding query keywords from compounding languages. In *Proceedings of ACL-08: HLT, Short Papers*, pages 253–256, Columbus, Ohio.

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 (ICSLP)*, pages 1165–1168, Philadelphia, Pennsylvania, USA.

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 EACL*, pages 249–256, Trento, Italy.

David Chiang, Steve DeNeefe, Yee Seng Chan, and Hwee Tou Ng. 2008. Decomposability of translation metrics for improved evaluation and efficient algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 610–619, Honolulu, Hawaii.

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

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.

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

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

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*, Prague, Czech Republic.

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.

Philipp Koehn. 2005. Europarl: a parallel corpus for statistical machine translation. In *Proceedings of MT Summit X*, Phuket, Thailand.

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

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

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of ACL*, pages 160–167, Sapporo, Japan.

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.

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.

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

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

Sara Stymne and Maria Holmqvist. 2008. Processing of Swedish compounds for phrase-based statistical machine translation. In *Proceedings of the European Machine Translation Conference (EAMT08)*, pages 180–189, Hamburg, Germany.

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.

C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf. 1997. Accelerated DP based search for statistical translation. In *Proceedings of the 5 th European Conference on Speech Communication and Technology*, pages 2667–2670, Rhodes, Greece.

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.