Sign Language Machine Translation Overkill

Daniel Stein, Christoph Schmidt and Hermann Ney

Human Language Technology and Pattern Recognition
RWTH Aachen University, Germany
surname@cs.rwth-aachen.de

Abstract

Sign languages represent an interesting niche for statistical machine translation that is typically hampered by the scarceness of suitable data, and most papers in this area apply only a few, well-known techniques and do not adapt them to small-sized corpora. In this paper, we will propose new methods for common approaches like scaling factor optimization and alignment merging strategies which helped improve our baseline. We also conduct experiments with different decoders and employ state-of-the-art techniques like soft syntactic labels as well as trigger-based and discriminative word lexica and system combination. All methods are evaluated on one of the largest sign language corpora available.

1. Introduction

Sign languages are natural languages with a grammar structure and vocabulary that is distinct from spoken languages. The application of statistical machine translation (SMT) on sign languages is an active field of research, but it is typically hampered due to the scarce resources available.

In this paper, we present recent results on the RWTH-PHOENIX-v3.0 corpus, which is one of the largest corpora available. We want both to analyze features and techniques used by other groups as well as employ our own algorithms. Keeping in mind that most techniques were initially designed for large-scale corpora, we also want to offer new solutions that are tailored to scarce resources. More specifically, we examine how to optimize the scaling factors when holding back a development set would already take away a large portion of the precious training data. We also look at the performance of different alignment merging strategies and propose new methods on how to apply them to small-sized data. In addition, we conduct experiments with different decoders and employ state-of-the-art techniques like soft syntactic labels as well as trigger-based and discriminative word lexica. All results are reported on BLEU [1] and TER [2], and their improvements over the baseline are checked for statistical significance.

1.1. Paper Structure and Related Work

After some preliminaries in Section 2, we begin our experiments with a sanity check in Section 3 to analyze whether a translation process is really necessary. In a recent paper [3], the authors work on the translation of an intermediate, signed form of the Czech language and obtain translation results of up to 81 BLEU, which probably is due to the similarity between this hybrid language and written Czech. We examine whether such a similarity is also true in the case of German Sign Language.

In the next section, we review the optimization procedure. The authors in [4] use factored models on a standard phrase-based system. They report that their data behaves unexpectedly during optimization in that they achieve the best results on their test set if they use the complete training set for the estimation of the scaling factors. We will give a detailed analysis of their procedure in Section 4.

In Section 5, we examine the impact of the alignment that is used for the phrase extraction. We propose to merge the phrase tables obtained from different alignments, introducing new feature functions or to use a system combination to obtain a better translation performance.

In the next section, we perform syntactically motivated experiments. In [5], the authors report that their proposed methods failed to improve a baseline system, and conjecture that this is due to the fact that the syntactic parser produces too many different labels for the small-sized training to work properly. They suggest to use automatic clustering techniques on the data, and we will review this in Section 6.

After this, we train a phrase-based system in Section 7. In [6], a collaborative effort with the Dublin City University, we employed two different decoders but only expressed our intention to combine them. In Section 10, we will perform a system combination to combine the hierarchical and the phrase-based system.

In Section 8, we will review the choice of the error measure and examine whether the standard approach of optimizing on BLEU is actually the best choice. [7] report problems when using the standard error measures in sign language translation. They point out that for small test sets and for unstable data, BLEU is a bad choice as an optimization metric, since e.g. sometimes no correct four-gram can be found. The same authors report in [6] that in more recent experiments the BLEU scores are on reasonable levels again, but leave the question open whether this is due to better data or better machine translation systems.

In Section 9, we will examine the impact of extended

lexicon models on sign language machine translation. In [8], the authors report remarkable improvements by applying extended lexicon models to large-scale machine translation tasks, and we examine whether we can transfer this improvement to sign language translation.

After presenting the final system in Section 10, we conclude our paper in Section 11.

2. Preliminaries

We use the RWTH-PHOENIX-v3.0 corpus, which consists of parallel sentences in German Sign Language and written German, taken from the domain of weather forecast news. These forecasts are part of the main evening news program, which is broadcast on the public television channel "PHOENIX" and which features live interpretation into German Sign Language. The signs are transcribed from video into glosses by human experts, while the German target sentences consist of manually corrected speech recognition output. See Figure 1 for a broadcast news snapshot. The system is part of a larger framework that starts with the automatic recognition of glosses from videos. For sign languages, no official notation system exists and the glosses only form a semantic representation of the meaning of the signs. Glosses are not directly legible and are only used by a minority of the signers, which is why a translation from German Sign Language glosses to written German is necessary.

For translation into the other direction, we refer to [5]. Compared to the previous version of the corpus, the new corpus differs in two aspects: first, it includes newly annotated data from March 2010 until August 2010. Second, in the old version we manually chunked the sentences into smaller, meaningful segments, since broadcast news typically consist of very long, grammatically complicated sentences. In this version we dropped this procedure and took the raw data. The German sentences now have an average length of 14 words, and the glosses have an average length of 10 words per sentence, in old experiments the sentences were roughly only half as long. We further opted for a random 5:1-split into training and test set, so that the test set is comparable to other small-sized evaluation campaign data. See Table 1 for corpus statistics.

A language model is trained with the SRI toolkit[1] using the modified Kneser-Ney discounting for smoothing. The perplexity of the 4-gram language model is 21.4 and does not improve further for larger n-grams. We use this language model consistently throughout the paper.

In this paper, we use a phrase-based and a hierarchical phrase-based decoder and also apply system combination to some hypotheses. In the following, we will describe the systems that we used.

[1] http://www-speech.sri.com/projects/srilm/

Figure 1: Snapshot of the PHOENIX broadcast news

		Glosses	German
Train:	Sentences	2565	
	Running Words	31 208	41 306
	Vocabulary	1 027	1 763
	Singletons	371	641
Test:	Sentences	512	
	Running Words	6 115	8 230
	Vocabulary	570	915
	OOVs	86	133

Table 1: Corpus statistics for the weather forecast corpus

2.1. Phrase-based Translation

We used an in-house phrase-based translation system as described in [9]. The training corpus is word-aligned using GIZA++[2], and phrase pairs consistent with this alignment are extracted. Different models which capture particular aspects of a translation, such as fluency of the output or the accurate translation of individual words, are integrated into a log-linear framework: For a given source sentence f, the system chooses that translation $\hat{e}$ which maximizes the sum over the m different models h_m, weighted by some scaling factors λ_m:

$$\hat{e}(f) := \operatorname*{argmax}_e \left\{ \sum_m \lambda_m h_m(e, f) \right\} . \qquad (1)$$

The models h_m used in the phrase-based translation system are phrase- and word translation probabilities in both directions, a standard n-gram language model, word-, phrase- and distortion penalties and a discriminative reordering model.

2.2. Hierarchical Phrase-based Translation

We also employ our open-source hierarchical translation system JANE, which was officially released in [10]. It is able

[2] http://www.htlpr.rwth-aachen.de/~och/software/GIZA++.html

	BLEU	TER	PER
MT system	18.1	71.0	63.0
simple lower casing	2.1	85.7	81.5
4-letter stems	2.6	81.1	74.8

Table 2: Results for the sanity check (Section 3)

to translate hierarchical phrases and is based on a log-linear model as described in the previous section. Its standard models consist of translation probabilities, IBM-like word lexica, word- and phrase penalty as well as binary markers for hierarchical phrases, paste phrases and glue rules.

2.3. System Combination

For system combination, we use an in-house system as described in [11]. Here, we compute a weighted majority voting on a confusion network, similarly to the well-established ROVER approach for combining speech recognition hypotheses. To create the confusion network, pairwise word alignments of the original MT hypotheses are learned using an enhanced statistical alignment algorithm that explicitly models word reordering. Rather than a single sentence, the context of a whole corpus is taken into account in order to achieve high alignment quality. The confusion network is re-scored with a special language model, and the consensus translation is extracted as the best path.

3. Sanity Check: Lower Case Translation

Before we work on statistical machine translation itself, we first perform a sanity check to see whether the machine translation process is actually necessary. Since sign languages are typically transcribed as glosses that are represented as upper case words of the corresponding spoken language, a casual viewer might question whether the glosses could simply be written in lowercase letters to generate an acceptable output. To show that the languages differ considerably, we compute the translation metrices on the lower-cased glosses. Since the glosses are not conjugated nor inflected in a usual way, we also tried to make a fairer comparison by eliminating the inflection using simple word stems: each word in the hypothesis and the reference was truncated to its first four letters. Figure 2 shows an example sentence pair ("In the evening, the wind turns to the west.") The example shows that the spoken language uses more filler words ("gegen" (about) and the article "der") and that by truncating the words, the conjugation of the verb "drehen" (to turn) could be removed to create a match between Glosses and German.

As a comparison, we set up a hierarchical phrase-based translation system as a baseline system. The system is trained on 2000 sentences and optimized on a development set that was separated from the training data.

The results can be found in Table 2. As expected, the SMT system clearly performs better than simply lowercasing the glosses. For clarity, we also included the position-independent error rate (PER) in the table, which is defined as the percentage of words that were translated incorrectly, regardless of their position in the sentence.

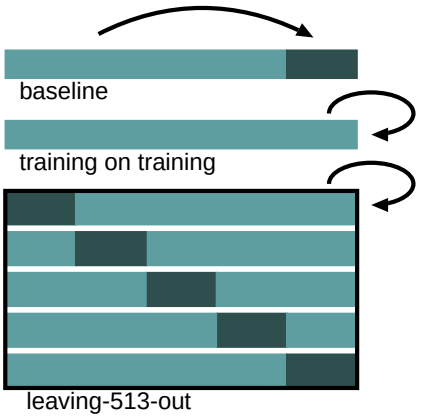

Figure 3: Graphical representation of the different optimization methods (Section 4)

4. On the Choice of the Development Corpus

The purpose of a development set is to obtain scaling factors λ_m for the feature functions so that the translation system generalizes well to unseen data. It seems crucial to keep the development set separate from the training and the test set, which is not a big constraint for normal-sized corpora since the withheld sentences only make up a negligible portion (usually around $.1\%$) of the whole training set. In our case, however, holding back a development set of the same size as our test set strips away 20% of the training material. In this section, we are therefore looking for alternative optimization approaches.

First, we define a traditional split into disjunct training and development sets as our *baseline*. In [4], the authors claim that the best way to optimize the scaling factors on their corpus is to train them on the complete training set, thus not utilizing a development set at all. This approach, which we will denote as *training-on-training*, obviously bears the danger of overfitting. Instead, we propose to have five different translation systems, each trained on a disjoint sub-set of the training corpus. In each optimization iteration, we concatenate the n-best lists of all individual systems for a complete training set translation. We decided to split our training set into five disjoint sets, each excluding 513 sentences, and call this procedure *leaving-513-out*. See Figure 3 for a graphical representation.

The results can be seen in Table 3. In our experiments, the classic approach performs significantly better $(p < .1)$ than the training-on-training method, and the leaving-513-out method is even better $(p < .05)$ than this approach. We

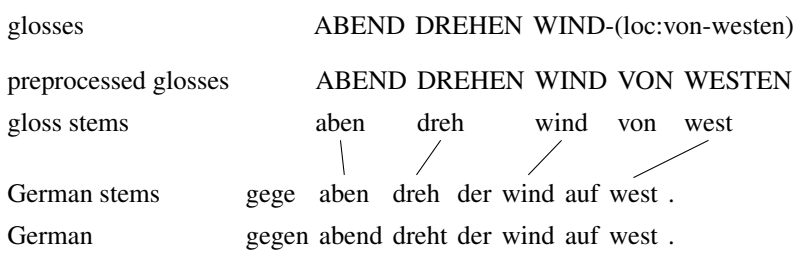

Figure 2: Example sentence of sanity check translation

	BLEU	TER
baseline	18.1	71.0
training-on-training	17.0	74.1
leaving-513-out	19.9	69.7

Table 3: Results for the hierarchical phrase-based decoder on the various development set decisions (Section 4)

are thus not able to reproduce the findings of [4] on our corpus.

5. Harvesting Multiple Alignment Strategies

In this section, we will analyze the impact of the alignment on the translation quality. Probably due to the small data, we noticed that many words are incorrectly aligned, and we expect that thus the errors made at the phrase extraction level are likely to propagate into the translation.

We compute alignments based on the IBM Models with GIZA++. Since the IBM Models are not symmetric, the resulting alignment differs depending on the translation direction. Usually, one alignment is produced for each direction, and the two alignments are merged. We compared the following merging strategies:

intersection Here, we only take alignment points that appear in both alignments.

union We take every alignment point that appears in either alignment.

grow-diag-final-and This algorithm as presented in [12] is probably most widely used. Starting from the intersection alignment, it iteratively extends every alignment point whenever there is a direct neighbor, i.e. when a vertically, horizontally or diagonally adjacent alignment point is part of the union alignment but not yet of the merged alignment. In a finalizing step called *final-and*, we ignore the adjacency and add points from the unification alignment when both source and target row are unaligned.

	precision	recall	F	AER
unify	40.9	62.0	49.3	50.6
intersect	80.1	28.8	42.0	57.6
grow-diag-final-and	44.2	55.2	49.1	50.8
grow-mono-final-and	47.6	45.4	46.5	53.5

Table 4: Precision, recall, F-Measure and alignment error rate (AER) of the different alignment merging strategies (Section 5)

grow-mono-final-and An alternative to this approach is presented in [13], where the possible neighbors are restricted to vertically and horizontally adjacent positions. Further, blocks are avoided by only allowing extension to one direction at a time. We will denote this method as *grow-mono-final-and*.

To obtain alignment quality measures, we hand-aligned 400 sentences to compute precision, recall, the F-Measure and the alignment error rate (AER). See Table 4 for a quality estimation of the various merging strategies on these sentences. Especially the intersection alignment performs very poorly in all measures except the precision score (as could be expected), but for the other three alignments, the F-measure and AER are comparable whereas precision and recall are quite different.

Table 5 reports the translation performance for these settings. As expected, the intersection suffers from a low recall value, but the other systems differ as well. The union alignment has a significantly ($p < .01$) better TER score, but does not perform best in BLEU.

On the basis of these results, we performed three additional experiments in order to bundle synergy effects of the merging strategies. First, we simply concatenated the training corpus four times, but each time with a different alignment. Second, we repeated this procedure but for each phrase pair memorized the percentage that each alignment contributed to it and added this information in four new features into the log-linear model. This means that if a phrase pair was only seen for one alignment, it would have the corresponding feature value of 1 and all others would be set to

	BLEU	TER
union	19.9	69.7
intersect	19.1	80.2
grow-diag-final-and	20.1	72.7
grow-mono-final-and	19.4	72.3
merged phrase table	19.7	71.8
+ features	20.1	68.7
system combination	20.7	68.5

Table 5: Results for the hierarchical phrase-based decoder on the alignment merging strategies (Section 5)

	BLEU	TER
grow-diag-final-and	20.1	72.7
parse-match	19.1	72.9
soft-syntax	19.6	73.4
parse-match + soft-syntax	19.3	72.5

Table 6: Results for the hierarchical phrase-based decoder on the syntactic enhancements (Section 6)

zero, and if it would have been seen equally often in each alignment, all the new features would equal to .25. In this way, the decoder should be able to penalize or favor the alignment strategy that works best. In the last experiment, we combined all these different results and performed a ROVER system combination.

The results are also included in Table 5. Simply merging the phrase tables gave no improvement, and the TER is worse than for some individual systems. By adding the features that marked the source of the alignment for each phrase we still saw no improvements in BLEU and only some small improvement in TER. The best system is the one produced by the system combination, which not only has the best performance on the test set but is actually significantly better than the single grow-diag-final-and system.

6. Linguistically Motivated Experiments

In [5], we applied a syntactic analysis to the target language by extracting additional information with the freely available Stanford parser[3]. We first penalized phrases that did not match the yield of a syntax tree-node, as in [14]. We denote these experiments as *parse-match*. Second, we employed soft syntactic labels as in [15]. With this, we introduce new non-terminal labels into an additional feature function and penalize phrases that do not match the syntactic labels of the non-terminal that they are replacing. This is denoted as *soft-syntax*.

As stated earlier, the corpus is no longer identical to the one used in the old experiments, since the sentence length and the grammar complexity is now higher, which is why we repeat the experiments. See Table 6 for an overview of the results. Based on the results we can conclude that the proposed methods still do not help for this small corpus. In the outlook of the paper, we wrote that the methods probably did not help because of the rather large number of labels, and that automatic clustering techniques could help in this matter. For this, we now cluster the words and phrases into only a few classes by making use of the tools `makecls` [16] and CLUTO [17].

[3] http://nlp.stanford.edu/software/lex-parser.shtml

	BLEU	TER
grow-diag-final-and	20.1	72.7
2 word classes, 2 phrase classes	19.7	72.3
2 word classes, 5 phrase classes	19.7	72.0
5 word classes, 2 phrase classes	19.4	74.8
5 word classes, 5 phrase classes	19.7	71.1

Table 7: Results for the hierarchical phrase-based decoder on the clustering techniques (Section 6)

	BLEU	TER
union	21.6	68.7
intersect	22.1	67.8
grow-diag-final-and	20.8	67.1
grow-mono-final-and	22.2	66.7

Table 8: Results for the phrase-based decoder on the alignment merging strategy (Section 7)

The results can be found in Table 7. While the results are not as deteriorated as the pure syntactic approaches, the methods still do not improve over the baseline. We therefore draw the conclusion that on this small corpus, the methods do not help in a single system and could only be used as an additional system in a system combination approach.

7. Phrase-based translation

As stated in the beginning, we want to employ different decoders in this work and use them in a final system combination. We therefore trained our in-house phrase-based translation system, as described in Section 2.1, on the same data. Routinely, we examined the alignment merging techniques as in Section 5.

The results are summarized in Table 8. In general, the phrase-based decoder outperforms the hierarchical system on this task. We were surprised to find that the alignment merging strategies that work good on the hierarchical system turn out to be bad choices for the phrase-based system and vice-versa.

In general, our findings emphasize that all components

Criterion	Results BLEU	TER	PER
BLEU	22.2	66.7	58.6
TER	19.4	62.8	56.9
BLEU - TER	20.5	62.8	56.4
PER	21.0	63.3	56.6

Table 9: Results for the phrase-based decoder using different optimization criteria (Section 8)

	BLEU	TER
Baseline	22.2	66.7
Triplet	22.5	67.0
DWL	22.1	67.6
DWL+Triplet	22.7	68.0

Table 10: Results for the phrase-based decoder using extended lexicon models (Section 9)

need to be re-examined for different decoders and that conclusions made on one of them do not necessarily carry over. For the phrase-based decoder experiments in Section 8 and Section 9, we proceed to take the *grow-mono-final-and* alignment extraction as our baseline setting.

8. On the Choice of the Training Criterion

Current statistical machine translation systems are usually optimized on BLEU, that is, the scaling factors λ_m of the log-linear model (Eqn. 2.1) are adjusted such that the BLEU score on a development corpus is maximized.

[7] however argued whether the standard metrics such as BLEU, WER or PER are suitable for sign languages, since they were unable to produce interpretable results. More precisely, the BLEU metric could in some instances not find a single 4-gram that was correct and thus reports an overall score of 0.

In the previous sections, we have already shown that an SMT system can be set up and trained using the standard techniques, and the authors themselves have already stated in [6] that with newer data and better translation systems, this problem does no longer exist. However, the question remains which evaluation metric is most suitable for training. In this section, we optimized the phrase-based translation system on the different metrics. The results are summarized in Table 9.

As expected, optimizing the system on BLEU leads to optimal performance with regard to that measure, with all other systems being significantly worse ($p < .01$). Interestingly, among the other systems, the one optimized on PER ranks best, being significantly better than the other systems ($p < .01$). On the contrary, when evaluating on TER and PER, the system optimized on BLEU performed significantly worse ($p < .01$) than all other systems, while their differences are not significant.

9. Extended lexicon models

In [8], the authors could achieve an improvement of about 1% in BLEU over a competitive phrase-based system by including two extended lexicon models. In the following, we will briefly review the two models.

In a standard phrase-based translation model, only local dependencies are taken into account. The phrase-model and the language model limit the dependencies to a phrase or a small number of target words, usually up to 6 words, respectively. The following lexicon models also take into account long range dependencies across the whole source sentence.

9.1. Triplet Model

The well-known IBM-model 1 [18] estimates word translation probabilities $p(e|f)$. The triplet model extends this model by estimating the probability $p(e|f, f')$ of a target word e based on two source words f, f'. Like IBM-model 1, the triplet model is trained iteratively using the EM algorithm. During extraction and decoding, f is the source word aligned to the target word e to be translated, while f' loops over the whole source sentence. Thus, the second source word f' enables the model to make more informed decisions about translating f into e.

9.2. Discriminative Word Lexicon

The discriminative word lexicon uses the whole source sentence to predict target words, thus taking into account global dependencies. It is modeled as a combination of simple classifiers for each word e from the target vocabulary V_E. Each of these classifiers models whether a certain word e is present in the target sentence ($\delta_e = 1$) or not ($\delta_e = 0$), given the set of source words $\mathbf{f}$. The probability of the target sentence is then modeled as the product of all positive classifications over all words in the target sentence times the product of all negative classifications over all words not contained in the target sentence:

$$P(\mathbf{e}|\mathbf{f}) = \prod_{e \in \mathbf{e}} P(\delta_e = 1|\mathbf{f}) \cdot \prod_{e \in \mathbf{V_E} \setminus \mathbf{e}} P(\delta_e = 0|\mathbf{f})$$

One advantage of this model is that the training can be easily parallelized, because the classifiers for each word e can be trained independently. Moreover, the estimation of $p(\delta_e|\mathbf{f})$ seems to be more stable than the estimation of $p(e|f)$

The results of using extended lexicon models are summarized in Table 10. While the triplet model and its combination with the DWL model lead to slight improvements over the baseline in terms of BLEU, the TER score gets worse when applying the extended lexicon models. However, none of the differences are statistically significant. It seems that

in the case of small corpora like sign language translation, the extended lexicon models tend to help less than on large corpora.

10. Final System

The final system is a system combination of three hierarchical and three standard phrase-based systems. For hierarchical systems, we used the grow-diag-final-and baseline system, the soft-syntax system and the system with 5 word and phrase clusters. For the phrase-based system, we used the grow-mono-final-and system, the triplet enhanced system and the DWL system.

See Table 11 for an overview of the systems that were chosen and of the results of the system combination. We managed to get a significant improvement over both baseline decoders, with a gain of 3.3 BLEU and 7.2 TER if compared to the hierarchical baseline system, and still a gain of 1.2 in BLEU and TER when compared to the phrase-based baseline system.

Figure 4 shows that for some sentences, the translation quality has increased on a semantic and grammatic level. Since the reference sentence does not mirror these improvements, we plan to employ multiple references in upcoming experiments to better measure the performance.

11. Conclusion

In this paper, we presented and analyzed various statistical translation techniques on a small sign language corpus. We reviewed several papers in the area and checked if the findings hold true for our corpus. Typical procedures like scaling factor optimization and selecting the best alignment work reasonably well, but can be improved with properly tailored methods. More sophisticated algorithms like syntactic enhancements failed to improve over the baseline. Moreover, clustering techniques and discriminative training did not show significant improvement. However, we were able to make use of all the single systems and produce the best system using a ROVER-like combination.

Overall, we tried to cover some open issues that came up in the recent literature on sign language translation. It might be interesting to see whether these findings hold true for other under-resourced language pairs. Generally speaking, though, we believe that it is worthwhile to "overkill" these small corpora, since a lot of different approaches help for a better translation quality and since the experiments run quite fast due to the limited size of the training material.

12. Acknowledgments

We would like to express our gratitude to Jens Forster, Markus Nußbaum-Thom,Philipp Szymanski, and Uwe Zelle for their help in the corpus creation.

This work has been partly funded by the European Community's Seventh Framework Programme (FP7-ICT-2007-3. Cognitive Systems, Interaction, Robotics - STREP) under grant agreement n° 231424.

13. References

[1] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, July 2002, pp. 311–318.

[2] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA, August 2006, pp. 223–231.

[3] J. Kanis and L. Müller, "Advances in Czech – Signed Speech Translation," in *Lecture Notes in Computer Science*, vol. 5729. Springer, 2009, pp. 48–55.

[4] G. Massó and T. Badia, "Dealing with sign language morphemes for statistical machine translation," in *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, Valletta, Malta, May 2010, pp. 154–157.

[5] D. Stein, J. Forster, U. Zelle, P. Dreuw, and H. Ney, "Analysis of the german sign language weather forecast corpus," in *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, Valletta, Malta, May 2010, pp. 225–230.

[6] S. Morrissey, A. Way, D. Stein, J. Bungeroth, and H. Ney, "Towards a hybrid data-driven mt system for sign languages," in *Machine Translation Summit*, Copenhagen, Denmark, Sept. 2007, pp. 329–335.

[7] S. Morrissey and A. Way, "Lost in translation: the problems of using mainstream mt evaluation metrics for sign language translation," in *Proceedings of the 5th SALTMIL Workshop on Minority Languages at LREC'06*, Genoa, Italy, 2006, pp. 91–98.

[8] A. Mauser, S. Hasan, and H. Ney, "Extending Statistical Machine Translation with Discriminative and Trigger-Based Lexicon Models," in *Conference on Empirical Methods in Natural Language Processing*, Singapore, Aug. 2009, pp. 210–218.

[9] R. Zens and H. Ney, "Improvements in Dynamic Programming Beam Search for Phrase-based Statistical Machine Translation," in *International Workshop on Spoken Language Translation*, Honolulu, Hawaii, Oct. 2008, pp. 195–205.

decoder	system	BLEU	TER
hierarchical system	grow-diag-final-and	20.1	72.7
	soft-syntax	19.6	73.4
	5 word and phrase classes	19.7	71.1
phrase-based system	grow-mono-final-and	22.2	66.7
	+ Triplet	22.5	67.0
	+ DWL + Triplet	22.7	68.0
both	system combination	23.4	65.5

Table 11: Results of the final system combination (Section 10)

Hierarchical	gegen abend verschmilzt .
Phrase-based	gegen abend mit regen nach .
System Combination	gegen abend laesst der regen nach .
Reference	gegen abend lassen die schauer nach .
Hierarchical	jahres bis 1500 meter schneeverwehungen gerechnet werden .
Phrase-based	am alpenrand 1500 metern schnee .
System Combination	an den alpen , oberhalb von 1500 metern schnee .
Reference	in den alpen liegt die schneefallgrenze bei 1500 metern .

Figure 4: Example sentences: Improvements by system combination

[10] D. Vilar, D. Stein, M. Huck, and H. Ney, "Jane: Open Source Hierarchical Translation, Extended with Reordering and Lexicon Models," in *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July 2010, pp. 262–270.

[11] E. Matusov, G. Leusch, R. E. Banchs, N. Bertoldi, D. Dechelotte, M. Federico, M. Kolss, Y.-S. Lee, J. B. Marino, M. Paulik, S. Roukos, H. Schwenk, and H. Ney, "System combination for machine translation of spoken and written language," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 7, pp. 1222–1237, Sept. 2008.

[12] P. Koehn, F. J. Och, and D. Marcu, "Statistical Phrase-Based Translation," in *Proceedings of the Human Language Technology, North American Chapter of the Association for Computational Linguistics*, Edmonton, Canada, May 2003, pp. 54–60.

[13] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[14] D. Vilar, D. Stein, and H. Ney, "Analysing Soft Syntax Features and Heuristics for Hierarchical Phrase Based Machine Translation," in *International Workshop on Spoken Language Translation*, Waikiki, Hawaii, Oct. 2008, pp. 190–197.

[15] A. Venugopal, A. Zollmann, N. Smith, and S. Vogel, "Preference Grammars: Softening Syntactic Constraints to Improve Statistical Machine Translation," in *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, June 2009, pp. 236–244.

[16] F. J. Och, "An efficient method for determining bilingual word classes," in *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, June 1999, pp. 8–12.

[17] Y. Zhao and G. Karypis, "Clustering in Life Sciences," *Functional Genomics: Methods and Protocols*, vol. 224, pp. 183–218, 2003.

[18] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, June 1993.