Phrase Model Training for Statistical Machine Translation with Word Lattices of Preprocessing Alternatives

Joern Wuebker and Hermann Ney
Human Language Technology and Pattern Recognition Group
RWTH Aachen University
Aachen, Germany
{wuebker,ney}@cs.rwth-aachen.de

Abstract

In statistical machine translation, word lattices are used to represent the ambiguities in the preprocessing of the source sentence, such as word segmentation for Chinese or morphological analysis for German. Several approaches have been proposed to define the probability of different paths through the lattice with external tools like word segmenters, or by applying indicator features. We introduce a novel lattice design, which explicitly distinguishes between different preprocessing alternatives for the source sentence. It allows us to make use of specific features for each preprocessing type and to lexicalize the choice of lattice path directly in the phrase translation model. We argue that forced alignment training can be used to learn lattice path and phrase translation model simultaneously. On the news-commentary portion of the German→English WMT 2011 task we can show moderate improvements of up to 0.6% BLEU over a state-of-the-art baseline system.

1 Introduction

The application of statistical machine translation (SMT) to word lattice input was first introduced for the translation of speech recognition output. Rather than translating the single-best transcription, the speech recognition system encodes all possible transcriptions and their probabilities within a word lattice, which is then used as input for the machine translation system (Ney, 1999; Matusov et al., 2005; Bertoldi et al., 2007).

Since then, several groups have adapted this approach to model ambiguities in representing the source language with lattices and were able to report improvements over their respective baselines. The probabilities for different paths through the lattice are usually modeled by assigning probabilities to arcs as a byproduct of the lattice generation or by defining binary indicator features. Applying the first method only makes sense if the lattice construction is based on a single, comprehensive probabilistic method, like a Chinese word segmentation model as is used by Xu et al. (2005). In applications like the one described by Dyer et al. (2008), where several different segmenters for Chinese are combined to create the lattice, this is not possible. Also, our intuition suggests that simply defining indicator features for each of the segmenters may not be ideal, if we assume that there is not a single best segmenter, but rather that for different data instances a different one works best.

In this paper, we propose to model the lattice path implicitly within the phrase translation model. We introduce a novel lattice design, which explicitly distinguishes between different ways of preprocessing the source sentence. It enables us to define specific binary features for each preprocessing type and to learn lexicalized lattice path probabilities and the phrase translation model simultaneously with a forced alignment training procedure.

To train the phrase translation model, most state-of-the-art SMT systems rely on heuristic phrase extraction from a word-aligned training corpus. Using a modified version of the translation decoder to

Proceedings of the 7th Workshop on Statistical Machine Translation, pages 450–459,
Montréal, Canada, June 7-8, 2012. ©2012 Association for Computational Linguistics

force-align the training data provides a more consistent way of training. Wuebker et al. (2010) introduce a leave-one-out method which can overcome the over-fitting effects inherent to this training procedure (DeNero et al., 2006). The authors report this to yield both a significantly smaller phrase table and higher translation quality than the heuristic phrase extraction.

We argue that applying forced alignment training helps to exploit the full potential of word lattice translation. The effects of the training on lattice input are analyzed on the news-commentary portion of the German→English WMT 2011 task. Our results show moderate improvements of up to 0.6% BLEU over the baseline.

This paper is organized as follows: We will review related work in Section 2, describe the decoder in Section 3 and present our novel lattice design in Section 4. The phrase training algorithm is introduced in Section 5, and Section 6 gives a detailed account of the experimental setup and discusses the results. Finally, our findings are summarized in Section 7.

2 Related work

Word lattices have been used for machine translation of text in a variety of ways. Dyer et al. (2008) use it to encode different Chinese word segmentations or Arabic morphological analyses. For the phrase-based model, they report improvements of up to 0.9% BLEU for Chinese→English and 1.6% BLEU for Arabic→English over the respective single best word segmented and morphologically analyzed source. These results are achieved without an explicit way of modeling probabilities for different paths within the lattice. The training of the phrase model is done by generating one version of the training data for each segmentation method or morphological analysis. The word alignments are trained separately, and are then concatenated for phrase extraction. Our work differs from (Dyer et al., 2008) in that we explicitly distinguish the various preprocessing types in the lattice so that we can define specific path features and lexicalize the lattice path probabilities within the phrase model.

In (Xu et al., 2005) the probability of a segmentation, as given by the Chinese word segmentation model, and the translation model are combined into a global decision rule. This is done by weighting the lattice edges with a source language model. The authors report an improvement of 1.5% BLEU over translation of the single best segmentation with a phrase-based SMT system.

Dyer (2009) introduces a maximum entropy model for compound word splitting, which he uses to create word lattices for translation input. He shows improvements in German-English, Hungarian-English and Turkish-English over state-of-the-art baselines.

For the German→English WMT 2010 task, Hardmeier et al. (2010) encode the morphological reduction and decompounding of the German surface form as alternative paths in a word lattice. They show improvements of roughly 0.5% BLEU over the baseline. A binary indicator feature is added to the log-linear framework for the alternative edges. Additionally, they integrate long-range reorderings of the source sentence into the lattice, in order to match the word order of the English language, which yields another improvement of up to 0.5% BLEU.

Niehues and Kolss (2009) also use lattices to encode different alternative reorderings of the source sentence which results in an improvement of 2.0% BLEU over the baseline on the WMT 2008 German→English task.

Onishi et al. (2010) propose a method of modeling paraphrases in a lattice. They perform experiments on the English→Japanese and English→Chinese IWSLT 2007 tasks, and report improvements of 1.1% and 0.9% BLEU over a paraphrase-augmented baseline.

Schroeder et al. (2009) generalize usage of lattices to combine input from multiple source languages.

Factored translation models (Koehn and Hoang, 2007) approach the idea of integrating annotation into translation from the opposite direction. Where lattices allow the decoder to choose a single level of annotation as translation source, factored models are designed to jointly translate several annotation levels (factors). Thus, they are more suited to integrate low-level annotation that by itself does not provide sufficient information for accurate translation, like part-of-speech tags, gender, etc. On the other hand, they require a one-to-one correspondence between the factors, which makes them unsuitable to model word segmentation or decompounding.

The problem of performing real training for the phrase translation model has been approached in a number of different ways in the past. The first one, to the best of our knowledge, was the joint probability phrase model presented by Marcu and Wong (2002). It is shown to perform slightly inferior to the standard heuristic phrase extraction from word alignments by Koehn et al. (2003).

A detailed analysis of the inherent over-fitting problems when training a generative phrase model with the EM algorithm is given in (DeNero et al., 2006). These findings are in principle confirmed by Moore and Quirk (2007) who, however, can show that their model is less sensitive to reducing computational resources than the state-of-the-art heuristic.

Birch et al. (2006) and DeNero et al. (2008) present alternative training procedures for the joint model introduced by Marcu and Wong (2002), which are shown to improve its performance.

In (Mylonakis and Sima'an, 2008) a phrase model is described, whose training procedure is designed to counteract the inherent over-fitting problem by including prior probabilities based on Inversion Transduction Grammar and smoothing as learning objective. It yields a small improvement over a standard phrase-based baseline.

Ferrer and Juan (2009) present an approach, where the phrase model is trained by a semi-hidden Markov model.

In this work we apply the phrase training method introduced by Wuebker et al. (2010), where the phrase translation model of a fully competitive SMT system is trained in a generative way. The key to avoiding the over-fitting effects described by DeNero et al. (2006) is their novel leave-one-out procedure.

3 Decoding

3.1 Phrase-based translation

We use a standard phrase-based decoder which searches for the best translation $\hat{e}_1^{\hat{I}}$ for a given input sentence f_1^J by maximizing the posterior probability

$$\hat{e}_1^{\hat{I}} = \arg\max_{I,e_1^I} Pr(e_1^I|f_1^J). \qquad (1)$$

Generalizing the noisy channel approach (Brown et al., 1990) and making use of the maximum approximation (Viterbi), the decoder directly models the posterior probability by a log-linear combination of several feature functions $h_m(e_1^I, s_1^K, f_1^J)$ weighted with scaling factors λ_m, which results in the decision rule (Och and Ney, 2004)

$$\hat{e}_1^{\hat{I}} = \arg\max_{I,e_1^I,K,s_1^K} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right\}. \qquad (2)$$

Here, s_1^K denotes the segmentation of e_1^I and f_1^J into K phrase-pairs and their alignment. The features used are the language model, phrase translation and lexical smoothing models in both directions, word and phrase penalty and a simple distance-based reordering penalty.

3.2 Lattice translation

For lattice input we generalize Equation 2 to also maximize over the set of sentences $\mathcal{F}(\mathcal{L})$ encoded by a given source word lattice $\mathcal{L}$:

$$\hat{e}_1^{\hat{I}} = \arg\max_{I,e_1^I,K,s_1^K,f_1^J\in\mathcal{F}(\mathcal{L})} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right\} \qquad (3)$$

Note that in this formulation there are no probabilities assigned to the arcs of $\mathcal{L}$. We define additional binary indicator features h_m and lexicalize path probabilities by encoding the path into the word identities. To translate lattice input, we adapt the standard phrase-based decoding algorithm as described in (Matusov et al., 2008). The decoder keeps track of the covered *slots*, which represent the topological order of the nodes, rather than the covered words. When expanding a hypothesis, it has to be verified that there is no overlap between the covered nodes and that a path exists from start to goal node,

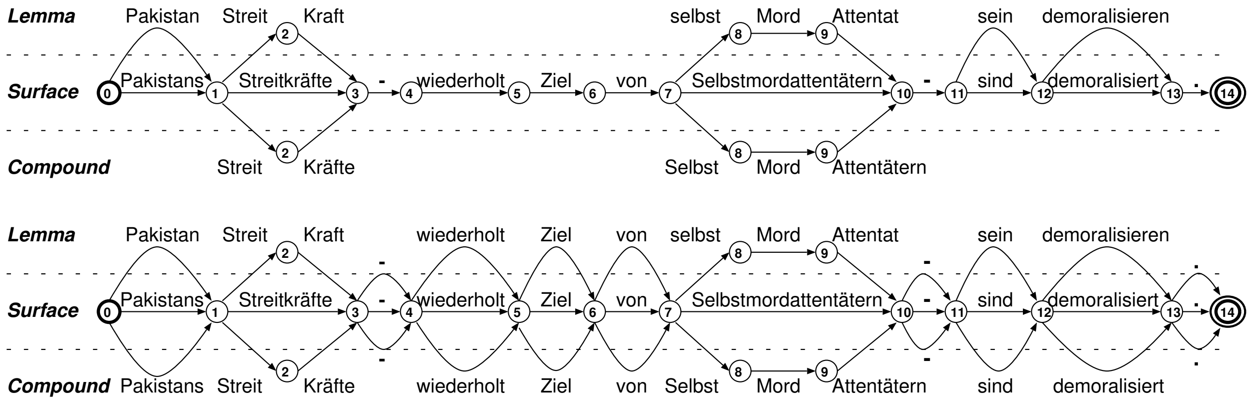

Figure 1: Top: Slim lattice. Bottom: Full lattice. The sentence is taken from the training data. The three layers *Surface*, *Compound* and *Lemma* are separated with dashed lines. Nodes are labeled with slot information. Slots are ordered horizontally, layers vertically.

which passes through all covered nodes. In practice, when considering a possible expansion covering slots $j', ..., j''$ with start and end states n' and n'', we make sure that the following two conditions hold:

- n' is reachable from the lattice node that corresponds to the nearest already covered slot to the left of j'.
- The node that corresponds to the nearest already covered slot to the right of j'' is reachable from n''.

It was noted by Dyer et al. (2008) that the standard distance-based reordering model needs to be redefined for lattice input. We define the distortion penalty as the difference in slot number. Using the shortest path within the lattice is reported to have better performance in (Dyer et al., 2008), however we did not implement it due to time constraints.

4 Lattice design

We construct lattices from three different preprocessing variants of the German source side of the data. The surface form is the standard tokenization of the source sentence. The word compounds are produced by the frequency-based compound splitting method described in (Koehn and Knight, 2003), applied to the tokenized sentence. From the compound split sentence we produce the lemma of the German words by applying the TreeTagger toolkit (Schmid, 1995). Each of the different preprocessing variants is assigned a separate *layer* within the lattice. For the phrase model, word identities are defined by both the word and its layer. In this way, the phrase model can assign different scores to phrases in different layers, allowing it to guide the search towards a specific layer for each word. In practice, this is done by annotating words with a unique identifier for each layer. For example, the word *sein* from the lemmatized layer will be written as *LEM.sein* within both the data and the phrase table. If *sein* appears in the surface form layer, it will be written as *SUR.sein* and is treated as a different word. *SUR* is the identifier for the compound layer.

We experiment with two different lattice designs. In the *full* lattice, all three layers are included for each source word in surface form. The *slim* lattice only includes arcs for the lemma layer if it differs from the surface form, and only includes arcs for the compound layer if it differs from both surface form and lemma. Figure 1 shows a slim and a full lattice for the same training data sentence.

For each layer, we add two indicator features to the phrase table: One binary feature which is set to 1 if the phrase is taken from this layer, and one feature which is equal to the number of words from this layer. This results in six additional feature functions, whose weights are optimized jointly with the standard features described in Section 3.1. We will denote them as *layer features*.

5 Phrase translation model training

To train the phrase model, we use a modified version of the translation decoder to force-align the training data. We apply the method described in (Wuebker et al., 2010), but with word lattices on the source side. To avoid over-fitting, we use their cross-validation technique, which is described as a low-cost alternative to leave-one-out. For cross-validation we segment the training data into batches containing 5000 sentences. For each batch, the phrase table is updated by reducing the phrase counts by the local counts produced by the current batch in the previous training iteration. For the first iteration, we perform the standard phrase extraction separately for each batch to produce the local counts. Singleton phrases are assigned the probability $\beta^{(|\tilde{f}|+|\tilde{e}|)}$ with the source and target phrase lengths $|\tilde{f}|$ and $|\tilde{e}|$ and fixed $\beta = e^{-5}$ (length-based leave-one-out). Sentences for which the decoder is not able to find an alignment are discarded (about 4% for our experiments). To estimate the probabilities of the phrase model, we count all phrase pairs used in training within an n-best list (equally weighted). The translation probability for a phrase pair $(\tilde{f}, \tilde{e})$ is estimated as

$$p_{FA}(\tilde{e}|\tilde{f}) = \frac{C_{FA}(\tilde{f}, \tilde{e})}{C_{mon}(\tilde{f})}, \qquad (4)$$

where $C_{FA}(\tilde{f}, \tilde{e})$ is the count of the phrase pair $(\tilde{f}, \tilde{e})$ in the force-aligned training data. In order to learn the lattice path along with the phrase translation probabilities, we make the following modification to the original formulation in (Wuebker et al., 2010). The denominator $C_{mon}(\tilde{f})$ is the count of $\tilde{f}$ in the target side of the training data, rather than using the real marginal counts. This means that it is independent of the training procedure, and can be computed by ignoring one side of the training data and performing a simple n-gram count on the other. In this way the model learns to prefer lattice paths which are taken more often in training. For example, if the phrase *(LEM.Streit LEM.Kraft)* is used to align the sentence from Figure 1, $C_{mon}(\tilde{f})$ will be increased for $\tilde{f}$ = *(SUR.Streitkräfte)* and $\tilde{f}$ = *(SPL.Streit SPL.Kräfte)* without affecting their joint counts. This leads to a lower probability for these phrases, which is not the case if marginal counts are used. Note that on the source side we have one training corpus for each lattice layer, which are concatenated to compute $C_{mon}(\tilde{f})$. The size of the n-best lists used in this work is fixed to 20000. Using smaller n-best lists was tested, but seems to have disadvantages for the application to lattices. After reestimation of the phrase model, the feature weights are optimized again.

In order to achieve a good coverage of the training data, we allow the decoder to generate *backoff phrases*. If a source phrase consisting of a single word does not have any translation candidates left after the bilingual phrase matching, one phrase pair is added to the translation candidates for each word in the target sentence. The backoff phrases are assigned a fixed probability $\gamma = e^{-12}$. Note that this is smaller than the probability the phrase would be assigned according to the length-based leave-one-out heuristic, leading to a preference of singleton phrases over backoff phrases. The lexical smoothing models are applied in the usual way to both singleton and backoff phrases. After each sentence, the backoff phrases are discarded. However, in the experiments for this work, introducing backoff phrases only increases the coverage from 95.8% to 96.2% of the sentences.

6 Experimental evaluation

6.1 Experimental setup

Our experiments are carried out on the news-commentary portion of the German→English data provided for the *EMNLP 2011 Sixth Workshop on Statistical Machine Translation* (WMT 2011).[*] We use `newstest2008` as development set and `newstest2009` and `newstest2010` as unseen test sets. The word alignments are produced with GIZA++ (Och and Ney, 2003). To optimize the log-linear parameters, the Downhill-Simplex algorithm (Nelder and Mead, 1965) is applied with BLEU (Papineni et al., 2002) as optimization criterion. The

[*] `http://www.statmt.org/wmt11`

		German			English
		Surface	Compound	Lemma	
Train	Sentences	136K			
	Running Words	3.4M	3.5M		3.3M
	Vocabulary Size	118K	81K	52K	57K
newstest2008	Sentences	2051			
	Running Words	48K	50K		50K
	Vocabulary Size	10.3K	9.7K	7.3K	8.1K
	OOVs (Running Words)	3041	2092	1742	2070
newstest2009	Sentences	2525			
	Running Words	63K	66K		66K
	Vocabulary Size	12.2K	11.4K	8.4K	9.4K
	OOVs (Running Words)	4058	2885	2400	2729
newstest2010	Sentences	2489			
	Running Words	62K	65K		62K
	Vocabulary Size	12.3K	11.4K	8.5K	9.2K
	OOVs (Running Words)	4357	2952	2565	2742

Table 1: Corpus Statistics for the WMT 2011 news-commentary data, the development set (`newstest2008`) and the two test sets (`newstest2009`, `newstest2010`). For the source side, three different preprocessing alternatives are included: Surface, Compound and Lemma.

language model is a standard 4-gram LM with modified Kneser-Ney smoothing (Chen and Goodman, 1998) produced with the SRILM toolkit (Stolcke, 2002). It is trained on the full bilingual data and parts of the monolingual News crawl corpus provided for WMT 2011. Numbers are replaced with a single category symbol in a separate preprocessing step and we apply the long-range part-of-speech based reordering rules proposed by (Popović and Ney, 2006).

Table 1 shows statistics for the bilingual training data and the development and test corpora for the three different German preprocessing alternatives. It can be seen that both compound splitting and lemmatization reduce the vocabulary size and number of out-of-vocabulary (OOV) words. Results are measured in BLEU and TER (Snover et al., 2006), which are computed case-insensitively with a single reference.

6.2 Baseline experiments

To get an overview over the effects of the different preprocessing alternatives for the German source, we built three baseline systems, one for each preprocessing type. The phrase tables are extracted heuristically in the standard way from the word-aligned training data. Additionally, we performed phrase training for the compound split version of the data. The results are shown in Table 2. When moving from the Surface to the Compound layer, we observe improvements of up to 1.0% in BLEU and 1.1% in TER. Reducing the morphological richness further (Lemma) leads to a clear performance drop. Application of phrase training on the compound split data yields a small degradation in TER on all data sets and in BLEU on `newstest2010`. We assume that this is due to the small size of the training data and its heterogeneity, which makes it hard for the decoder to find good phrase alignments.

6.3 Lattice experiments: Heuristic extraction

We generated both slim and full lattices for all data sets. Similar to (Dyer et al., 2008), we concatenate the three training data sets and their word alignments to extract the phrases. Note that this only produces single-layer phrases. It can be seen in Table 2 that without the application of layer features the slim lattice slightly outperforms the full lattice. In-

		newstest2008		newstest2009		newstest2010	
		Bleu$^{[\%]}$	Ter$^{[\%]}$	Bleu$^{[\%]}$	Ter$^{[\%]}$	Bleu$^{[\%]}$	Ter$^{[\%]}$
Baseline	Surface	19.5	64.6	18.6	64.4	20.6	62.8
	Compounds	20.5	**63.5**	19.1	63.5	21.1	61.9
	FA Compounds	20.5	63.9	19.1	63.8	20.9	62.3
	Lemma	19.2	65.4	18.2	65.2	19.9	63.9
Slim Lattice	without layer feat.	19.9	64.4	18.9	64.1	20.8	62.6
(heuristic)	with layer feat.	20.5	63.8	19.4	63.9	21.0	62.4
Full Lattice	without layer feat.	19.8	64.6	18.7	64.2	20.6	62.8
(heuristic)	with layer feat.	20.4	64.0	19.5	63.8	21.3	62.3
Full Lattice	without layer feat.	20.0	64.3	19.3	64.1	20.8	62.6
(FA w/o layer feat.)	with layer feat.	20.2	64.3	19.1	64.2	20.7	62.8
Full Lattice	without layer feat.	20.5	63.7	19.5	63.6	21.3	62.1
(FA w/ layer feat.)	with layer feat.	**20.7**	63.6	**19.7**	**63.4**	**21.4**	**61.8**

Table 2: Results on the German-English WMT 2011 data. Scores are computed case-insensitively for Bleu [%] and Ter [%]. We evaluate performance of the baseline systems, one for each of the three different encodings, with both slim and full lattices using heuristic phrase extraction and with full lattices using forced alignment phrase model training (*FA*). All lattice systems are evaluated with and without layer features. The best scores in each column are in boldface, statistically significant improvement over the *Compounds* baseline is marked with blue color.

troducing layer features boosts the performance for both lattice types. However, the performance increase is considerably larger for the full lattice systems, which now outperform the slim lattice systems on newstest2009 and newstest2010. Compared to the *Compounds* baseline, the full lattice system with layer features shows a small improvement of up to 0.4% Bleu on newstest2009 and newstest2010, but a degradation in Ter.

6.4 Lattice experiments: Phrase training

The experiments on phrase training are setup as follows. The phrase table is initialized with the standard extraction and is identical to the one used for the experiments in Section 6.3. The log-linear scaling factors used in training are the optimized parameters on the corresponding lattice, also taken from the experiments described in Section 6.3. The forced alignment procedure was run for one iteration. Further iterations were tested, but did not give any improvements.

The phrase training was performed on the full lattice design. The reason for this is that we want the system to learn all possible phrases. Even if there is no difference in wording between the layers in training, the additional phrases could be useful for unseen test data. The training was performed both with and without layer features. The resulting systems were also optimized with and without layer features, resulting in four different setups.

From the results in Table 2 it is clear that phrase training without layer features does not have the desired effect. Even if we apply layer features to the system trained without them, we do not reach the performance of the best standard lattice system. We conclude that, without these indicator features, the standard lattice system does not produce good phrase alignments.

When the layer features are applied for both training and translation, we observe improvements of up to 0.2% in Bleu and 0.5% in Ter over the corresponding standard lattice system. The gap between the systems with and without layer features is much smaller than for the heuristically trained lattices. This indicates that our goal of encoding the best lattice path directly in the phrase model was at least partially achieved. However, in order to exceed the performance of our state-of-the-art baseline on both measures, the layer features are still needed within the phrase training procedure and for translation. Al-

source	Das Warten hat gedauert mehr als NUM Minuten, was im Fall einer Straße, wo werden erwartet NUM Menschen, **ist unverständlich.**
reference	The wait lasted more than NUM minutes, something incomprehensible for a race where you expect more than NUM people.
lattice (heuristic)	The wait has taken more than NUM minutes, which in the case of a street, where NUM people are expected to be, **can't understand it.**
lattice (FA)	The wait has taken more than NUM minutes, which in the case of a street, where expected NUM people, **is incomprehensible.**

Figure 2: Example sentence from the `newstest2009` data set. The faulty phrase in the heuristic lattice translation is marked in boldface.

together, our phrase trained lattice approach outperforms the state-of-the-art baseline on all three data sets by up to 0.6% BLEU. On `newstest2009`, this result is statistically significant with 95% confidence according to the bootstrap resampling method described by Koehn (2004).

For a direct comparison between the heuristic and phrase-trained full lattice systems, we manually inspected the optimized log-linear parameter values for the layer features. We observe that for the standard lattices, paths through the lemmatized layer are heavily penalized. In the phrase trained lattice setup, the penalty is much smaller. As a result, the number of words from the Lemma layer used for translation of the `newstest2009` data set is increased by 49% from 1828 to 2715 words. However, a manual inspection of the translations reveals that the main improvement seems to come from a better choice of phrases from the Compound layer. More specifically, the used phrases tend to be shorter – the average phrase length of Compound layer phrases is 1.5 words for both the baseline and the heuristic lattice system. In the phrase trained lattice system, it is 1.3 words. An example is given in Figure 2. We focus on the end of the sentence, where the heuristic system uses the rather disfluent phrase (ist unverständlich. # can't understand it.), whereas the forced alignment trained system applies the three phrases (ist # is), (unverständlich # incomprehensible) and (. # .).

This effect can be explained by the leave-one-out procedure. As lemmatized phrases usually map to several phrases in the other layers, their count is generally higher. Application of leave-one-out, which reduces the counts of all phrases extracted from the current sentence by a fixed value, therefore has a stronger penalizing effect on Surface and Compound layer phrases. In the extreme case, phrases which are singletons in the Compound layer are unlikely to be used at all in training, if the corresponding phrase in the Lemma layer has a higher count. While this rarely leads to the competing lemmatized phrases being used in free translation, it allows for shorter, more general phrases from the more expressive layers to be applied. Indeed, the 'bad' phrase (ist unverständlich. # can't understand it.) from the example in Figure 2 is a singleton.

7 Conclusion and future work

In this work we apply a forced alignment phrase training technique to input word lattices in SMT for the first time. The goal of encoding better lattice path probabilities directly into the phrase model was at least partially successful. The proposed method outperforms our baseline by up to 0.6% BLEU. To achieve this, we presented a novel lattice design, which distinguishes between different *layers*, for which we can define separate indicator features. Although these layer features are still necessary for the final system to improve over state-of-the-art performance, they are less important than in the heuristically trained setup.

One advantage of our approach is its adaptability to a variety of scenarios. In future work, we plan to apply it to additional language pairs. Arabic and Chinese on the source side, where the layers could represent different word segmentations, seem a natural choice. We also hope to be able to leverage larger training data sets. As a natural extension we plan to allow learning of cross-layer phrases. Further, applying this framework to lattices modeling different reorderings could be an interesting direction.

Acknowledgments

This work was partially realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation, and also partially funded by the European Union under the FP7 project T4ME Net, Contract No. 249119.

References

N. Bertoldi, R. Zens, and M. Federico. 2007. Speech translation by confusion network decoding. In *Proceedings of ICASSP 2007*, pages 1297–1300, Honolulu, Hawaii, April.

Alexandra Birch, Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Constraining the phrase-based, joint probability statistical translation model. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 154–157, Jun.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16:79–85, June.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, Aug.

John DeNero, Dan Gillick, James Zhang, and Dan Klein. 2006. Why Generative Phrase Models Underperform Surface Heuristics. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 31–38, New York City, June.

John DeNero, Alexandre Buchard-Côté, and Dan Klein. 2008. Sampling Alignment Structure under a Bayesian Translation Model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 314–323, Honolulu, October.

C. Dyer, S. Muresan, and P. Resnik. 2008. Generalizing word lattice translation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 1012–1020, Columbus, Ohio, June.

Chris Dyer. 2009. Using a maximum entropy model to build segmentation lattices for mt. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the ACL*, pages 406–414, Boulder, Colorado, June.

Jesús-Andrés Ferrer and Alfons Juan. 2009. A phrase-based hidden semi-markov approach to machine translation. In *Proceedings of European Association for Machine Translation (EAMT)*, Barcelona, Spain, May. European Association for Machine Translation.

C. Hardmeier, A. Bisazza, and M. Federico. 2010. FBK at WMT 2010: Word Lattices for Morphological Reduction and Chunk-based Reordering. In *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR*, pages 88–92, Uppsala, Sweden, July.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876.

P. Koehn and K. Knight. 2003. Empirical Methods for Compound Splitting. In *Proceedings of European Chapter of the ACL (EACL 2009)*, pages 187–194.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. pages 388–395, Barcelona, Spain, July.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, July.

E. Matusov, H. Ney, and R. Schlüter. 2005. Phrase-based translation of speech recognizer word lattices using loglinear model combination. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 110–115, San Juan, Puerto Rico.

Evgeny Matusov, Björn Hoffmeister, and Hermann Ney. 2008. ASR Word Lattice Translation with Exhaustive Reordering is Possible. In *Interspeech*, pages 2342–2345, Brisbane, Australia, September.

Robert C. Moore and Chris Quirk. 2007. An iteratively-trained segmentation-free phrase translation model for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 112–119, Prague, June.

Markos Mylonakis and Khalil Sima'an. 2008. Phrase Translation Probabilities with ITG Priors and Smoothing as Learning Objective. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 630–639, Honolulu, October.

J.A. Nelder and R. Mead. 1965. The Downhill Simplex Method. *Computer Journal*, 7:308.

H. Ney. 1999. Speech translation: Coupling of recognition and translation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 517–520, Phoenix, Arizona, USA, March.

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, March.

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

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, December.

T. Onishi, M. Utiyama, and E. Sumita. 2010. Paraphrase lattice for statistical machine translation. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 1–5, Uppsala, Sweden, July.

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 on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

M. Popović and H. Ney. 2006. POS-based Word Reorderings for Statistical Machine Translation. In *International Conference on Language Resources and Evaluation*, pages 1278–1283.

H. Schmid. 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50, Dublin, Ireland, March.

Josh Schroeder, Trevor Cohn, and Philipp Koehn. 2009. Word lattices for multi-source translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 719–727, Athens, Greece.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA*, pages 223–231, Aug.

A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. Int. Conf. on Spoken Language Processing*, volume 2, pages 901 – 904, Denver, Colorado, USA, September.

Joern Wuebker, Arne Mauser, and Hermann Ney. 2010. Training phrase translation models with leaving-one-out. In *Proceedings of the 48th Annual Meeting of the Assoc. for Computational Linguistics*, pages 475–484, Uppsala, Sweden, July.

Jia Xu, Evgeny Matusov, Richard Zens, and Hermann Ney. 2005. Integrated chinese word segmentation in statistical machine translation. In *International Workshop on Spoken Language Translation*, pages 141–147, Pittsburgh, PA, USA, October.