Smooth Bilingual N-gram Translation

Holger Schwenk
LIMSI-CNRS, BP 133
91403 Orsay cedex, FRANCE
schwenk@lismi.fr

Marta R. Costa-jussà and ***José A.R. Fonollosa***
UPC - TALP
Barcelona 08034, Spain
{mruiz,adrian}@gps.tsc.upc.edu

Abstract

We address the problem of smoothing translation probabilities in a bilingual N-gram-based statistical machine translation system. It is proposed to project the bilingual tuples onto a continuous space and to estimate the translation probabilities in this representation. A neural network is used to perform the projection and the probability estimation.

Smoothing probabilities is most important for tasks with a limited amount of training material. We consider here the BTEC task of the 2006 IWSLT evaluation. Improvements in all official automatic measures are reported when translating from Italian to English. Using a continuous space model for the translation model and the target language model, an improvement of 1.5 BLEU on the test data is observed.

1 Introduction

The goal of statistical machine translation (SMT) is to produce a target sentence $\mathbf{e}$ from a source sentence $\mathbf{f}$. Among all possible target language sentences the one with the highest probability is chosen:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \Pr(\mathbf{e}|\mathbf{f}) = \arg\max_{\mathbf{e}} \Pr(\mathbf{f}|\mathbf{e}) \Pr(\mathbf{e})$$

where $\Pr(\mathbf{f}|\mathbf{e})$ is the translation model and $\Pr(\mathbf{e})$ is the target language model. This approach is usually referred to as the *noisy source-channel* approach in statistical machine translation (Brown et al., 1993).

During the last few years, the use of context in SMT systems has provided great improvements in translation. SMT has evolved from the original word-based approach to phrase-based translation systems (Och et al., 1999; Koehn et al., 2003). A phrase is defined as a group of source words $\tilde{\mathbf{f}}$ that should be translated together into a group of target words $\tilde{\mathbf{e}}$. The translation model in phrase-based systems includes the phrase translation probabilities in both directions, i.e. $P(\tilde{\mathbf{e}}|\tilde{\mathbf{f}})$ and $P(\tilde{\mathbf{f}}|\tilde{\mathbf{e}})$.

The use of a maximum entropy approach simplifies the introduction of several additional models explaining the translation process :

$$\begin{aligned} \mathbf{e}^* &= \arg\max p(\mathbf{e}|\mathbf{f}) \\ &= \arg\max_{e} \{exp(\sum_i \lambda_i h_i(\mathbf{e}, \mathbf{f}))\} \end{aligned} \qquad (1)$$

The feature functions h_i are the system models and the λ_i weights are typically optimized to maximize a scoring function on a development set (Och and Ney, 2002).

The phrase translation probabilities $P(\tilde{\mathbf{e}}|\tilde{\mathbf{f}})$ and $P(\tilde{\mathbf{f}}|\tilde{\mathbf{e}})$ are usually obtained using relative frequency estimates. Statistical learning theory, however, tells us that relative frequency estimates have several drawbacks, in particular high variance and low bias. Phrase tables may contain several millions of entries, most of which appear only once or twice, which means that we are confronted with a data sparseness problem. Surprisingly, there seems to be little work addressing the issue of smoothing of the phrase table probabilities.

On the other hand, smoothing of relative frequency estimates was extensively investigated in the

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 430–438, Prague, June 2007.

area of language modeling. A systematic comparison can be for instance found in (Chen and Goodman, 1999). Language models and phrase tables have in common that the probabilities of rare events may be overestimated. However, in language modeling probability mass must be redistributed in order to account for the unseen n-grams. Generalization to unseen events is less important in phrase-based SMT systems since the system searches only for the best segmentation and the best matching phrase pair among the existing ones.

We are only aware of one work that performs a systematic comparison of smoothing techniques in phrase-based machine translation systems (Foster et al., 2006). Two types of phrase-table smoothing were compared: black-box and glass-box methods. Black-methods do not look inside phrases but instead treat them as atomic objects. By these means, all the methods developed for language modeling can be used. Glass-box methods decompose $P(\tilde{\mathbf{e}}|\tilde{\mathbf{f}})$ into a set of lexical distributions $P(e|\tilde{\mathbf{f}})$. For instance, it was suggested to use IBM-1 probabilities (Och et al., 2004), or other lexical translation probabilities (Koehn et al., 2003; Zens and Ney, 2004). Some form of glass-box smoothing is now used in all state-of-the-art statistical machine translation systems.

Another approach related to phrase table smoothing is the so-called N-gram translation model (Mariño et al., 2006). In this model, bilingual tuples are used instead of the phrase pairs and n-gram probabilities are considered rather than relative frequencies. Therefore, smoothing is obtained using the standard techniques developed for language modeling. In addition, a context dependence of the phrases is introduced. On the other hand, some restrictions on the segmentation of the source sentence must be used. N-gram-based translation models were extensively compared to phrase-based systems on several tasks and typically achieve comparable performance.

In this paper we propose to investigate improved smoothing techniques in the framework of the N-gram translation model. Despite the undeniable success of n-graam back-off models, these techniques have several drawbacks from a theoretical point of view: the words are represented in a discrete space, the vocabulary. This prevents "true interpolation" of the probabilities of unseen n-grams since a change in this word space can result in an arbitrary change of the n-gram probability. An alternative approach is based on a *continuous representation* of the words (Bengio et al., 2003). The basic idea is to convert the word indices to a continuous representation and to use a probability estimator operating in this space. Since the resulting distributions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. Probability estimation and interpolation in a continuous space is mathematically well understood and numerous powerful algorithms are available that can perform meaningful interpolations even when only a limited amount of training material is available. This approach was successfully applied to language modeling in large vocabulary continuous speech recognition (Schwenk, 2007) and to language modeling in phrase-based SMT systems (Schwenk et al., 2006).

In this paper, we investigate whether this approach is useful to smooth the probabilities involved in the bilingual tuple translation model. Reliable estimation of unseen n-grams is very important in this translation model. Most of the trigram tuples encountered in the development or test data were never seen in the training data. N-gram hit rates are reported in the results section of this paper. We report experimental results for the BTEC corpus as used in the 2006 evaluations of the international workshop on spoken language translation IWSLT (Paul, 2006). This task provides a very limited amount of resources in comparison to other tasks like the translation of journal texts (NIST evaluations) or of parliament speeches (TC-STAR evaluations). Therefore, new techniques must be deployed to take the best advantage of the limited resources. Among the language pairs tested in this years evaluation, Italian to English gave the best BLEU results in this year evaluation. The better the translation quality is, the more it is challenging to outperform it without adding more data. We show that a new smoothing technique for the translation model achieves a significant improvement in the BLEU score for a state-of-the-art statistical translation system.

This paper is organized as follows. In the next section we first describe the baseline statistical machine translation systems. Section 3 presents the architecture and training algorithms of the continuous space translation model and section 4 summarizes the experimental evaluation. The paper concludes with a discussion of future research directions.

2 N-gram-based Translation Model

The N-gram-based translation model has been derived from the finite-state perspective; more specifically, from the work of Casacuberta (2001). However, different from it, where the translation model is implemented by using a finite-state transducer, the N-gram-based system implements a bilingual N-gram model. It actually constitutes a language model of bilingual units, referred to as tuples, which approximates the joint probability between source and target languages by using N-grams, such as described by the following equation:

$$p(\mathbf{e}, \mathbf{f}) \approx \prod_{k=1}^{K} p((e,f)_k | (e,f)_{k-1}, \ldots, (e,f)_{k-4}) \quad (2)$$

where e refers to target, f to source and $(e,f)_k$ to the k^{th} tuple of a given bilingual sentence pair.

Bilingual units (tuples) are extracted from any word-to-word alignment according to the following constraints:

- a monotonic segmentation of each bilingual sentence pairs is produced,
- no word inside the tuple is aligned to words outside the tuple, and
- no smaller tuples can be extracted without violating the previous constraints.

As a consequence of these constraints, only one segmentation is possible for a given sentence pair.

Two important issues regarding this translation model must be considered. First, it often occurs that a large number of single-word translation probabilities are left out of the model. This happens for all words that are always embedded in tuples containing two or more words, then no translation probability for an independent occurrence of these embedded words will exist. To overcome this problem, the tuple trigram model is enhanced by incorporating 1-gram translation probabilities for all the embedded words detected during the tuple extraction step. These 1-gram translation probabilities are computed from the intersection of both the source-to-target and the target-to-source alignments.

The second issue has to do with the fact that some words linked to NULL end up producing tuples with NULL source sides. Since no NULL is actually expected to occur in translation inputs, this type of tuple is not allowed. Any target word that is linked to NULL is attached either to the word that precedes or the word that follows it. To determine this, an approach based on the IBM1 probabilities was used, as described in (Mariño et al., 2006).

2.1 Additional features

The following feature functions were used in the N-gram-based translation system:

- A **target language model**. In the baseline system, this feature consists of a 4-gram back-off model of words, which is trained from the target side of the bilingual corpus.

- A **source-to-target lexicon model and a target-to-source lexicon model**. These feature, which are based on the lexical parameters of the IBM Model 1, provide a complementary probability for each tuple in the translation table.

- A **word bonus function**. This feature introduces a bonus based on the number of target words contained in the partial-translation hypothesis. It is used to compensate for the system's preference for short output sentences.

All these models are combined in the decoder. Additionally, the decoder allows for a non-monotonic search with the following distorsion model.

- A word distance-based **distorsion model**.

  $$P(t_1^K) = exp(-\sum_{k=1}^{K} d_k)$$

  where d_k is the distance between the first word of the k^{th} tuple (unit), and the last word+1 of the $(k-1)^{th}$ tuple.

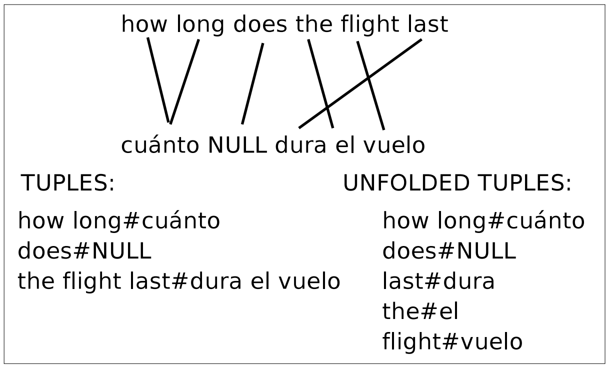

Figure 1: *Comparing regular and unfolded tuples.*

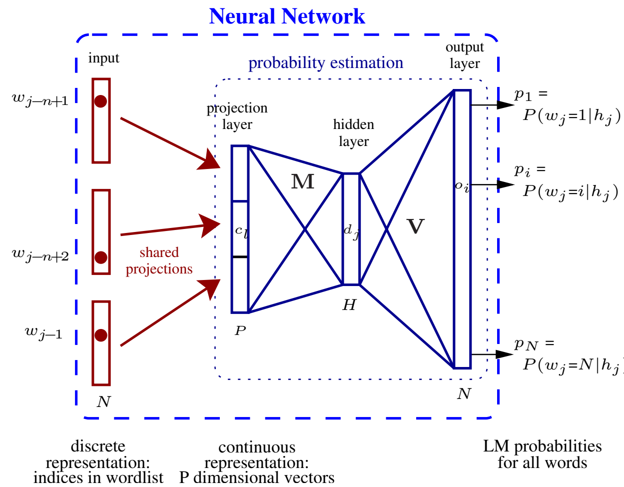

Figure 2: Architecture of the continuous space LM. h_j denotes the context $w_{j-n+1}, \ldots, w_{j-1}$. P is the size of one projection and H,N is the size of the hidden and output layer respectively. When short-lists are used the size of the output layer is much smaller than the size of the vocabulary.

Distance are measured in words referring to the units source side.

To reduce the computational cost we place limits on the search using two parameters: the distortion limit (the maximum distance measured in words that a tuple is allowed to be reordered, m) and the reordering limit (the maximum number of reordering jumps in a sentence, j). Tuples need to be extracted by an unfolding technique (Mariño et al., 2006). This means that the tuples are broken into smaller tuples, and these are sequenced in the order of the target words. In order not to lose the information on the correct order, the decoder performs a non-monotonic search. Figure 1 shows an example of tuple unfolding compared to the monotonic extraction. The unfolding technique produces a different bilingual n-gram language model with reordered source words.

In order to combine the models in the decoder suitably, an optimization tool based on the Simplex algorithm is used to compute log-linear weights for each model.

3 Continuous Space N-gram Models

The architecture of the neural network n-gram model is shown in Figure 2. A standard fully-connected multi-layer perceptron is used. The inputs to the neural network are the indices of the $n-1$ previous units (words or tuples) in the vocabulary $h_j = w_{j-n+1}, \ldots, w_{j-2}, w_{j-1}$ and the outputs are the posterior probabilities of *all* units of the vocabulary:

$$P(w_j = i | h_j) \qquad \forall i \in [1, N] \tag{3}$$

where N is the size of the vocabulary. The input uses the so-called 1-of-n coding, i.e., the ith unit of the vocabulary is coded by setting the ith element of the vector to 1 and all the other elements to 0. The ith line of the $N \times P$ dimensional projection matrix corresponds to the continuous representation of the ith unit. Let us denote c_l these projections, d_j the hidden layer activities, o_i the outputs, p_i their softmax normalization, and m_{jl}, b_j, v_{ij} and k_i the hidden and output layer weights and the corresponding biases. Using these notations, the neural network performs the following operations:

$$d_j = \tanh\left(\sum_l m_{jl}\, c_l + b_j\right) \tag{4}$$

$$o_i = \sum_j v_{ij}\, d_j + k_i \tag{5}$$

$$p_i = e^{o_i} \;/\; \sum_{r=1}^{N} e^{o_r} \tag{6}$$

The value of the output neuron p_i corresponds directly to the probability $P(w_j = i | h_j)$.

Training is performed with the standard back-propagation algorithm minimizing the following error function:

$$E = \sum_{i=1}^{N} t_i \, log \, p_i + \beta \left(\sum_{jl} m_{jl}^2 + \sum_{ij} v_{ij}^2 \right) \quad (7)$$

where t_i denotes the desired output, i.e., the probability should be 1.0 for the next unit in the training sentence and 0.0 for all the other ones. The first part of this equation is the cross-entropy between the output and the target probability distributions, and the second part is a regularization term that aims to prevent the neural network from over-fitting the training data (weight decay). The parameter β has to be determined experimentally. Training is done using a re-sampling algorithm as described in (Schwenk, 2007).

It can be shown that the outputs of a neural network trained in this manner converge to the posterior probabilities. Therefore, the neural network directly minimizes the perplexity on the training data. Note also that the gradient is back-propagated through the projection-layer, which means that the neural network learns the projection of the units onto the continuous space that is best for the probability estimation task.

In general, the complexity to calculate one probability with this basic version of the neural network n-gram model is dominated by the dimension of the output layer since the size of the vocabulary (10k to 64k) is usually much larger than the dimension of the hidden layer (200 to 500). Therefore, in previous applications of the continuous space n-gram model, the output was limited to the s most frequent units, s ranging between 2k and 12k (Schwenk, 2007). This is called a short-list.

	Sents	Words
Train (bitexts)	20k	155.4/166.3k
Dev	489	5.2k
Eval	500	6k

Table 1: Available data in the *supplied resources* of the 2006 IWSLT evaluation.

4 Experimental Evaluation

In this work we report results on the *Basic Traveling Expression Corpus* (BTEC) as used in the 2006 evaluations of the international workshop on spoken language translation (IWSLT). This corpus consists of typical sentences from phrase books for tourists in several languages (Takezawa et al., 2002). We report results on the supplied development corpus of 489 sentences and the official test set of the IWSLT'06 evaluation. The main measure is the BLEU score, using seven reference translations. The scoring is case insensitive and punctuations are ignored. Details on the available data are summarized in Table 1. We concentrated first on the translation from Italian to English. All participants in the IWSLT evaluation achieved much better performances for this language pair than for the other considered translation directions. This makes it more difficult to achieve additional improvements.

A non-monotonic search was performed following a local reordering named in Section 2, setting $m = 5$ and $j = 3$. Also we used histogram pruning in the decoder, i.e. the maximum number of hypotheses in a stack is limited to 50.

4.1 Language-dependent preprocessing

Italian contracted prepositions have been separated into preposition + article, such as 'alla'→'a la', 'degli'→'di gli' or 'dallo'→'da lo', among others.

4.2 Model training

The training and development data for the bilingual back-off and neural network translation model were created as follows. Given the alignment of the training parallel corpus, we perform a unique segmentation of each parallel sentence following the criterion of unfolded segmentation seen in Section 2. This segmentation is used in a sequence as training text for building the language model. As an example, given the alignment and the unfold extraction of Figure 1, we obtain the following training sentence:

<s> how_long#cuánto does#NULL last#dura the#el flight#vuelo </s>

The reference bilingual trigram back-off translation model was trained on these bilingual tuples using the SRI LM toolkit (Stolcke, 2002). Different smoothing techniques were tried, and best results were obtained using Good-Turing discounting.

The neural network approach was trained on exactly the same data. A context of two tuples was used (trigram model). The training corpus contains about 21,500 different bilingual tuples. We decided to limit the output of the neural network to the 8k most frequent tuples (short-list). This covers about 90% of the requested tuple n-grams in the training data.

Similar to previous applications, the neural network is not used alone but interpolation is performed to combine several n-gram models. First of all, the neural network and the reference back-off model are interpolated together - this always improved performance since both seem to be complementary. Second, four neural networks with different sizes of the continuous representation were trained and interpolated together. This usually achieves better generalization behavior than training one larger neural network. The interpolation coefficients were calculated by optimizing perplexity on the development data, using an EM procedure. The obtained values are 0.33 for the back-off translation model and about 0.16 for each neural network model respectively. This interpolation is used in all our experiments. For the sake of simplicity we will still call this the continuous space translation model.

Each network was trained independently using early stopping on the development data. Convergence was achieved after about 10 iterations through the training data (less than 20 minutes of processing on a standard Linux machine). The other parameters are as follows:

- Context of two tuples (trigram)
- The dimension of the continuous representation of the tuples were c =120,140,150 and 200,
- The dimension of the hidden layer was set to $P = 200$,
- The initial learning rate was 0.005 with an exponential decay,
- The weight decay coefficient was set to $\beta = 0.00005$.

N-gram models are usually evaluated using perplexity on some development data. In our case, i.e. using bilingual tuples as basic units ("words"), it is less obvious if perplexity is a useful measure. Nevertheless, we provide these numbers for completeness. The perplexity on the development data of the trigram back-off translation model is 227.0. This could be reduced to 170.4 using the neural network. It is also very informative to analyze the n-gram hit-rates of the back-off model on the development data: 10% of the probability requests are actually a true trigram, 40% a bigram and about 49% are finally estimated using unigram probabilities. This means that only a limited amount of phrase context is used in the standard N-gram-based translation model. This makes this an ideal candidate to apply the continuous space model since probabilities are interpolated for all possible contexts and never backed-up to shorter contexts.

4.3 Results and analysis

The incorporation of the neural translation model is done using n-best list. Each hypothesis is composed of a sequence of bilingual tuples and the corresponding scores of all the feature functions. Figure 3 shows an example of such an n-best list. The neural trigram translation model is used to replace the scores of the trigram back-off translation model. This is followed by a re-optimization of the coefficients of all feature functions, i.e. maximization of the BLEU score on the development data using the numerical optimization tool CONDOR (Berghen and Bersini, 2005). An alternative would be to add a feature function and to combine both translation models under the log-linear model framework, using maximum BLEU training.

Another open question is whether it might by better to already use the continuous space translation model during decoding. The continuous space model has a much higher complexity than a back-off n-gram. However, this can be heavily optimized when rescoring n-best lists, i.e. by grouping together all calls in the whole n-best list with the same context, resulting in only one forward pass through the neural network. This is more difficult to perform when the continuous space translation model is used during decoding. Therefore, this was not investigated in this work.

spiacente#sorry tutto_occupato#it_'s_full
spiacente#i_'m_sorry tutto_occupato#it_'s_full
spiacente#i_'m_afraid tutto_occupato#it_'s_full
spiacente#sorry tutto#all occupato#busy
spiacente#sorry tutto#all occupato#taken

Figure 3: Example of sentences in the n-best list of bilingual tuples. The special character '#' is used to separate the source and target sentence words. Several words in one tuple a grouped together using '_.'

In all our experiments 1000-best lists were used. In order to evaluate the quality of these n-best lists, an oracle trigram back-off translation model was build on the development data. Rescoring the n-best lists with this translation model resulted in an increase of the BLEU score of about 10 points (see Table 2). While there is an decrease of about 6% for the position dependent word error rate (mWER), a smaller change in the position independent word error rate was observed (mPER). This suggests that most of the alternative translation hypothesis result in word reorderings and not in many alternative word choices. This is one of the major drawbacks of phrase- and N-gram-based translation systems: only translations observed in the training data can be used. There is no generalization to new phrase pairs.

	Back-off	Oracle	**Neural**
BLEU	42.34	52.45	**43.87**
mWER	41.6%	35.6%	**40.3%**
mPER	31.5%	28.2%	**30.7%**

Table 2: Comparison of different N-gram-translation models on the development data.

When the 1000-best lists are rescored with the neural network translation model the BLEU score increases by 1.5 points (42.34 to 43.87). Similar improvements were observed in the word error rates (see Table 2). For comparison, a 4-gram back-off translation model was also built, but no change of the BLEU score was observed. This suggests that careful smoothing is more important than increasing the context when estimating the translation probabilities in an N-gram-based statistical machine translation system.

In previous work, we have investigated the use of the neural network approach to modeling the target language for the IWSLT task (Schwenk et al., 2006). We also applied this technique to this improved N-gram-based translation system. In our implementation, the neural network target 4-gram language model gives an improvement of 1.3 points BLEU on the development data (42.34 to 43.66), in comparison to 1.5 points for the neural translation model (see Table 3).

	Back-off TM+LM	neural TM	neural LM	**neural TM+LM**
BLEU	42.34	43.87	43.66	**44.83**

Table 3: Combination of a neural translation model (TM) and a neural language model (LM). BLEU scores on the development data.

The neural translation and target language model were also applied to the test data, using of course the same feature function coefficients as for the development data. The results are given in Table 4 for all the official measures of the IWSLT evaluation. The new smoothing method of the translation probabilities achieves improvement in all measures. It gives also an additional gain (again in all measures) when used together with a neural target language model. Surprisingly, neural TM and neural LM improvements almost add up: when both techniques are used together, the BLEU scores increases by 1.5 points (36.97 → 38.50). Remember that the reference N-gram-based translation system already uses a local reordering approach.

	Back-off TM+LM	neural TM	neural LM	**neural TM+LM**
BLEU	36.97	37.21	38.04	**38.50**
mWER	48.10	**47.42**	47.83	47.61
mPER	38.21	38.07	37.26	**37.12**
NIST	8.3	8.3	8.6	**8.7**
Meteor	63.16	63.40	64.70	**65.20**

Table 4: Test set scores for the combination of a neural translation model (TM) and a neural language model (LM).

5 Discussion

Phrase-based approaches are the de-facto standard in statistical machine translation. The phrases are extracted automatically from the word alignments of parallel texts, and the different possible translations of a phrase are weighted using relative frequency. This can be problematic when the data is sparse. However, there seems to be little work on possible improvements of the relative frequency estimates by some smoothing techniques. It is today common practice to use additional feature functions like IBM-1 scores to obtain some kind of smoothing (Och et al., 2004; Koehn et al., 2003; Zens and Ney, 2004), but better estimation of the phrase probabilities is usually not addressed.

An alternative way to represent phrases is to define bilingual tuples. Smoothing, and context dependency, is obtained by using an n-gram model on these tuples. In this work, we have extended this approach by using a new smoothing technique that operates on a continuous representation of the tuples. Our method is distinguished by two characteristics: better estimation of the numerous unseen n-grams, and a **discriminative** estimation of the tuple probabilities. Results are provided on the BTEC task of the 2006 IWSLT evaluation for the translation direction Italian to English. This task provides very limited amount of resources in comparison to other tasks. Therefore, new techniques must be deployed to take the best advantage of the limited resources. We have chosen the Italian to English task because it is challenging to enhance a good quality translation task (over 40 BLEU percentage). Using the continuous space model for the **translation** and **target** language model, an improvement of 2.5 BLEU on the development data and 1.5 BLEU on the test data was observed.

Despite these encouraging results, we believe that additional research on improved estimation of probabilities in N-gram- or phrase-based statistical machine translation systems is needed. In particular, the problem of **generalization** to new translations seems to be promising to us. This could be addressed by the so-called factored phrase-based model as implemented in the Moses decoder (Koehn et al., 2007). In this approach words are decomposed into several factors. These factors are translated and a target phrase is generated. This model could be complemented by a factored continuous tuple N-gram. Factored word language models were already successfully used in speech recognition (Bilmes and Kirchhoff, 2003; Alexandrescu and Kirchhoff, 2006) and an extension to machine translation seems to be promising.

The described smoothing method was explicitly developed to tackle the data sparseness problem in tasks like the BTEC corpus. It is well known from language modeling that careful smoothing is less important when large amounts of data are available. We plan to investigate whether this also holds for smoothing of the probabilities in phrase- or tuple-based statistical machine translation systems.

6 Acknowledgments

This work has been partially funded by the European Union under the integrated project TC-STAR (IST-2002-FP6-506738), by the French Government under the project INSTAR (ANR JCJC06_143038) and the the Spanish government under a FPU grant and the project AVIVAVOZ (TEC2006-13964-C03).

References

A. Alexandrescu and K. Kirchhoff. 2006. Factored neural language models. In *HLT-NAACL*.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(2):1137–1155.

F. Vanden Berghen and H. Bersini. 2005. CONDOR, a new parallel, constrained extension of powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181:157–175.

J. A. Bilmes and K. Kirchhoff. 2003. Factored language models and generalized backoff. In *HLT-NAACL*.

P. Brown, S. Della Pietra, V. J. Della Pietra, and R: Mercer. 1993. The mathematics of statistical machine translation. *Computational Linguistics*, 19(2):263–311.

F. Casacuberta, D. Llorens, C. Martínez, S. Molau, F. Nevado, H. Ney, M. Pastor, D. Picó, A. Sanchis, E. Vidal, and J.M. Vilar. 2001. Speech-to-speech translation based on finite-state transducers. *International Conference on Acoustic, Speech and Signal Processing*, 1.

S. F. Chen and J. T. Goodman. 1999. An empirical study of smoothing techniques for language modeling. *CSL*, 13(4):359–394.

G. Foster, R. Kuhn, and H. Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *EMNLP06*, pages 53–61.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrased-based machine translation. In *Human Language Technology Conference (HLT-NAACL)*, pages 127–133.

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 ACL, demonstration session*.

J.B. Mariño, R.E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J.A.R. Fonollosa, and M. R. Costa-jussà. 2006. Bilingual n-gram statistical machine translation. *Computational Linguistics*, 32(4):527–549, December.

F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *ACL*, pages 295–302.

F. J. Och, C. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Copora*, pages 20–28.

F.-J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. 2004. A smorgasbord of features for statistical machine translation. In *HLT-NAACL*, pages 161–168.

M. Paul. 2006. Overview of the IWSLT 2006 campaign. In *IWSLT*, pages 1–15.

H. Schwenk, M. R. Costa-jussà, and J. A. R. Fonollosa. 2006. Continuous space language models for the iwslt 2006 task. *IWSLT*, pages 166–173.

H. Schwenk. 2007. Continuous space language models. *Computer Speech and Language*, 21:492–518.

A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *ICSLP*, pages II: 901–904.

T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto. 2002. Toward a borad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *LREC*, pages 147–152.

R. Zens and H. Ney. 2004. Improvements in phrase-based statistical machine translation. In *HLT/NACL*, pages 257–264.