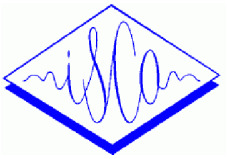
ISCA Archive
http://www.isca-speech.org/archive



International Workshop on Spoken
Language Translation (IWSLT) 2006
Keihanna Science City, Kyoto, Japan
November 27-28, 2006

Continuous Space Language Models for the IWSLT 2006 Task

Holger Schwenk

LIMSI-CNRS, BP 133
91403 Orsay cedex, FRANCE
schwenk@limsi.fr

Marta R. Costa-jussà and *José A. R. Fonollosa*

Universitat Politècnica de Catalunya (UPC)
Barcelona 08034, Spain
mruiz,adrian@gps.tsc.upc.edu

Abstract

The language model of the target language plays an important role in statistical machine translation systems. In this work, we propose to use a new statistical language model that is based on a continuous representation of the words in the vocabulary. A neural network is used to perform the projection and the probability estimation. This kind of approach is in particular promising for tasks where a very limited amount of resources are available, like the BTEC corpus of tourism related questions.

This language model is used in two state-of-the-art statistical machine translation systems that were developed by UPC for the 2006 IWSLT evaluation campaign: a phrase- and an n-gram-based approach. An experimental evaluation for four different language pairs is provided (translation of Mandarin, Japanese, Arabic and Italian to English). The proposed method achieved improvements in the BLEU score of up to 3 points on the development data and of almost 2 points on the official test data.

1. Introduction

Speech translation of dedicated tasks like the BTEC corpus of tourism related questions is challenging. Statistical methods have obtained very good performances at the last evaluation campaigns organized by the International Workshop on Spoken Language Translation (IWSLT). However, these techniques rely on representative corpora to train the underlying models: sentence aligned bilingual texts to train the translation models and text in the target language to develop a statistical language model (LM). In the 2006 IWSLT evaluation 40k sentences of bitexts were provided in the *Supplied Resources* in the "open data track". The English side of the bitexts is used to train the target language model (326k words). This is 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. For instance, it was proposed to use a translation lexicon that was extracted by applying the Competitive Linking Algorithm on the bilingual training data [1]. By that way, important improvements in the BLEU score were obtained. With respect to language modeling, most of the statistical machine translation systems (SMT) that participated in the 2005 IWSLT evaluation used 4-gram back-off LM. Some sites reported improvements using 5-gram word or class-based LMs [2, 3], or even 9-gram prefix and suffix LMs [4]. Language model adaptation was investigated in [5]. Other interesting approaches include factored [6] or syntax-based language models [7], but to the best of our knowledge, there were not yet applied to the BTEC corpus.

In this paper, we investigate if the so-called continuous space language model can be used in a state-of-the-art statistical machine translation system for the IWSLT task. The basic idea of the continuous space LM, also called neural network LM, is to project the word indices onto a continuous space and to use a probability estimator operating on this space [8]. Since the resulting probability functions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. A neural network can be used to simultaneously learn the projection of the words onto the continuous space and to estimate the n-gram probabilities. This is still a n-gram approach, but the LM posterior probabilities are "interpolated" for any possible context of length n-1 instead of backing-off to shorter contexts.

This approach was successfully used in large vocabulary continuous speech recognition [9], and initial experiments have shown that it can be used to improve a word-based statistical machine translation system [10]. Here, the continuous space LM is applied the first time to a state-of-the-art phrase-based SMT system. Translation of four different languages is considered: Mandarin, Japanese, Arabic and Italian to English. These languages exhibit very different characteristics, e.g. with respect to word order, which may affect the role of the target LM, although a reordering model is used in the SMT systems. We also investigate the use of the continuous space LM in a SMT system based on bilingual n-grams.

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 LM and section 4 summarizes the experimental evaluation. The paper concludes with a discussion of future research directions.

2. Baseline systems

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. In parallel to the phrase-based approach, the use of bilingual n-grams gives comparable results, as shown by Crego et al. [11]. Two basic issues differentiate the n-gram-based system from the phrase-based: training data are monotonically segmented into bilingual units; and the model considers n-gram probabilities rather than relative frequencies. This translation approach is described in detail by Mariño et al. [12].

Both systems follow a maximum entropy approach, in which a log-linear combination of multiple models is implemented, as an alternative to the source-channel approach: This simplifies the introduction of several additional models explaining the translation process, as the search becomes:

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

where f and e are sentences in the source and target language respectively. 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. Both the n-gram-based and the phrase-based system use a language model on the target language as feature function, i.e. $P(e)$, but they differ in the translation model. In both cases, it is based on bilingual units. A bilingual unit consists of two monolingual fragments, where each one is supposed to be the translation of its counterpart. During training, each system learns its dictionary of bilingual fragments.

Both SMT approaches were evaluated in IWSLT'06 evaluation and they are described in detail in [13, 14]. Therefore, we only give a short summary in the following two sections.

2.1. N-gram-based Translation Model

The translation model can be thought of a language model of bilingual units (here called tuples). These tuples define a monotonic segmentation of the training sentence pairs (f, e), into K units $(t_1, ..., t_K)$.

The translation model is implemented using an n-gram language model, (for $N = 4$):

$$p(e, f) = Pr(t_1^K) = \prod_{k=1}^{K} p(t_k \mid t_{k-2}, t_{k-1}) \quad (2)$$

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 4-gram 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, we use the POS entropy approach, see de Gispert et al. [15].

2.2. Phrase-based Translation Model

Given a sentence pair and a corresponding word alignment, a phrase (or bilingual phrase) is any pair of m source words and n target words that satisfies two basic constraints:

1. Words are consecutive along both sides of the bilingual phrase,
2. No word on either side of the phrase is aligned to a word out of the phrase.

Given the collected phrase pairs, we estimate the phrase translation probability distribution by relative frequency.

$$P(f|e) = \frac{N(f,e)}{N(e)} \qquad P(e|f) = \frac{N(e,f)}{N(f)} \quad (3)$$

where N(f,e) means the number of times the phrase *f* is translated by *e*; N(e), the number of times the phrase *e* appears; and, N(f), the number of times the phrase *f*, appears. Notice that the phrase-based system has two feature functions ($P(f|e)$ and $P(e|f)$) which are considered translation models.

2.3. Additional features

Both systems share the additional features which follows.

- A **target language model**. In the baseline system, this feature consists of a 4-gram model of words, which is trained from the target side of the bilingual corpus.
- A **source-to-target lexicon model**. This feature, which is based on the lexical parameters of the IBM Model 1, provides a complementary probability for each tuple in the translation table. These lexicon parameters are obtained from the source-to-target alignments.
- A **target-to-source lexicon model**. Similarly to the previous feature, this feature is based on the lexical parameters of the IBM Model 1 but, in this case, these parameters are obtained from target-to-source alignments.
- 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.
- A **phrase bonus function**. This feature is used only in the phrase-based system and it introduces a bonus based on the number of target phrases contained in the partial-translation hypothesis.

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

- A word distance-based **distortion 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} unit, and the last word +1 of the $(k-1)^{th}$ unit. Distance is 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). This feature is independent of the reordering approach presented in this paper, so they can be used simultaneously.

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 Language Models

The architecture of the neural network LM is shown in Figure 1. A standard fully-connected multi-layer perceptron is used. The inputs to the neural network are the indices of the $n-1$ previous words 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* words of the vocabulary:

$$P(w_j = i|h_j) \qquad \forall i \in [1, N] \tag{4}$$

where N is the size of the vocabulary. The input uses the so-called 1-of-n coding, i.e., the ith word 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 word. 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:

$$d_j = \tanh\left(\sum_l m_{jl}\, c_l + b_j\right) \tag{5}$$

$$o_i = \sum_j v_{ij}\, d_j + k_i \tag{6}$$

$$p_i = e^{o_i} \,/\, \sum_{r=1}^{N} e^{o_r} \tag{7}$$

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

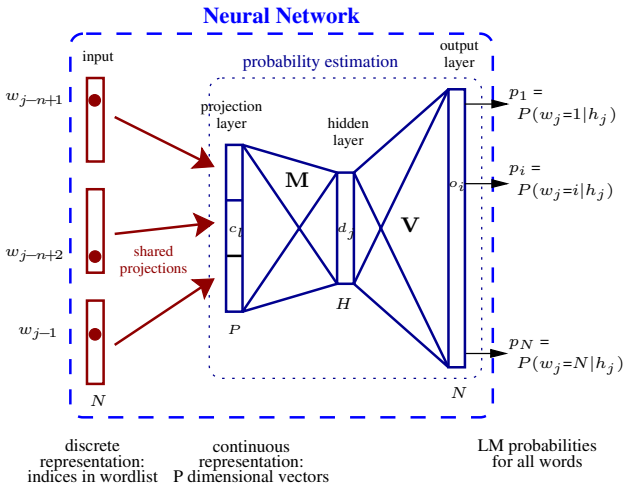


Figure 1: Architecture of the continuous space LM. h_j denotes the context $w_{j-n+1}, \ldots, w_{j-1}$. P is the size of one projection and H,N is the size of the hidden and output layer respectively. When short-lists are used the size of the output layer is much smaller then the size of the vocabulary.

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

$$E = \sum_{i=1}^{N} t_i \log p_i + \beta \left(\sum_{jl} m_{jl}^2 + \sum_{ij} v_{ij}^2 \right) \quad (8)$$

where t_i denotes the desired output, i.e., the probability should be 1.0 for the next word 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 overfitting the training data (weight decay). The parameter β has to be determined experimentally. Training is done using a resampling algorithm [9].

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 words 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 LM is dominated by the dimension of the output layer since the size of the vocabulary (up to 200k) is usually much larger than the dimension of the hidden layer (300. . .600). Therefore, in previous applications of the continuous space LM, the output was limited to the s most frequent words, s ranging between 2k and 12k [9]. This was not necessary for the BTEC task since the whole training corpus contains less than 10k different words. Thus, this is the first time that the continuous space LM is used to predict the LM probabilities of all words in the vocabulary.

The incorporation of the continuous space LM into the phrase- or n-gram-based translation system is done by using n-best lists. In all our experiments, the language model probabilities provided by the continuous space LM are used to replace those of the default 4-gram LM (a particular feature function). The coefficients of all the feature functions were then optimized on the development data.

We did not try to use the continuous space LM directly during decoding since this would result in increased decoding times. Calculating a LM probability with a back-off model corresponds basically to a table look-up using hashing techniques, while a forward pass through the neural network is necessary for the continuous space LM. Very efficient optimizations are possible, in particular when n-grams with the same context can be grouped together, but a reorganization of the decoder may be necessary. More details on optimizing the neural network LM can be found in [9].

4. Experimental Evaluation

In this work we report results on the *Basic Traveling Expression Corpus* (BTEC). This corpus consists of typical sentences from phrase books for tourists in several languages [16]. Translation to English from four languages is considered: Mandarin, Japanese, Arabic and Italian. The reference phrase- and n-gram-based SMT systems participated in the *open data track* of the 2006 IWSLT evaluation [13, 14], i.e. only the supplied subset of the full BTEC corpus was used to train all the statistical models. Details on the data preprocessed as in [13, 14] are summarized in Table 1. We report results on the supplied development corpus of 489 sentences (less than 6k words) using the BLEU score with seven references translations. The scoring is case insensitive and punctuations are ignored.

Transl. dir.:		Ma/En	Ja/En	Ar/En	It/En
Train	sent.	40k	40k	20k	20k
(bitexts)	words	314.4k	390.2k	183.3k	155.4k
	English	326k	324.8k	166.3k	166.3k
Dev	sent.	489			
	words	5.5k	6.8k	5.9k	5.2k
Eval	sent.	500			
	words	5.9k	7.4k	6.6k	6k

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

For all tasks, a non-monotonic search was performed taking the limits of: $m = 5$ and $j = 3$. Except for the Italian to English in the phrase-based system, we used a monotonic search. Also we used the histogram pruning in the decoder (i.e. limit the maximum number of hypotheses in a stack), $b = 50$ both in the development set and in the test set, for all tasks. The decoder first generates a lattice from which we then extract 1000-best lists. Rescoring these 1000-best list with the continuous space LM takes about 20min a Linux server with two 2.8 GHz Xeon processors.

The reference 4-gram back-off LM was trained on the English part of the bitexts (40k sentences, 326k words) using the SRI LM toolkit [17]. The neural network LM was trained on exactly the same data. Like in previous applications, the neural network is not used alone but interpolation is performed to combine several language models. First of all, the neural network and the reference back-off LM are interpolated together - this always improved performance since both seem to be complementary. Second, seven neural networks with different sizes of the continuous representation were trained and interpolated together.[1] 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

[1] We did not try to find the minimal number of neural networks to be combined. It is quite likely that the interpolation of less networks would result in the same BLEU scores.

	Phrase-based system			N-gram-based system		
	Oracle	Ref.	CSLM	Oracle	Ref.	CSLM
Mandarin	33.1	20.68	21.97	32.0	20.84	21.83
Japanese	26.9	17.29	18.27	28.6	18.34	19.77
Arabic	40.1	27.92	30.28	41.6	29.09	30.89
Italian	56.2	41.66	44.03	58.1	41.65	44.67

Table 2: BLEU scores on the development data. *Oracle* uses a back-off LM trained on the references, *Ref.* is the default system as submitted in the official IWSLT evaluation and *CSLM* uses the continuous space LM.

procedure. The obtained values are 0.33 for the back-off LM and about 0.1 for each neural network LM respectively. This interpolation is used in all our experiments. For the sake of simplicity we will still call this the continuous space LM.

An alternative would be to add a feature function and to combine all LMs under the log-linear model framework, using maximum BLEU training. This raises the interesting question whether the two criteria (minimal perplexity versus maximal BLEU score) lead to equivalent performance when multiple language models are used in a SMT system. In previous experiments with a word-based statistical machine translation system, both approaches yielded similar performance [10].

Each network was trained independently using early stopping on development data. Convergence was achieved after about 10 iterations through the training data (less than 1 hour of processing on a standard Linux machine). The other parameters are as follows:

- Context of three words (4-gram),
- The dimension of the continuous representation of the words were $c =$100,120,140,150,160,180 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 $\beta = 0.00005$.

Perplexity on development data is a popular and easy to calculate measure to evaluate the quality of a language model. However, it is not clear if perplexity is a good criterion to predict the improvements when the language model will be used in a SMT system. All seven reference translations were concatenated to one development corpus of 40k words. The perplexity of the 4-gram reference back-off LM on this data is 124.0. This could be reduced to 96.6 using the continuous space LM.

4.1. Development results and analysis

Table 2 gives the BLEU scores when the back-off and continuous space LM is used with a phrase- and n-gram-based SMT system. Often it is informative to have an idea of the oracle BLEU score of the n-best lists. This was estimated by rescoring the n-best lists with a "cheating back-off LM" that was build on the concatenated seven reference translation.

The continuous space network LM always achieved important improvements of the BLEU score, ranging from 1 point absolute (phrase-based system for Japanese) up to 3 points absolute (n-gram-based system for Italian).

In the case of the n-gram-based system, the translation model includes an implicit target language model. However, the additional target language model provided by the neural network gives also a clear improvement in the translation quality. In average the gain is slightly higher than for the phrase-based system. The n-gram-based system has less bilingual units. However, it seems to offer slightly higher variability in the n-best lists because of the single segmentation of the training parallel corpus. In the n-best lists generated by the phrase-based system we usually find several equal sentences corresponding to different segmentations. On the contrary, the n-gram-based system rarely outputs equal target sentences with different segmentations. That is why the n-best lists of the n-gram-based system tend to offer a higher oracle than the n-best lists of the phrase-based system.

Finally, the performance of the continuous space LM differs sensibly for the different translation directions. It is not surprising that the best improvements were obtained for the translation from Italian to English, two languages that are quite similar with respect to word order. The gain brought by the neural network LM is 2.4 points for the phrase-based system and 3 points BLEU for the n-gram-based system. For the two Asian languages Mandarin and Japanese, whose sentence structure is more different from English word order, the improvement brought by the new LM is about 1 point BLEU. In fact, the hypotheses in the n-best lists differ mostly in the choice of words and phrases, but there is less variation in the word order. It is surprising to see the good performance of the neural network LM for the translation of the Arabic language: the gain is 2.36 for the phrase-based system and 1.8 for the n-gram-based system.

The (position independent) word error rates are given in Table 3. The most interesting case is the translation from Arabic to English: the word error rate decreases by 3.3 percent and the position independent word error rate by 2.4 percent. There are also important word error reductions for the translation direction Japanese to English when using the n-gram SMT system. We are currently investigating why the

	Phrase-based			N-gram-based		
	Oracle	Ref.	CSLM	Oracle	Ref.	CSLM
Mandarin/English:						
mWER	59.1	67.4	66.5	58.1	67.8	66.6
mPER	44.5	50.8	50.1	45.3	51.5	50.6
Japanese/English:						
mWER	70.8	74.6	77.0	63.5	73.0	71.3
mPER	48.5	52.2	54.6	46.1	53.4	51.8
Arabic/English:						
mWER	49.1	56.0	52.7	48.1	55.7	52.8
mPER	40.3	45.7	43.3	39.6	44.0	42.5
Italian/English:						
mWER	34.1	42.3	40.7	33.1	42.8	40.8
mPER	26.6	31.6	30.5	26.0	31.9	30.7

Table 3: Word error rates on the development data.

error rates increase for the phrase-based system. Note also, that the absolute values are much higher for this language. One can also notice that the position independent oracle word error rates are only about 5 percentage lower than the ones of the best system. This may indicate that the n-best list generation could be improved in order to include more alternative translations. As already stated above, the n-best lists produced by the n-gram-based SMT system seem to be better than those provided by the phrase-based system.

In addition to the automatic scores we give some example translations in Figure 2. It seems that the neural network LM manages to improve the fluency of the translation in some cases, for instance the phrase *"we arrive time is two thirty"* is replaced by *"we arrive at two thirty"*, *"two and the fifty minutes"* by *"two and fifty minutes"*, *"two o'clock and thirty"* by *"two thirty"*, *"you ask"* by *"you can ask"* and *"are very busy"* by *"I'm very busy"*. Although the meaning seems to be pretty much preserved in the translations from the Asian languages, the fluency is clearly less good. Looking at these examples, one has the impression that the fluency of the n-gram-based SMT systems is slightly better than the one of the phrase-based systems.

4.2. Test set results

The results on the official test data of the 2006 IWSLT evaluation are summarized in Table 4. The numbers in the columns *"reference"* corresponds to the official results of the phrase and n-gram-based SMT system that UPC has developed. The numbers in the columns *"CSLM"* were obtained by rescoring the n-best lists of these official systems with the continuous space language model described in this paper, using the coefficients of the feature functions that were tuned on the development data.

As usually observed in SMT, the improvements on the test data are smaller than those on the development data which was used to tune the parameters. As a rule of thumb,

	Phrase-based		N-gram-based	
	Ref.	CSLM	Ref.	CSLM
Mandarin/English:				
BLEU	19.74	21.01	20.34	21.16
NIST	6.24	6.55	6.22	6.40
mWER	67.95	68.16	68.30	67.63
mPER	52.46	51.87	52.81	52.31
Japanese/English:				
BLEU	15.11	15.73	16.14	16.35
NIST	5.83	5.99	5.86	5.87
mWER	77.51	78.15	75.45	75.59
mPER	55.14	54.96	55.52	55.29
Arabic/English:				
BLEU	23.72	24.86	23.83	23.70
NIST	6.72	6.69	6.80	6.70
mWER	63.04	60.89	62.81	61.97
mPER	49.43	48.61	49.41	48.85
Italian/English:				
BLEU	35.55	37.41	35.95	37.65
NIST	8.32	8.53	8.40	8.57
mWER	49.12	47.22	48.78	47.59
mPER	38.17	36.62	38.12	37.26

Table 4: Result summary on the test data.

the gain on the test data is often half as large as on the Dev-data. It seems that the phrase-based system generalizes slightly better than the n-gram-based approach: there is less difference between the improvements on the development and those on the test data. This seems not to be related to the use of the continuous space LM since such behavior was previously observed on other tasks. One possibility could be that the n-gram approach is more sensible to overfitting when tuning the feature function coefficients. However, the n-gram-based SMT systems still achieve better BLEU scores than the phrase-based systems in three out of four tasks.

Another surprising result is the bad performance of the continuous space language model for the translation of Arabic to English with the n-gram-based system: the BLEU and NIST scores decrease despite an improvement in the word error rates. In fact, the reference n-gram-based SMT system performs not very well on the test data in comparison to the phrase-based system. While its BLEU score was almost 1.2 better on the development data it achieves basically the same result on the test data (23.83 BLEU with respect to 23.72). This needs further investigation.

5. Discussion

This paper investigated the use of a neural network LM that performs probability estimation in a continuous space. Since the resulting probability functions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. This is particularly interesting for

tasks like the BTEC corpus where only limited amounts of appropriate LM training material are available. The continuous space LM is used to rescore the n-best lists of a phrase- and n-gram-based statistical machine translation system that was developed by UPC for the 2006 IWSLT evaluation.

The results show a significant improvement for four different languages pairs and for both systems. The improvements on the development data range from one point BLEU when translating from Japanese to English, up to three points BLEU for the pair Italian to English. The new approach also achieves good improvements on the test data, the BLEU score increases by up to 1.9 points. All these results were obtained on top of the evaluation systems. In the case of the n-gram-based system, the translation model includes an implicit target language model. However, the additional target language model provided by the neural network also provides a clear improvement in the translation quality.

In this work, we have only studied 4-gram language models, but it is easy to train a neural network LM with much longer contexts since the complexity of our approach increases only slightly with the size of the context. Another promising direction that we have not yet explored, is to use the neural network LM for the translation model of the n-gram-based system. This would result in a continuous space translation model.

6. Acknowledgments

This work has been partially funded by the European Union under the integrated project TC-STAR (IST-2002-FP6-506738, http://www.tc-star.org) and the Spanish government under a FPU grant.

7. References

[1] B. Chen, R. Cattoni, N. Bertoldi, M. Cettelo, and M. Federico, "The ITC-irst SMT system for IWSLT-2005," in *IWSLT*, 2005, pp. 98–104.

[2] M. Paul, T. Doi, Y. Hwang, K. Imamura, H. Okuma, and E. Sumita, "Nobody is perfect: ATR's hybrid approach to spoken language translation," in *IWSLT*, 2005, pp. 55–62.

[3] A. Menezes and C. Quirk, "Microsoft research treelet translation system: IWSLT evaluation," in *IWSLT*, 2005, pp. 105–108.

[4] H. Tsukada, T. Watanabe, J. Suzuki, H. Kazawa, and H. Isozaki, "The NTT statistical machine translation system for IWSLT2005," in *IWSLT*, 2005, pp. 128–133.

[5] S. Hewavitharana, B. Zhao, A. S. Hildebrand, M. Eck, C. Hori, S. Vogel, and A. Waibel, "The CMU statistical machine translation system for IWSLT2005," in *IWSLT*, 2005, pp. 63–70.

[6] K. Kirchhoff and M. Yang, "Improved language modeling for statistical machine translation," in *ACL'05 workshop on Building and Using Parallel Text*, 2005, pp. 125–128.

[7] E. Charniak, K. Knight, and K. Yamada, "Syntax-based language models for machine translation," in *MT Summit*, 2003.

[8] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. 2, pp. 1137–1155, 2003.

[9] H. Schwenk, "Continuous space language models," *accpeted for publication in Computer Speech and Language*, 2007.

[10] H. Schwenk, D. Déchelotte, and J.-L. Gauvain, "Continuous space language models for statistical machine translation," in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 2006, pp. 723–730.

[11] J. Crego, M. R. Costa-jussà, J. Mariño, and J. A. Fonollosa, "Ngram-based versus phrase-based statistical machine translation," in *IWSLT*, 2005, pp. 177–190.

[12] J. Mariño, R. Blanchs, J. Crego, A. de Gispert, P. Lambert, M. R. Costa-jussà, and J. Fonollosa, "Bilingual n-gram statistical machine translation," in *MT Summit*, 2005, pp. 275–82.

[13] M. Costa-jussà, J. Crego, A. de Gispert, P. Lambert, M. Khalilov, J. Mariño, J. Fonollosa, and R. Banchs, "TALP phrase-based statistical translation system and TALP systems combination for the IWSLT 2006," in *IWSLT*, November 2006.

[14] A. de Gispert, J. Crego, P. Lambert, M. Costa-jussà, M. Khalilov, R. Banchs, J. Mariño, , and J. Fonollosa, "N-gram-based SMT system enhanced with reordering patterns for the IWSLT 2006," in *IWSLT*, November 2006.

[15] A. de Gispert and J. Mariño, "Linguistic tuple segmentation in ngram-based statistical machine translation," in *Interspeech*, September 2006, pp. 1149–1152.

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

[17] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *ICSLP*, 2002, pp. II: 901–904.

Reference translations:

ref 1: *for your information we will arrive at two thirty and your departure time is at two fifty*
oh sorry i'm very busy now you can ask someone else

ref 2: *please refer to this information arrival at two thirty and departure at two fifty*
sorry i'm a bit tied up at this moment could you ask someone else

ref 3: *for your reference landing time is two thirty and take off time is two fifty*
i'm sorry i can't help since i'm busy now please try someone else

ref 4: *we will arrive at two thirty and set out at two fifty please consider this information*
sorry i'm a bit too busy right now it'll be nice if you'd ask someone else

ref 5: *you will need to consider that we will arrive at two thirty and your next flight is at two fifty*
i'm sorry but i'm busy at the moment please ask someone else

ref 6: *please bear in mind that we touch down at two thirty and your connecting flight is at two fifty*
i'm sorry but i'm very busy now you can try asking someone else

ref 7: *please be mindful that we arrive at two thirty and your next departure is at two fifty*
sorry but i'm very busy now you might want to ask someone else

Phrase-based system:

Zh/En baseline: *could you we arrive time is two thirty departure time is two five ten*
oh i'm sorry but i'm busy right now you can ask someone else

CSLM: *you can the time we arrive at two thirty departure time is two fifty*
oh i'm sorry but i'm busy right now you can ask someone else please

Ja/En baseline: *we arrive at two thirty take off time at two fifty in your it you please*
i'm sorry but my hand you had better ask someone else you can

CSLM: *we arrive at two thirty take off schedule at two fifty in you it you please*
i'm sorry my hand you had better ask someone else i can do

Ar/En baseline: *information your will we arrive at two thirty and an appointment is two and the fifty minutes*
excuse me i'm very busy 's ask someone else

CSLM: *information i'll arrive at two thirty and time is two and fifty minutes*
excuse me i'm very busy's ask someone else

It/En baseline: *for your information we'll be arriving at two o'clock and thirty and your departure time is at two o'clock and fifty*
oh sorry i'm very busy right now you can ask someone else

CSLM: *for your information we'll arrive at two thirty and your departure time is at two fifty*
oh sorry i'm very busy now you can ask someone else

N-gram-based system:

Zh/En baseline: *you can reference our arrival time is two thirty departure time is two fifty*
oh i'm sorry i'm in a hurry now you ask someone else

CSLM: *you can reference our arrival time is two thirty departure time is two fifty*
oh i'm sorry i'm busy right now you can ask someone else

Ja/En baseline: *we arrive at two thirty takeoff time is fifty two o'clock so you reference you please*
i'm sorry but i can't get them eyes someone else to ask you can

CSLM: *we arrive at two thirty take off time is two o'clock in fifty so you your reference please*
i'm sorry but i can't get them eyes someone else ask you can

Ar/En baseline: *i'll information you arrive at two thirty time and is two and fifty minutes*
excuse me i'm very busy it's ask someone else

CSLM: *i'll information you arrive at two thirty and time is two and fifty minutes*
excuse me i'm very busy it's ask someone else

It/En baseline: *it's for your information we'll be arriving at two thirty and your departure time is at two fifty*
oh excuse me are very busy right now you can ask someone else

CSLM: *it's for your information we'll arrive at two thirty and your departure time is at two fifty*
oh sorry i'm very busy right now you can ask someone else

Figure 2: Example translations using the baseline back-off and the continuous space language model (CSLM).