CEU-UPV English–Spanish system for WMT11

Francisco Zamora-Martínez
D. Física, Matemática, y Computación
Universidad CEU-Cardenal Herrera
Alfara del Patriarca (Valencia), Spain
fzamora@dsic.upv.es

M.J. Castro-Bleda
D. Sistemas Informáticos y Computación
Universitat Politècnica de València
Valencia, Spain
mcastro@dsic.upv.es

Abstract

This paper describes the system presented for the English-Spanish translation task by the collaboration between CEU-UCH and UPV for 2011 WMT. A comparison of independent phrase-based translation models interpolation for each available training corpora were tested, giving an improvement of 0.4 BLEU points over the baseline. Output N-best lists were rescored via a target Neural Network Language Model. An improvement of one BLEU point over the baseline was obtained adding the two features, giving 31.5 BLEU and 57.9 TER for the primary system, computed over lowercased and detokenized outputs. The system was positioned second in the final ranking.

1 Introduction

The goal of Statistical Machine Translation (SMT) is to translate a sentence between two languages. Giving the source language sentence $\mathbf{f}$, it would be translated to an equivalent target language sentence $\mathbf{e}$. The most extended formalization is done via log-linear models (Papineni et al., 1998; Och and Ney, 2002) as follows:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \sum_{k=1}^{K} \lambda_k h_k(\mathbf{f}, \mathbf{e}) \qquad (1)$$

where $h_k(\mathbf{f}, \mathbf{e})$ is a score function representing an important feature for the translation of $\mathbf{f}$ into $\mathbf{e}$, K is the number of models (or features) and λ_k are the weights of the log-linear combination. Typically, the weights λ_k are optimized during the tuning stage with the use of a development set.

SMT systems rely on a bilingual sentence aligned training corpus. These sentences are aligned at the word level (Brown et al., 1993), and after that, different h_k feature functions are trained. In some practical cases, the out-of-domain training data is larger than the in-domain training data. In these cases the target Language Model (LM) is composed of a linear interpolation of independent LMs, one for each available training domain or corpus. Nevertheless, the training of phrase-based translation models is an open problem in these cases.

Some recent works (Resnik and Smith, 2003; Yasuda et al., ; Koehn and Schroeder, 2007; Matsoukas et al., 2009; Foster et al., 2010; Sanchis-Trilles and Casacuberta, 2010) related to corpus weighting, make use of data selection, data weighting, and translation model adaptation to overcome this problem. In this work, we explore a simple corpus weighting technique to interpolate any number of different phrase tables. Two different approaches are tested, obtaining similar performance. On the one hand, a count-based smoothing technique that applies a weight to the counting of phrases and lexical links depending on the relevance of each corpus. On the other hand, a linear interpolation of independent trained phrase tables.

Another important feature of this work is the use of Neural Network Language Models (NN LMs) (Bengio, 2008). This kind of LMs has been successfully applied in some connectionist approaches to language modeling (Bengio et al., 2003; Castro-Bleda and Prat, 2003; Schwenk et al., 2006;

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 490–495,
Edinburgh, Scotland, UK, July 30–31, 2011.

Schwenk, 2010). The advantage of these NN LMs is the projection of words on a continuous space were the probabilities of n-grams are learned. A Neural Network (NN) is proposed to learn both the word projections and the n-gram probabilities.

The presented system combines a standard, state-of-the-art SMT system with a NN LM via log-linear combination and N-best output re-scoring. We chose to participate in the English-Spanish direction.

2 Translation models

A standard phrase-based translation model is composed of the following five h_k features:

- inverse phrase translation probability $p(\overline{f}|\overline{e})$
- inverse lexical weighting $l(\overline{f}|\overline{e})$
- direct phrase translation probability $p(\overline{e}|\overline{f})$
- direct lexical weighting $l(\overline{e}|\overline{f})$
- phrase penalty (always $e = 2.718$).

We rely only on the first four features. They are computed from word alignments at the sentence level, by counting over the alignments, and using the inverse and direct lexical dictionaries. Given a pair of phrases, $\overline{f}$ on the source language and $\overline{e}$ in the target language, the phrase translation probabilities are computed by relative frequency as:

$$\begin{aligned} p(\overline{f}|\overline{e}) &= \frac{\text{count}(\overline{f},\overline{e})}{\sum_{e'}\text{count}(\overline{f},\overline{e}')} \\ p(\overline{e}|\overline{f}) &= \frac{\text{count}(\overline{f},\overline{e})}{\sum_{f'}\text{count}(\overline{f}',\overline{e})} \end{aligned}$$

Given a word f on the source language, and a word e in the target language, the lexical translation distribution is computed again by relative frequency as:

$$\begin{aligned} w(f|e) &= \frac{\text{count}(f,e)}{\sum_{e'}\text{count}(f,e')} \\ w(e|f) &= \frac{\text{count}(f,e)}{\sum_{f'}\text{count}(f',e)} \end{aligned}$$

Given the previous lexical translation distribution, two phrase pairs $\overline{f}$ and $\overline{e}$, and a, the word alignment between the source word positions $i = 1, \ldots, n$ and the target word positions $j = 1, \ldots, m$, the inverse lexical weighting is computed as:

$$l(\overline{f}|\overline{e}) = \prod_{i=1}^{n} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{(i,j)\in a} w(f_i|e_j)$$

and the direct lexical weighting is computed as:

$$l(\overline{e}|\overline{f}) = \prod_{j=1}^{m} \frac{1}{|\{i|(i,j) \in a\}|} \sum_{(i,j)\in a} w(e_j|f_i)$$

3 Weighting different translation models

The proposed modifications of the phrase-based translation models are similar to (Foster et al., 2010; Matsoukas et al., 2009), but in this case the weighting is simpler and focused at the corpus level. If we have T different training sets, we could define β_t as the weight of the set t, for $1 \leq t \leq T$. The word alignments are computed via Giza++ (Och and Ney, 2003) over the concatenation of all the training material available for the translation models (in this case, Europarl, News-Commentary, and United Nations). After that, we could recompute the lexical translation distribution using the weights information, and compute the phrase-based translation models taking into account these weights. The `count` function will be redefined to take into account only information of the corresponding training set.

3.1 Count smoothing

The weight β_t is applied to the `count` function, in order to modify the corpus effect on the probability of each phrase pair alignment, and each word pair alignment. First, we modify the lexical translation distribution in this way:

$$\begin{aligned} w(f|e) &= \frac{\sum_t \beta_t \text{count}_t(f,e)}{\sum_t \beta_t \sum_{e'} \text{count}_t(f,e')} \\ w(e|f) &= \frac{\sum_t \beta_t \text{count}_t(f,e)}{\sum_t \beta_t \sum_{f'} \text{count}_t(f',e)} \end{aligned}$$

having a global lexical translation distribution for the alignment between words. Second, we modify the phrase translation probabilities for each direction, remaining without modification the lexical weightings:

$$\begin{aligned} p(\overline{f}|\overline{e}) &= \frac{\sum_t \beta_t \text{count}_t(\overline{f},\overline{e})}{\sum_t \beta_t \sum_{\overline{e}'} \text{count}_t(\overline{f},\overline{e}')} \\ p(\overline{e}|\overline{f}) &= \frac{\sum_t \beta_t \text{count}_t(\overline{f},\overline{e})}{\sum_t \beta_t \sum_{\overline{f}'} \text{count}_t(\overline{f}',\overline{e})} \end{aligned}$$

When some phrase/word count is not found, count is set to zero.

3.2 Linear interpolation

In this case, we compute independently the translation models for each training set. We have T models, one for each set. The final translation models are obtained by means of a linear interpolation of each independent translation model. If some phrase pair is not found, the translation model is set to have zero probability.

First, we redefine the lexical translation distribution. In this case we have $w_1, w_2, \ldots, w_T$ lexical dictionaries:

$$\begin{aligned} w_t(f|e) &= \frac{\text{count}_t(f,e)}{\sum_{e'} \text{count}_t(f,e')} \\ w_t(e|f) &= \frac{\text{count}_t(f,e)}{\sum_{f'} \text{count}_t(f',e)}. \end{aligned}$$

Then, we could compute the linear interpolation of phrase translation probabilities as follows:

$$\begin{aligned} p(\overline{f}|\overline{e}) &= \sum_t \beta_t \frac{\text{count}_t(\overline{f},\overline{e})}{\sum_{\overline{e}'} \text{count}_t(\overline{f},\overline{e}')} \\ p(\overline{e}|\overline{f}) &= \sum_t \beta_t \frac{\text{count}_t(\overline{f},\overline{e})}{\sum_{\overline{f}'} \text{count}_t(\overline{f}',\overline{e})} \end{aligned}$$

And finally, the inverse lexical weighting is obtained as:

$$l(\overline{f}|\overline{e}) = \sum_t \beta_t \prod_{i=1}^{n} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{(i,j)\in a} w_t(f_i|e_j)$$

and the direct lexical weighting:

$$l(\overline{e}|\overline{f}) = \sum_t \beta_t \prod_{j=1}^{m} \frac{1}{|\{i|(i,j) \in a\}|} \sum_{(i,j)\in a} w_t(e_j|f_i).$$

4 Neural Network Language Models

In SMT the most useful language models are n-grams (Bahl et al., 1983; Jelinek, 1997; Bahl et al., 1983). They compute the probability of each word given the context of the $n-1$ previous words:

$$p(s_1 \ldots s_{|S|}) \approx \prod_{i=1}^{|S|} p(s_i|s_{i-n+1} \ldots s_{i-1}) \qquad (2)$$

where S is the sequence of words for which we want compute the probability, and $s_i \in S$, from a vocabulary Ω.

A NN LM is a statistical LM which follows equation (2) as n-grams do, but where the probabilities that appear in that expression are estimated with a NN (Bengio et al., 2003; Castro-Bleda and Prat, 2003; Schwenk, 2007; Bengio, 2008). The model naturally fits under the probabilistic interpretation of the outputs of the NNs: if a NN, in this case a MLP, is trained as a classifier, the outputs associated to each class are estimations of the posterior probabilities of the defined classes (Bishop, 1995).

The training set for a LM is a sequence $s_1 s_2 \ldots s_{|S|}$ of words from a vocabulary Ω. In order to train a NN to predict the next word given a history of length $n-1$, each input word must be encoded. A natural representation is a local encoding following a "1-of-$|\Omega|$" scheme. The problem of this encoding for tasks with large vocabularies (as is typically the case) is the huge size of the resulting NN. We have solved this problem following the ideas of (Bengio et al., 2003; Schwenk, 2007), learning a distributed representation for each word. Figure 1 illustrates the architecture of the feed-forward NN used to estimate the NN LM.

This n-gram NN LM predicts the posterior probability of each word of the vocabulary given the $n-1$ previous words. A single forward pass of the MLP gives $p(\omega|s_{i-n+1} \ldots s_{i-1})$ for every word $\omega \in \Omega$. After training the projection layer is replaced by a table look-up.

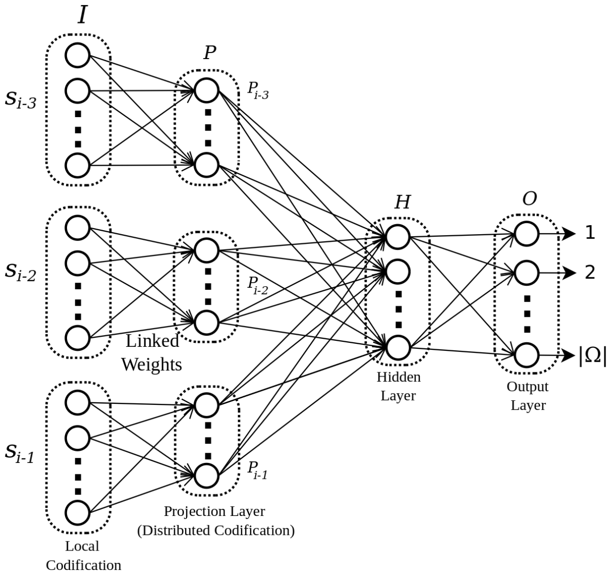

Figure 1: Architecture of the continuous space NN LM during training. The input words are $s_{i-n+1}, \ldots, s_{i-1}$ (in this example, the input words are s_{i-3}, s_{i-2}, and s_{i-1} for a 4-gram). I, P, H, and O are the input, projection, hidden, and output layer, respectively, of the MLP.

Table 1: Spanish corpora statistics. NC stands for News-Commentary and UN for United Nations, while $|\Omega|$ stands for vocabulary size, and M/K for millions/thousands of elements. All numbers are computed with tokenized and lowercased data.

Set	# Lines	# Words	$\|\Omega\|$
NC v6	$159K$	$4.44M$	$80K$
News-Shuffled	$9.17M$	$269M$	$596K$
Europarl v6	$1.94M$	$55M$	$177K$
UN	$6.22M$	$214M$	$579K$
Total	$21.93M$	$678M$	$1.03M$

Table 2: Weights of different combination of phrase-based translation models.

System	Europarl	NC	UN
Smooth1	0.35	0.35	0.30
Smooth2	0.40	0.40	0.20
Smooth3	0.15	0.80	0.05
Linear	0.35	0.35	0.30

The major advantage of the connectionist approach is the automatic smoothing performed by the neural network estimators. This smoothing is done via a continuous space representation of the input words. Learning the probability of n-grams, together with their representation in a continuous space (Bengio et al., 2003), is an appropriate approximation for large vocabulary tasks. However, one of the drawbacks of such approach is the high computational cost entailed whenever the NN LM is computed directly, with no simplification whatsoever. For this reason, the vocabulary size will be restricted in the experiments presented in this work.

5 Experiments

The baseline SMT system is built with the open-source SMT toolkit Moses (Koehn et al., 2007), in its standard setup. The decoder includes a log-linear model comprising a phrase-based translation model, a language model, a lexicalized distortion model and word and phrase penalties. The weights of the log-linear interpolation were optimized by means of MERT (Och, 2003), using the News-Commentary test set of the 2008 shared task as a development set. The phrase-based translation model uses the concatenation of News-Commentary, United Nations, and Europarl corpora, to estimate the four translation model features.

The baseline LM was a regular n-gram LM with Kneser-Ney smoothing (Kneser and Ney, 1995) and interpolation by means of the SRILM toolkit (Stolcke, 2002). Specifically, we trained a 6-gram LM on United Nations, a 5-gram on Europarl and News-Shuffled, and a 4-gram on News-Commentary. Once these LMs had been built, they were interpolated so as to maximize the perplexity of the News-Commentary test set of the 2009 shared task. The final model was pruned out using a threshold of 10^{-8}. This was done so according to preliminary research.

Three different weights for the count smoothing technique described in section 3.1 were tested. For the interpolation model of section 3.2, we select the weights minimizing the perplexity of the corresponding three LMs (Europarl, NC, and UN) over the News2008 set. Table 2 summarizes these weights.

NN LM was trained with all the corpora described in Table 1, using a weighted replacement algorithm to modify the impact of each corpus in the training algorithm. The weights were the same that for the standard LM. In order to reduce the complexity of the model, the input vocabulary of the NN LM was restricted using only words that appears more than 10 times in the corpora. The vocabulary is formed by the 107 607 more frequent words, with two additional inputs: one to represent the words out of this vocabulary, and another for the begin-of-sentence cue. The output of the NN LM was restricted much more, using only a shortlist (Schwenk, 2007) of the $10K$ more frequent words, plus the end-of-sentence cue. The rest of words are collected by an additional output in the neural network. When the probability of an out-of-shortlist word is required, its probability is computed multiplying this additional output activation by the unigram probability distribution of every out-of-shortlist word. This implies that 10.7% of the running words of the News2009 set, and 11.1% of the running words of the News2011 official test set, will be considered as out-of-shortlist words for the NN LM.

A 6-gram NN LM was trained for this task, based in previous works (Zamora-Martínez and Sanchis-Trilles, 2010). Four NN LMs with different values for the projection of each word (128, 192, 256, 320) were linearly combined for the final NN LM. Each NN LM had 320 units in the hidden layer. The combination weights were computed maximizing the perplexity over the News2009 set. The training procedure was conducted by means of the stochastic back-propagation algorithm with weight decay, with a replacement of $300K$ training samples and $200K$ validation samples in each training epoch, selecting the random sample using a different distribution weight for each corpus. The validation set was the News2009 set. The networks were stopped after 99, 70, 53, and 42 epochs respectively (unfortunately, without achieving convergence, due to the competition timings). This resulted in very few training samples compared with the size of the training set: $29M$ in the best case, versus more than $500M$ of the full set. The training of the NN LMs was accomplished with the April toolkit (España-Boquera et al., 2007; Zamora-Martínez et al., 2009). The perplexity achieved by the 6-gram NN LM in the Spanish News2009 set was 281, versus 145 obtained with the standard 6-gram language model with interpolation and Kneser-Ney smoothing (Kneser and Ney, 1995).

Table 3: Main results of the experimentation

	News2010		News2011	
System	BLEU	TER	BLEU	TER
Baseline	29.2	60.0	30.5	58.9
Smooth1	29.3	59.9	–	–
Smooth2	29.2	59.9	–	–
Smooth3	29.5	59.6	30.9	58.5
+ NN LM	29.9	59.2	31.4	58.0
Linear	29.5	59.5	30.9	58.7
+ NN LM	**30.2**	**58.8**	**31.5**	**57.9**

The number of sentences in the N-best list was set to 2 000 unique output sentences. Results can be seen in Table 3. In order to assess the reliability of such results, we computed pairwise improvement intervals as described in (Koehn, 2004), by means of bootstrapping with 1 000 bootstrap iterations and at a 95% confidence level. Such confidence test reported the improvements to be statistically significant. A difference of more than 0.3 points of BLEU is considered significant in the pairwise comparison. The final results leads to 31.5 points of BLEU, positioning this system as second in the final classification.

6 Conclusions and future work

The presented CEU-UPV system, using phrase translation models combinations and NN LMs, leads an improvement of 0.4 points of BLEU in the two cases: the count smoothing approach (Smooth3 system) and the linear interpolation approach (Linear system). The incorporation of NN LMs in both systems gets an additional improvement of 0.5 BLEU points for the Smooth3 system, and 0.6 BLEU points for the Linear system. The final result for the primary system is 31.5 BLEU points.

The combination of translation models could be enhanced optimizing the β_t weights over the BLEU score. Currently the weights are manually set for the Smooth[1,2,3] systems, and fixed to the LM weights for the Linear system. Nevertheless, both approaches achieve similar results. Finally, it is important to emphasize that the use of NN LMs implies an interesting improvement, though this year's gain is lower than that obtained by our 2010 system.[1]

[1] Note that the NN LMs didn't achieve convergence due to

Acknowledgments

This work has been partially supported by the Spanish Goverment under project TIN2010-1958.

References

L. R. Bahl, F. Jelinek, and R. L. Mercer. 1983. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Trans. on Pat. Anal. and Mach. Intel.*, 5(2):179–190.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3(2):1137–1155.

Y. Bengio. 2008. Neural net language models. *Scholarpedia*, 3(1):3881.

C. M. Bishop. 1995. *Neural networks for pattern recognition*. Oxford University Press.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311.

M.J. Castro-Bleda and F. Prat. 2003. New Directions in Connectionist Language Modeling. In *Computational Methods in Neural Modeling*, volume 2686 of *LNCS*, pages 598–605. Springer-Verlag.

S. España-Boquera, F. Zamora-Martínez, M.J. Castro-Bleda, and J. Gorbe-Moya. 2007. Efficient BP Algorithms for General Feedforward Neural Networks. In *Bio-inspired Modeling of Cognitive Tasks*, volume 4527 of *LNCS*, pages 327–336. Springer.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proc. of EMNLP*, EMNLP'10, pages 451–459, Stroudsburg, PA, USA. Association for Computational Linguistics.

F. Jelinek. 1997. *Statistical Methods for Speech Recognition*. Language, Speech, and Communication. The MIT Press.

R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'95)*, volume II, pages 181–184, May.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proc. of WMT'07*, pages 224–227.

P. Koehn et al. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of the ACL'07 Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*, EMNLP'04, pages 388–395. Association for Computational Linguistics.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proc. of EMNLP'09*, volume 2, pages 708–717, Stroudsburg, PA, USA. Association for Computational Linguistics.

F. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of ACL'02*, pages 295–302.

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

F.J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of ACL'03*, pages 160–167, Sapporo, Japan.

K. Papineni, S. Roukos, and T. Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'98)*, pages 189–192.

Philip Resnik and Noah A. Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29:349–380.

Germán Sanchis-Trilles and Francisco Casacuberta. 2010. Bayesian adaptation for statistical machine translation. In *Proc. of SSSPR'10*, pages 620–629.

H. Schwenk, D. Déchelotte, and J. L. Gauvain. 2006. Continuous space language models for statistical machine translation. In *Proc. of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 723–730.

H. Schwenk. 2007. Continuous space language models. *Computer Speech and Language*, 21(3):492–518.

H. Schwenk. 2010. Continuous space language models for statistical machine translation. *The Prague Bulletin of Mathematical Linguistics*, 93.

A. Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proc. of the International Conference in Spoken Language Processing (ICSLP'02)*, pages 901–904, September.

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumita. Method of selecting training data to build a compact and efficient translation model. In *Proc. of IJCNLP'10*, pages 655–660.

Francisco Zamora-Martínez and Germán Sanchis-Trilles. 2010. UCH-UPV English–Spanish System for WMT10. In *Proc. of WMT'10*, pages 207–211, July.

F. Zamora-Martínez, M.J. Castro-Bleda, and S. España-Boquera. 2009. Fast Evaluation of Connectionist Language Models. In *IWANN*, volume 5517 of *LNCS*, pages 33–40. Springer.

competition timings.