Continuous space translation models with neural networks

Le Hai Son, Alexandre Allauzen, François Yvon

Continuous Space Translation Models with Neural Networks

Le Hai Son and **Alexandre Allauzen** and **François Yvon**
Univ. Paris-Sud, France and LIMSI/CNRS
rue John von Neumann, 91403 Orsay cedex, France
Firstname.Lastname@limsi.fr

Abstract

The use of conventional maximum likelihood estimates hinders the performance of existing phrase-based translation models. For lack of sufficient training data, most models only consider a small amount of context. As a partial remedy, we explore here several continuous space translation models, where translation probabilities are estimated using a continuous representation of translation units in lieu of standard discrete representations. In order to handle a large set of translation units, these representations and the associated estimates are jointly computed using a multi-layer neural network with a SOUL architecture. In small scale and large scale English to French experiments, we show that the resulting models can effectively be trained and used on top of a n-gram translation system, delivering significant improvements in performance.

1 Introduction

The phrase-based approach to statistical machine translation (SMT) is based on the following inference rule, which, given a source sentence $\mathbf{s}$, selects the target sentence $\mathbf{t}$ and the underlying alignment $\mathbf{a}$ maximizing the following term:

$$P(\mathbf{t}, \mathbf{a}|\mathbf{s}) = \frac{1}{Z(\mathbf{s})} \exp\Big(\sum_{k=1}^{K} \lambda_k f_k(\mathbf{s}, \mathbf{t}, \mathbf{a})\Big), \quad (1)$$

where K feature functions (f_k) are weighted by a set of coefficients (λ_k), and Z is a normalizing factor. The phrase-based approach differs from other approaches by the hidden variables of the translation process: the segmentation of a parallel sentence pair into phrase pairs and the associated phrase alignments.

This formulation was introduced in (Zens et al., 2002) as an extension of the word based models (Brown et al., 1993), then later motivated within a discriminative framework (Och and Ney, 2004). One motivation for integrating more feature functions was to improve the estimation of the translation model $P(\mathbf{t}|\mathbf{s})$, which was initially based on relative frequencies, thus yielding poor estimates.

This is because the units of phrase-based models are *phrase pairs*, made of a source and a target phrase; such units are viewed as the events of discrete random variables. The resulting representations of phrases (or words) thus entirely ignore the morphological, syntactic and semantic relationships that exist among those units in both languages. This lack of structure hinders the generalization power of the model and reduces its ability to adapt to other domains. Another consequence is that phrase-based models usually consider a very restricted context[1].

This is a general issue in statistical Natural Language Processing (NLP) and many possible remedies have been proposed in the literature, such as, for instance, using smoothing techniques (Chen and Goodman, 1996), or working with linguistically enriched, or more abstract, representations. In statistical language modeling, another line of research considers numerical representations, trained automatically through the use of neural network (see eg.

[1] typically a small number of preceding phrase pairs for the n-gram based approach (Crego and Mariño, 2006), or no context at all, for the standard approach of (Koehn et al., 2007).

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 39–48,
Montréal, Canada, June 3-8, 2012.

(Collobert et al., 2011)). An influential proposal, in this respect, is the work of (Bengio et al., 2003) on *continuous space language models*. In this approach, n-gram probabilities are estimated using a continuous representation of words in lieu of standard discrete representations. Experimental results, reported for instance in (Schwenk, 2007) show significant improvements in speech recognition applications. Recently, this model has been extended in several promising ways (Mikolov et al., 2011; Kuo et al., 2010; Liu et al., 2011). In the context of SMT, Schwenk et al. (2007) is the first attempt to estimate translation probabilities in a continuous space. However, because of the proposed neural architecture, the authors only consider a very restricted set of translation units, and therefore report only a slight impact on translation performance. The recent proposal of (Le et al., 2011a) seems especially relevant, as it is able, through the use of class-based models, to handle arbitrarily large vocabularies and opens the way to enhanced neural translation models.

In this paper, we explore various neural architectures for translation models and consider three different ways to factor the joint probability $P(\mathbf{s}, \mathbf{t})$ differing by the units (respectively phrase pairs, phrases or words) that are projected in continuous spaces. While these decompositions are theoretically straightforward, they were not considered in the past because of data sparsity issues and of the resulting weaknesses of conventional *maximum likelihood* estimates. Our main contribution is then to show that such joint distributions can be efficiently computed by neural networks, even for very large context sizes; and that their use yields significant performance improvements. These models are evaluated in a n-best rescoring step using the framework of n-gram based systems, within which they integrate easily. Note, however that they could be used with any phrase-based system.

The rest of this paper is organized as follows. We first recollect, in Section 2, the n-gram based approach, and discuss various implementations of this framework. We then describe, in Section 3, the neural architecture developed and explain how it can be made to handle large vocabulary tasks as well as language models over bilingual units. We finally report, in Section 4, experimental results obtained on a large-scale English to French translation task.

2 Variations on the n-gram approach

Even though n-gram translation models can be integrated within standard phrase-based systems (Niehues et al., 2011), the n-gram based framework provides a more convenient way to introduce our work and has also been used to build the baseline systems used in our experiments. In the n-gram based approach (Casacuberta and Vidal, 2004; Mariño et al., 2006; Crego and Mariño, 2006), translation is divided in two steps: a source reordering step and a translation step. Source reordering is based on a set of learned rewrite rules that non-deterministically reorder the input words so as to match the target order thereby generating a lattice of possible reorderings. Translation then amounts to finding the most likely path in this lattice using a n-gram translation model [2] of *bilingual units*.

2.1 The standard n-gram translation model

n-gram translation models (TMs) rely on a specific decomposition of the joint probability $P(\mathbf{s}, \mathbf{t})$, where $\mathbf{s}$ is a sequence of I reordered source words $(s_1, ..., s_I)$ and $\mathbf{t}$ contains J target words $(t_1, ..., t_J)$. This sentence pair is further assumed to be decomposed into a sequence of L bilingual units called *tuples* defining a joint segmentation: $(\mathbf{s}, \mathbf{t}) = u_1, ..., u_L$. In the approach of (Mariño et al., 2006), this segmentation is a by-product of source reordering, and ultimately derives from initial word and phrase alignments. In this framework, the basic translation units are *tuples*, which are the analogous of phrase pairs, and represent a matching $u = (\overline{s}, \overline{t})$ between a source $\overline{s}$ and a target $\overline{t}$ phrase (see Figure 1). Using the n-gram assumption, the joint probability of a segmented sentence pair decomposes as:

$$P(\mathbf{s}, \mathbf{t}) = \prod_{i=1}^{L} P(u_i | u_{i-1}, ..., u_{i-n+1}) \qquad (2)$$

A first issue with this model is that the elementary units are bilingual pairs, which means that the underlying vocabulary, hence the number of parameters, can be quite large, even for small translation tasks. Due to data sparsity issues, such models are bound

[2] Like in the standard phrase-based approach, the translation process also involves additional feature functions that are presented below.

to face severe estimation problems. Another problem with (2) is that the source and target sides play symmetric roles, whereas the source side is known, and the target side must be predicted.

2.2 A factored n-gram translation model

To overcome some of these issues, the n-gram probability in equation (2) can be factored by decomposing tuples in two (source and target) parts :

$$\begin{aligned} &P(u_i|u_{i-1},...,u_{i-n+1}) = \\ &\quad P(\bar{t}_i|\bar{s}_i,\bar{s}_{i-1},\bar{t}_{i-1},...,\bar{s}_{i-n+1},\bar{t}_{i-n+1}) \\ &\quad \times P(\bar{s}_i|\bar{s}_{i-1},\bar{t}_{i-1}...,\bar{s}_{i-n+1},\bar{t}_{i-n+1}) \end{aligned} \quad (3)$$

Decomposition (3) involves two models: the first term represents a TM, the second term is best viewed as a reordering model. In this formulation, the TM only predicts the target phrase, given its source and target contexts.

Another benefit of this formulation is that the elementary events now correspond either to source or to target phrases, but never to pairs of such phrases. The underlying vocabulary is thus obtained as the union, rather than the cross product, of phrase inventories. Finally note that the n-gram probability $P(u_i|u_{i-1},...,u_{i-n+1})$ could also factor as:

$$\begin{aligned} &P(\bar{s}_i|\bar{t}_i,\bar{s}_{i-1},\bar{t}_{i-1},...,\bar{s}_{i-n+1},\bar{t}_{i-n+1}) \\ &\times P(\bar{t}_i|\bar{s}_{i-1},\bar{t}_{i-1},...,\bar{s}_{i-n+1},\bar{t}_{i-n+1}) \end{aligned} \quad (4)$$

2.3 A word factored translation model

A more radical way to address the data sparsity issues is to take (source and target) words as the basic units of the n-gram TM. This may seem to be a step backwards, since the transition from word (Brown et al., 1993) to phrase-based models (Zens et al., 2002) is considered as one of the main recent improvement in MT. One important motivation for considering phrases rather than words was to capture local context in translation and reordering. It should then be stressed that the decomposition of phrases in words is only re-introduced here as a way to mitigate the parameter estimation problems. Translation units are still pairs of *phrases*, derived from a bilingual segmentation in tuples synchronizing the source and target n-gram streams, as defined by equation (3). In fact, the estimation policy described in section 3 will actually allow us to design n-gram models with *longer contexts* than is typically possible in the conventional n-gram approach.

Let s_i^k denote the k^{th} word of source tuple $\bar{s}_i$. Considering again the example of Figure 1, s_{11}^1 is to the source word *nobel*, s_{11}^4 is to the source word *paix*, and similarly t_{11}^2 is the target word *peace*. We finally denote $h^{n-1}(t_i^k)$ the sequence made of the $n-1$ words preceding t_i^k in the target sentence: in Figure 1, $h^3(t_{11}^2)$ thus refers to the three word context *receive the nobel* associated with the target word *peace*. Using these notations, equation (3) is rewritten as:

$$\begin{aligned} P(\mathbf{s},\mathbf{t}) = \prod_{i=1}^{L} \Big[&\prod_{k=1}^{|\bar{t}_i|} P\big(t_i^k|h^{n-1}(t_i^k), h^{n-1}(s_{i+1}^1)\big) \\ &\times \prod_{k=1}^{|\bar{s}_i|} P\big(s_i^k|h^{n-1}(t_i^1), h^{n-1}(s_i^k)\big) \Big] \end{aligned} \quad (5)$$

This decomposition relies on the n-gram assumption, this time at the word level. Therefore, this model estimates the joint probability of a sentence pair using two sliding windows of length n, one for each language; however, the moves of these windows remain synchronized by the tuple segmentation. Moreover, the context is not limited to the current phrase, and continues to include words in adjacent phrases. Using the example of Figure 1, the contribution of the target phrase $\bar{t}_{11}$ = *nobel, peace* to $P(\mathbf{s},\mathbf{t})$ using a 3- gram model is

$$\begin{aligned} &P\big(\text{nobel}|[\text{receive, the}],[\text{la, paix}]\big) \\ &\times P\big(\text{peace}|[\text{the, nobel}],[\text{la, paix}]\big). \end{aligned}$$

Likewise, the contribution of the source phrase $\bar{s}_{11}$ =*nobel, de, la, paix* is:

$$\begin{aligned} &P\big(\text{nobel}|[\text{receive, the}],[\text{recevoir,le}]\big) \\ &\quad \times P\big(\text{de}|[\text{receive, the}],[\text{le,nobel}]\big) \\ &\quad \times P\big(\text{la}|[\text{receive, the}],[\text{nobel, de}]\big) \\ &\quad \times P\big(\text{paix}|[\text{receive, the}],[\text{de,la}]\big). \end{aligned}$$

A benefit of this new formulation is that the involved vocabularies only contain words, and are thus much smaller. These models are thus less bound to be affected by data sparsity issues. While the TM defined by equation (5) derives from equation (3), a variation can be equivalently derived from equation (4).

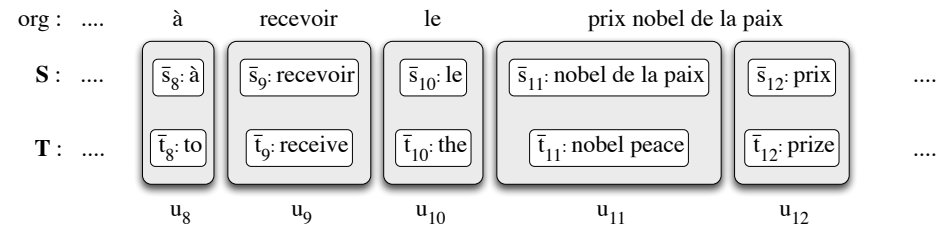

Figure 1: Extract of a French-English sentence pair segmented in bilingual units. The original (*org*) French sentence appears at the top of the figure, just above the reordered source $\mathbf{s}$ and target $\mathbf{t}$. The pair $(\mathbf{s}, \mathbf{t})$ decomposes into a sequence of L bilingual units (*tuples*) $u_1, ..., u_L$. Each tuple u_i contains a source and a target phrase: $\bar{s}_i$ and $\bar{t}_i$.

3 The SOUL model

In the previous section, we defined three different n-gram translation models, based respectively on equations (2), (3) and (5). As discussed above, a major issue with such models is to reliably estimate their parameters, the numbers of which grow exponentially with the order of the model. This problem is aggravated in natural language processing, due to well known data sparsity issues. In this work, we take advantage of the recent proposal of (Le et al., 2011a): using a specific neural network architecture (the *Structured OUtput Layer* model), it becomes possible to handle large vocabulary language modeling tasks, a solution that we adapt here to MT.

3.1 Language modeling in a continuous space

Let $\mathcal{V}$ be a finite vocabulary, n-gram language models (LMs) define distributions over finite sequences of tokens (typically words) w_1^L in $\mathcal{V}^+$ as follows:

$$P(w_1^L) = \prod_{i=1}^{L} P(w_i | w_{i-n+1}^{i-1}) \qquad (6)$$

Modeling the joint distribution of several discrete random variables (such as words in a sentence) is difficult, especially in NLP applications where $\mathcal{V}$ typically contains dozens of thousands words.

In spite of the simplifying n-gram assumption, maximum likelihood estimation remains unreliable and tends to underestimate the probability of very rare n-grams. Smoothing techniques, such as Kneser-Ney and Witten-Bell backoff schemes (see (Chen and Goodman, 1996) for an empirical overview, and (Teh, 2006) for a Bayesian interpretation), perform back-off to lower order distributions, thus providing an estimate for the probability of these unseen events.

One of the most successful alternative to date is to use *distributed word representations* (Bengio et al., 2003), where distributionally similar words are represented as neighbors in a continuous space. This turns n-grams distributions into smooth functions of the word representations. These representations and the associated estimates are jointly computed in a multi-layer neural network architecture. Figure 2 provides a partial representation of this kind of model and helps figuring out their principles. To compute the probability $P(w_i | w_{i-n+1}^{i-1})$, the $n-1$ context words are projected in the same continuous space using a shared matrix R; these continuous word representations are then concatenated to build a single vector that represents the context; after a non-linear transformation, the probability distribution is computed using a softmax layer.

The major difficulty with the neural network approach remains the complexity of inference and training, which largely depends on the size of the output vocabulary (i.e. the number of words that have to be predicted). One practical solution is to restrict the output vocabulary to a short-list composed of the most frequent words (Schwenk, 2007). However, the usual size of the short-list is under 20k, which does not seem sufficient to faithfully represent the translation models of section 2.

3.2 Principles of SOUL

To circumvent this problem, Structured Output Layer (SOUL) LMs are introduced in (Le et al., 2011a). Following Mnih and Hinton (2008), the SOUL model combines the neural network approach with a class-based LM (Brown et al., 1992). Struc-

turing the output layer and using word class information makes the estimation of distributions over the entire vocabulary computationally feasible.

To meet this goal, the output vocabulary is structured as a clustering tree, where each word belongs to only one class and its associated sub-classes. If w_i denotes the i^{th} word in a sentence, the sequence $c_{1:D}(w_i) = c_1, \ldots, c_D$ encodes the path for word w_i in the clustering tree, with D being the depth of the tree, $c_d(w_i)$ a class or sub-class assigned to w_i, and $c_D(w_i)$ being the leaf associated with w_i (the word itself). The probability of w_i given its history h can then be computed as:

$$P(w_i|h) = P(c_1(w_i)|h) \times \prod_{d=2}^{D} P(c_d(w_i)|h, c_{1:d-1}). \quad (7)$$

There is a softmax function at each level of the tree and each word ends up forming its own class (a leaf).

The SOUL model, represented on Figure 2, is thus the same as for the standard model up to the output layer. The main difference lies in the output structure which involves several layers with a softmax activation function. The first *(class layer)* estimates the class probability $P(c_1(w_i)|h)$, while other output *sub-class layers* estimate the sub-class probabilities $P(c_d(w_i)|h, c_{1:d-1})$. Finally, the *word layers* estimate the word probabilities $P(c_D(w_i)|h, c_{1:D-1})$. Words in the short-list remain special, since each of them represents a (final) class.

Training a SOUL model can be achieved by maximizing the log-likelihood of the parameters on some training corpus. Following (Bengio et al., 2003), this optimization is performed by stochastic back-propagation. Details of the training procedure can be found in (Le et al., 2011b).

Neural network architectures are also interesting as they can easily handle larger contexts than typical n-gram models. In the SOUL architecture, enlarging the context mainly consists in increasing the size of the projection layer, which corresponds to a simple look-up operation. Increasing the context length at the input layer thus only causes a linear growth in complexity in the worst case (Schwenk, 2007).

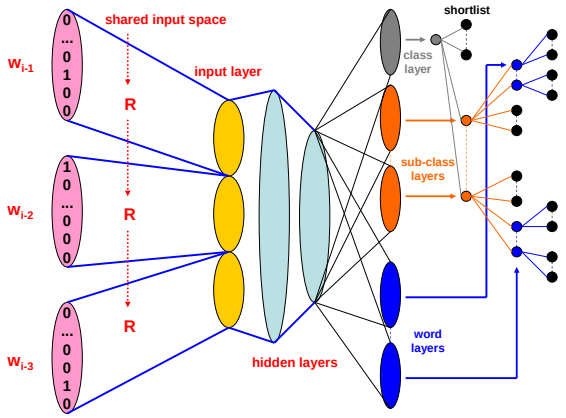

Figure 2: *The architecture of a SOUL Neural Network language model in the case of a* 4*-gram model.*

3.3 Translation modeling with SOUL

The SOUL architecture was used successfully to deliver (monolingual) LMs probabilities for speech recognition (Le et al., 2011a) and machine translation (Allauzen et al., 2011) applications. In fact, using this architecture, it is possible to estimate n-gram distributions for any kind of discrete random variables, such as a phrase or a tuple. The SOUL architecture can thus be readily used as a replacement for the standard n-gram TM described in section 2.1. This is because all the random variables are events over the same set of tuples.

Adopting this architecture for the other n-gram TM respectively described by equations (3) and (5) is more tricky, as they involve two different languages and thus two different vocabularies: the predicted unit is a target phrase (resp. word), whereas the context is made of both source and target phrases (resp. words). A subsequent modification of the SOUL architecture was thus performed to make up for "mixed" contexts: rather than projecting all the context words or phrases into the same continuous space (using the matrix R, see Figure 2), we used two different projection matrices, one for each language. The input layer is thus composed of two vectors in two different spaces; these two representations are then combined through the hidden layer, the other layers remaining unchanged.

4 Experimental Results

We now turn to an experimental comparison of the models introduced in Section 2. We first describe the tasks and data that were used, before presenting our n-gram based system and baseline set-up. Our results are finally presented and discussed.

Let us first emphasize that the design and integration of a SOUL model for large SMT tasks is far from easy, given the computational cost of computing n-gram probabilities, a task that is performed repeatedly during the search of the best translation. Our solution was to resort to a two pass approach: the first pass uses a conventional back-off n-gram model to produce a k-best list (the k most likely translations); in the second pass, the probability of a m-gram SOUL model is computed for each hypothesis, added as a new feature and the k-best list is accordingly reordered[3]. In all the following experiments, we used a fixed context size for SOUL of $m = 10$, and used $k = 300$.

4.1 Tasks and corpora

The two tasks considered in our experiments are adapted from the text translation track of IWSLT 2011 from English to French (the "TED" talk task): a *small data* scenario where the only training data is a small in-domain corpus; and a large scale condition using all the available training data. In this article, we only provide a short overview of the task; all the necessary details regarding this evaluation campaign are on the official website[4].

The in-domain training data consists of $107,058$ sentence pairs, whereas for the large scale task, all the data available for the WMT 2011 evaluation[5] are added. For the latter task, the available parallel data includes a large Web corpus, referred to as the GigaWord parallel corpus. This corpus is very noisy and is accordingly filtered using a simple perplexity criterion as explained in (Allauzen et al., 2011). The total amount of training data is approximately 11.5 million sentence pairs for the bilingual part, and about 2.5 billion of words for the monolingual part. As the provided development data was quite small, development and test set were inverted, and we finally used a development set of 1,664 sentences, and a test set of 934 sentences. The table 1 provides the sizes of the different vocabularies. The n-gram TMs are estimated over a training corpus composed of tuple sequences. Tuples are extracted from the word-aligned parallel data (using MGIZA++[6] with default settings) in such a way that a unique segmentation of the bilingual corpus is achieved, allowing to directly estimate bilingual n-gram models (see (Crego and Mariño, 2006) for details).

Model	**Vocabulary size**			
	Small task		*Large task*	
	src	*trg*	*src*	*trg*
Standard	$317k$		$8847k$	
Phrase factored	$96k$	$131k$	$4262k$	$3972k$
Word factored	$45k$	$53k$	$505k$	$492k$

Table 1: Vocabulary sizes for the English to French tasks obtained with various SOUL translation (TM) models. For the factored models, sizes are indicated for both source (*src*) and target (*trg*) sides.

4.2 n-gram based translation system

The n-gram based system used here is based on an open source implementation described in (Crego et al., 2011). In a nutshell, the TM is implemented as a stochastic finite-state transducer trained using a *n*-gram model of (source, target) pairs as described in section 2.1. Training this model requires to reorder source sentences so as to match the target word order. This is performed by a non-deterministic finite-state reordering model, which uses part-of-speech information generated by the TreeTagger to generalize reordering patterns beyond lexical regularities.

In addition to the TM, fourteen feature functions are included: a *target-language model*; four *lexicon models*; six *lexicalized reordering models* (Tillmann, 2004; Crego et al., 2011); a distance-based *distortion model*; and finally a *word-bonus model* and a *tuple-bonus model*. The four *lexicon models* are similar to the ones used in standard phrase-based systems: two scores correspond to the relative frequencies of the tuples and two lexical weights are estimated from the automatically generated word

[3]The probability estimated with the SOUL model is added as a new feature to the score of an hypothesis given by Equation 1. The coefficients are retuned before the reranking step.

[4]iwslt2011.org

[5]www.statmt.org/wmt11/

[6]geek.kyloo.net/software

alignments. The weights associated to feature functions are optimally combined using the Minimum Error Rate Training (MERT) (Och, 2003). All the results in BLEU are obtained as an average of 4 optimization runs[7].

For the small task, the target LM is a standard 4-gram model estimated with the Kneser-Ney discounting scheme interpolated with lower order models (Kneser and Ney, 1995; Chen and Goodman, 1996), while for the large task, the target LM is obtained by linear interpolation of several 4-gram models (see (Lavergne et al., 2011) for details). As for the TM, all the available parallel corpora were simply pooled together to train a 3-gram model. Results obtained with this large-scale system were found to be comparable to some of the best official submissions.

4.3 Small task evaluation

Table 2 summarizes the results obtained with the baseline and different SOUL models, TMs and a target LM. The first comparison concerns the standard n-gram TM, defined by equation (2), when estimated conventionally or as a SOUL model. Adding the latter model yields a slight BLEU improvement of 0.5 point over the baseline. When the SOUL TM is phrased factored as defined in equation (3) the gain is of 0.9 BLEU point instead. This difference can be explained by the smaller vocabularies used in the latter model, and its improved robustness to data sparsity issues. Additional gains are obtained with the word factored TM defined by equation (5): a BLEU improvement of 0.8 point over the phrase factored TM and of 1.7 point over the baseline are respectively achieved. We assume that the observed improvements can be explained by the joint effect of a better smoothing and a longer context.

The comparison with the condition where we only use a SOUL target LM is interesting as well. Here, the use of the word factored TM still yields to a 0.6 BLEU improvement. This result shows that there is an actual benefit in smoothing the TM estimates, rather than only focus on the LM estimates.

Table 3 reports a comparison among the different components and variations of the word

[7]The standard deviations are below 0.1 and thus omitted in the reported results.

Model	**BLEU**	
	dev	*test*
Baseline	31.4	25.8
Adding a SOUL model		
Standard TM	32.0	26.3
Phrase factored TM	32.7	26.7
Word factored TM	33.6	**27.5**
Target LM	32.6	26.9

Table 2: Results for the small English to French task obtained with the baseline system and with various SOUL translation (TM) or target language (LM) models.

Model	**BLEU**	
	dev	*test*
Adding a SOUL model		
+ $P\big(t_i^k \mid h^{n-1}(t_i^k), h^{n-1}(s_{i+1}^1)\big)$	32.6	**26.9**
+ $P\big(s_i^k \mid h^{n-1}(t_i^1), h^{n-1}(s_i^k)\big)$	32.1	26.2
+ the combination of both	33.2	27.5
+ $P\big(s_i^k \mid h^{n-1}(s_i^k), h^{n-1}(t_{i+1}^1)\big)$	31.7	26.1
+ $P\big(t_i^k \mid h^{n-1}(s_i^1), h^{n-1}(t_i^k)\big)$	32.7	**26.8**
+ the combination of both	33.4	27.2

Table 3: Comparison of the different components and variations of the word factored translation model.

factored TM. In the upper part of the table, the model defined by equation (5) is evaluated component by component: first the translation term $P\big(t_i^k \mid h^{n-1}(t_i^k), h^{n-1}(s_{i+1}^1)\big)$, then its distortion counterpart $P\big(s_i^k \mid h^{n-1}(t_i^1), h^{n-1}(s_i^k)\big)$ and finally their combination, which yields the joint probability of the sentence pair. Here, we observe that the best improvement is obtained with the translation term, which is 0.7 BLEU point better than the latter term. Moreover, the use of just a translation term only yields a BLEU score equal to the one obtained with the SOUL target LM, and its combination with the distortion term is decisive to attain the additional gain of 0.6 BLEU point. The lower part of the table provides the same comparison, but for the variation of the word factored TM. Besides a similar trend, we observe that this variation delivers slightly lower results. This can be explained by the restricted context used by the translation term which no longer includes the current source phrase or word.

Model	BLEU	
	dev	*test*
Baseline	33.7	28.2
Adding a word factored SOUL TM		
+ in-domain TM	35.2	29.4
+ out-of-domain TM	34.8	29.1
+ out-of-domain adapted TM	35.5	**29.8**
Adding a SOUL LM		
+ out-of-domain adapted LM	35.0	29.2

Table 4: Results for the large English to French translation task obtained by adding various SOUL translation and language models (see text for details).

4.4 Large task evaluation

For the large-scale setting, the training material increases drastically with the use of the additional out-of-domain data for the baseline models. Results are summarized in Table 4. The first observation is the large increase of BLEU (+2.4 points) for the baseline system over the small-scale baseline. For this task, only the word factored TM is evaluated since it significantly outperforms the others on the small task (see section 4.3).

In a first scenario, we use a word factored TM, trained only on the small in-domain corpus. Even though the training corpus of the baseline TM is one hundred times larger than this small in-domain data, adding the SOUL TM still yields a BLEU increase of 1.2 point[8]. In a second scenario, we increase the training corpus for the SOUL, and include parts of the out-of-domain data (the WMT part). The resulting BLEU score is here slightly worse than the one obtained with just the in-domain TM, yet delivering improved results with the respect to the baseline.

In a last attempt, we amended the training regime of the neural network. In a fist step, we trained conventionally a SOUL model using the same out-of-domain parallel data as before. We then *adapted* this model by running five additional epochs of the back-propagation algorithm using the in-domain data. Using this adapted model yielded our best results to date with a BLEU improvement of 1.6 points over the baseline results. Moreover, the gains obtained using this simple domain adaptation strategy are respectively of +0.4 and +0.8 BLEU, as compared with the small in-domain model and the large out-of-domain model. These results show that the SOUL TM can scale efficiently and that its structure is well suited for domain adaptation.

[8]Note that the in-domain data was already included in the training corpus of the baseline TM.

5 Related work

To the best of our knowledge, the first work on machine translation in continuous spaces is (Schwenk et al., 2007), where the authors introduced the model referred here to as the the standard n-gram translation model in Section 2.1. This model is an extension of the continuous space language model of (Bengio et al., 2003), the basic unit is the tuple (or equivalently the phrase pair). The resulting vocabulary being too large to be handled by neural networks without a structured output layer, the authors had thus to restrict the set of the predicted units to a 8k short-list . Moreover, in (Zamora-Martinez et al., 2010), the authors propose a tighter integration of a continuous space model with a n-gram approach but only for the target LM.

A different approach, described in (Sarikaya et al., 2008), divides the problem in two parts: first the continuous representation is obtained by an adaptation of the Latent Semantic Analysis; then a Gaussian mixture model is learned using this continuous representation and included in a hidden Markov model. One problem with this approach is the separation between the training of the continuous representation on the one hand, and the training of the translation model on the other hand. In comparison, in our approach, the representation and the prediction are learned in a joined fashion.

Other ways to address the data sparsity issues faced by translation model were also proposed in the literature. Smoothing is obviously one possibility (Foster et al., 2006). Another is to use *factored language models*, introduced in (Bilmes and Kirchhoff, 2003), then adapted for translation models in (Koehn and Hoang, 2007; Crego and Yvon, 2010). Such approaches require to use external linguistic analysis tools which are error prone; moreover, they did not seem to bring clear improvements, even when translating into morphologically rich languages.

6 Conclusion

In this paper, we have presented possible ways to use a neural network architecture as a translation model. A first contribution was to produce the first large-scale neural translation model, implemented here in the framework of the n-gram based models, taking advantage of a specific hierarchical architecture (SOUL). By considering several decompositions of the joint probability of a sentence pair, several bilingual translation models were presented and discussed. As it turned out, using a factorization which clearly distinguishes the source and target sides, and only involves word probabilities, proved an effective remedy to data sparsity issues and provided significant improvements over the baseline. Moreover, this approach was also experimented within the systems we submitted to the shared translation task of the seventh workshop on statistical machine translation (WMT 2012). These experimentations in a large scale setup and for different language pair corroborate the improvements reported in this article.

We also investigated various training regimes for these models in a cross domain adaptation setting. Our results show that adapting an out-of-domain SOUL TM is both an effective and very fast way to perform bilingual model adaptation. Adding up all these novelties finally brought us a 1.6 BLEU point improvement over the baseline. Even though our experiments were carried out only within the framework of n-gram based MT systems, using such models in other systems is straightforward. Future work will thus aim at introducing them into conventional phrase-based systems, such as Moses (Koehn et al., 2007). Given that Moses only implicitly uses n-gram based information, adding SOUL translation models is expected to be even more helpful.

Acknowledgments

This work was partially funded by the French State agency for innovation (OSEO), in the Quaero Programme.

References

Alexandre Allauzen, Gilles Adda, Hélène Bonneau-Maynard, Josep M. Crego, Hai-Son Le, Aurélien Max, Adrien Lardilleux, Thomas Lavergne, Artem Sokolov, Guillaume Wisniewski, and François Yvon. 2011. LIMSI @ WMT11. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 309–315, Edinburgh, Scotland.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *JMLR*, 3:1137–1155.

Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 4–6.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

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

Francesco Casacuberta and Enrique Vidal. 2004. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(3):205–225.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. ACL'96*, pages 310–318, San Francisco.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Josep M. Crego and José B. Mariño. 2006. Improving statistical MT by coupling reordering and decoding. *Machine Translation*, 20(3):199–215.

Josep M. Crego and François Yvon. 2010. Factored bilingual n-gram language models for statistical machine translation. *Machine Translation*, pages 1–17.

Josep M. Crego, François Yvon, and José B. Mariño. 2011. N-code: an open-source Bilingual N-gram SMT Toolkit. *Prague Bulletin of Mathematical Linguistics*, 96:49–58.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrase-table smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61, Sydney, Australia.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 181–184, Detroit, Michigan.

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

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 *Proc. ACL'07*, pages 177–180, Prague, Czech Republic.

Hong-Kwang Kuo, Lidia Mangu, Ahmad Emami, and Imed Zitouni. 2010. Morphological and syntactic features for Arabic speech recognition. In *Proc. ICASSP 2010*.

Thomas Lavergne, Alexandre Allauzen, Hai-Son Le, and François Yvon. 2011. LIMSI's experiments in domain adaptation for IWSLT11. In *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT), San Francisco, CA*.

Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon. 2011a. Structured output layer neural network language model. In *Proceedings of ICASSP'11*, pages 5524–5527.

Hai-Son Le, Ilya Oparin, Abdel Messaoudi, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon. 2011b. Large vocabulary SOUL neural network language models. In *Proceedings of InterSpeech 2011*.

Xunying Liu, Mark J. F. Gales, and Philip C. Woodland. 2011. Improving lvcsr system combination using neural network language model cross adaptation. In *INTERSPEECH*, pages 2857–2860.

José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrick Lambert, José A.R. Fonollosa, and Marta R. Costa-Jussà. 2006. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Proc. of ICASSP'11*, pages 5528–5531.

Andriy Mnih and Geoffrey E Hinton. 2008. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, volume 21, pages 1081–1088.

Jan Niehues, Teresa Herrmann, Stephan Vogel, and Alex Waibel. 2011. Wider context by using bilingual language models in machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 198–206, Edinburgh, Scotland. Association for Computational Linguistics.

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

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proc. of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167.

Ruhi Sarikaya, Yonggang Deng, Mohamed Afify, Brian Kingsbury, and Yuqing Gao. 2008. Machine translation in continuous space. In *Proceedings of Interspeech*, pages 2350–2353, Brisbane, Australia.

Holger Schwenk, Marta R. Costa-Jussà, and José A.R. Fonollosa. 2007. Smooth bilingual n-gram translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 430–438, Prague, Czech Republic.

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

Yeh W. Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proc. of ACL'06*, pages 985–992, Sidney, Australia.

Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL 2004*, pages 101–104.

Francisco Zamora-Martinez, Maria José Castro-Bleda, and Holger Schwenk. 2010. N-gram-based Machine Translation enhanced with Neural Networks for the French-English BTEC-IWSLT'10 task. In *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 45–52.

Richard Zens, Franz Josef Och, and Hermann Ney. 2002. Phrase-based statistical machine translation. In *KI '02: Proceedings of the 25th Annual German Conference on AI*, pages 18–32, London, UK. Springer-Verlag.