Computational Linguistics in the Netherlands Journal 3 (2013) 217-233 Submitted 05/2013; Published 12/2013

Neural network language models to select the best translation

Maxim Khalilov MAXIM@TAUSLABS.COM

TAUS Labs,
1011KW, Amsterdam, The Netherlands

José A.R. Fonollosa ADRIAN@GPS.TSC.UPC.EDU

Centre de Recerca TALP,
Universitat Politècnica de Catalunya
08034, Barcelona, Spain

Francisco Zamora-Martínez FRANCISCO.ZAMORA@UCH.CEU.ES

Dep. de Ciencias Físicas, Matemáticas y de la Computación
Universidad CEU-Cardenal Herrera
46115 Alfara del Patriarca (Valencia), Spain

Maria José Castro-Bleda MCASTRO@DSIC.UPV.ES
Salvador España-Boquera SESPANA@DSIC.UPV.ES

Dep. de Sistemas Informáticos y Computación
Universitat Politècnica de València
46022 Valencia, Spain

Abstract

The quality of translations produced by statistical machine translation (SMT) systems crucially depends on the generalization ability provided by the statistical models involved in the process. While most modern SMT systems use n-gram models to predict the next element in a sequence of tokens, our system uses a continuous space language model (LM) based on neural networks (NN). In contrast to works in which the NN LM is only used to estimate the probabilities of shortlist words (Schwenk 2010), we calculate the posterior probabilities of out-of-shortlist words using an additional neuron and unigram probabilities. Experimental results on a small Italian-to-English and a large Arabic-to-English translation task, which take into account different word history lengths (n-gram order), show that the NN LMs are scalable to small and large data and can improve an n-gram-based SMT system. For the most part, this approach aims to improve translation quality for tasks that lack translation data, but we also demonstrate its scalability to large-vocabulary tasks.

1. Introduction

Translating from one natural language into another is one of the most complex higher-order activities of the human brain. Machine translation (MT) technology is a field of computational linguistics investigating and modeling the translation of texts, while statistical machine translation (SMT), in contrast to many automatic rule-based translation systems, is a translation paradigm based on statistical learning techniques.

Language modeling is an important part of any MT system, but it has not received much specialized attention within the SMT community. Instead, the research has been focused on more

specialized translation models, decoding algorithms, and training techniques. In contrast, in other fields of natural language processing, particularly in automatic speech recognition or understanding, one can find a large body of research that addresses the specific problems of language modeling. To a great extent, this discrepancy is a consequence of noisy experimental results and inconsistencies between a language model (LM) configuration and a translation system performance. However, the increase of available training data has made the application of such monolingual techniques quite promising since, typically, the greater amount of data used to estimate the parameters of the LM, the better the LM performance.

Regardless of its internal configuration, an SMT system typically takes as its basis a log-linear combination approach in which the target language sentence is defined by a combination of feature functions. This set normally includes a target-side LM, which informs a translation decoder about the correctness of a given sentence and the fluency of the translation hypothesis.

In this paper we follow the continuous space LM approach, which is a coherent and natural evolution of probabilistic LMs. We show that it deals better with the smoothing challenge and thereby provides better generalizations to unknown n-grams and concentrate on the scalability problem crucial for this type of LMs.

While the use of a continuous space representation of a language has been successfully applied in recent neural network (NN) approaches to language modeling (Xu and Jelinek 2004, Bengio et al. 2003, Castro and Prat 2003, Arisoy et al. 2012), domain adaptation (Lavergne et al. 2011, Park et al. 2010) and speech recognition (Schwenk 2007), the neural network language model (NN LM) application in the state-of-the-art SMT systems is not so popular. The only works can be traced back to the studies done by Schwenk et al., in which the NN LM was applied both to train a target-side LM (Schwenk et al. 2006, Schwenk 2010) in the form of a fully-connected Multilayer Perceptron and to smooth the probabilities involved in the bilingual tuple translation model (Schwenk et al. 2007).

The NN LM described in this paper follows a similar approach, but differs in the way how the probabilities for out-of-shortlist words are estimated. More details can be found in Section 3.3.

The inability of modern SMT systems to accommodate an increased workload caused by NN LMs, opens the way for new less computationally expensive mechanisms *supporting* the translation process. In contrast to some previously described techniques that improve the performance of a SMT system by incorporating the NN LM when only a small amount of training material is available, we describe two alternative scenarios. First, we experiment on a small-vocabulary Italian-to-English translation task demonstrating the NN LM potential for MT. However, a more interesting alternative field to apply our approach is to address the scalability problem which is especially crucial when an NN LM is used in tasks with large amount of training data. Thus, at the second stage, we provide translation results for a real-world large-vocabulary Arabic-to-English task and demonstrate an improvement in terms of final translation quality achieved by circumvention of difficulties imposed by complex structure of natural languages.

The rest of the paper is structured as follows: Section 2 briefly outlines the n-gram-based SMT system. In Section 3 the novel feature presented in the paper is described, i.e. NN LMs and its training algorithm. Section 4 presents our experimental setup and obtained results, while Section 5 concludes the article with the leading discussions.

2. UPC n-gram-based SMT system

Most modern SMT systems follow phrase-based (Koehn et al. 2007) or hierarchical (Chiang 2007) translation approaches. In this study, we follow an alternative algorithm, which is the n-gram-based or tuple-based SMT (Mariño et al. 2006) that has proved to be competitive with the state-of-the-art systems in recent evaluation campaigns (Khalilov et al. 2008, Lambert et al. 2007).

An n-gram-based SMT deals with bilingual n-grams, which are the so-called tuples. Tuples are extracted from a word-to-word alignment (performed with GIZA++[1] (Och and Ney 2000)) according to certain constraints (Mariño et al. 2006) and are composed of one or more words from the source language and zero or more words from the target one.

Hence, tuples induce a unique segmentation of the pair of sentences. By contrast, phrase-based systems produce all the possible pair of phrases that are consistent, leading to a several number of segmentation possibilities given a pair of sentences.

While regular phrase-based SMT considers context only for phrase reordering but not for translation, the N-gram-based approach conditions translation decisions on previous translation decisions. The context used in this way is bilingual and a translation model can be seen here as an LM, where the language is composed of tuples.

Because of the unique segmentation of sentence pairs, in case of n-gram-based SMT, the translation procedure is regarded as a stochastic process maximizing the joint probability $p(f, e)$, which is approximated at the sentence level. Besides the n-gram translation model, the feature models taken into consideration are: (1) a target LM of words (trained with the SRI language modeling toolkit[2] (Stolcke 2002)); (2) a word bonus model (to penalize the target sentence length) and (3) two-directional lexicon models.

2.1 Decoding and optimization

The MARIE decoder[3] (Crego et al. 2005) with extended monotone distortion was used as a search engine for the translation system. The decoder implements a beam-search algorithm with pruning capabilities. The feature functions described above were taken into account in the decoding process. Given the development set and a set of reference translations, the log-linear combination of weights can be adjusted using the simplex optimization method (Nelder and Mead 1965) to maximize the score function according to a combination of automatic evaluation metrics.

2.2 Extended word reordering

The n-gram-based translation system is highly sensitive to the difference in word order between source and target languages. An extended monotone distortion model based on automatically extracted reordering patterns was used in our experiments. Reordering patterns are extracted in the training stage from the crossed links found in the word alignment; in the next step, the monotone search graph is extended with re-orderings following the patterns found in the training set (Crego and Mariño 2007). Once the search lattice is built, the decoder traverses the graph looking for the best translation.

2.3 Rescoring

An NN LM model is integrated in the n-gram-based SMT system within a discriminative rescoring/reranking framework (composed of two steps), which incorporates complex feature functions by using the entire translation hypothesis to generate a score. During the first step, the MARIE decoder produces a list of M candidate translations based on the vector of weights trained over the m basic features (excluding orthodox n-gram LM in order not to diminish the NN LM effect). Then, the statistical scores of each generated translation candidate are rescored using information provided by the NN LM. This module presumably should add knowledge not included during decoding to better distinguish between higher and lower quality translations. During this step, a rescoring vector is trained over $m + 1$ features and provides different, better motivated choices for the single-best translation hypothesis.

1. code.google.com/p/giza-pp/
2. www.speech.sri.com/projects/srilm/
3. talp.upc.edu/talp/index.php/en/resources/tools/marie

An alternative way of incorporating NN LM into a SMT system is to use the continuous space LM directly during decoding. We decided not to pursue this strategy since this would result in a dramatic increase of decoding time.

2.4 Translation scores

We adopted three widely-used metrics for automatic evaluation of the MT quality:

- The *BLEU* score that accounts for the translation quality evaluation, by measuring the distance between a given translation and the set of reference translations using an n-gram LM (a 4-gram in the framework of this study) (Papineni et al. 2002).

- The *NIST* score is a sensitive metric of the translation quality, based on the BLEU score, but weighting n-grams in order to provide less informative n-grams with higher weights (Doddington 2002).

- The *METEOR* score is a metric for the evaluation of the MT output, which is calculated as an averaged mean of precision and benefited recall, considering stems and synonyms matching (more details can be found in Banerjee and Lavie (2005)).

3. Neural network language models

It is very likely to encounter new n-grams that were never witnessed during training due to the heavy tailed structure of any natural language. N-gram LMs are often criticized because they lack any explicit representation of dependencies longer than $n-1$ preceding tokens, while the effective range of dependency is significantly longer than this. We address the problem of LM smoothing in a continuous domain using a connectionist LM.

A major difference between classical n-gram LM and NN LM approaches lies in their distinct mechanism used to implement the smoothing process. In regular n-gram models, a "de facto" standard smoothing algorithm is modified Kneser-Ney discounting (Chen and Goodman 1999), which can be considered an extension of absolute discounting. This method takes into account that the lower-order model is only significant when the higher order count is small or zero (James 2000, Chen and Goodman 1999). In contrast to Kneser-Ney backing-off, interpolated smoothing models (for instance, Jelinek-Mercer or interpolated Chen-Goodman models (Chen and Goodman 1999)) do use the information from lower-order models when determining the probability of n-grams with non-zero counts.

Within an NN LM, posterior probabilities are interpolated for any possible context of length $n-1$ rather than backing-off to shorter contexts. Unfortunately, this generality involves a greater computational cost of evaluating and training an NN LM that linearly depends on the number of weights. This number is dominated by the size of the last hidden layer multiplied by the vocabulary size. The growing number of required calculations quickly overwhelms modern computational resources and makes the implementation computationally intractable for even average-sized vocabulary tasks.

However, Zipf's law (Zipf 1949) states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table. Consequently, the most frequent word will occur approximately twice as often as the second most frequent word, which occurs twice as often as the fourth most frequent word, and so on. This observation explains why the input and output of the NN can be, in practice, limited to a shortlist of K most frequent words from the vocabulary. The choice of the shortlist and the n-gram order is a trade-off between the NN LM training time and the SMT system performance.

3.1 Model architecture

an NN LM is a statistical LM that follows the n-grams assumption to estimate the LM probability for a sequence of words of length $|W|$:

$$p(w_1 \dots w_{|W|}) \approx \prod_{i=1}^{|W|} p(w_i | w_{i-n+1} \dots w_{i-1}) \tag{1}$$

but where the probabilities that appear in the former expression are estimated within an NN. The model naturally fits under the probabilistic interpretation of the outputs of NNs: when an NN is trained as a classifier, the outputs associated to each class are estimations of the posterior probabilities of the defined classes (Bishop 1995, Bengio et al. 2003).

The training set for an LM is a sequence $w_1 w_2 \dots w_{|W|}$ of words from a vocabulary Ω. In order to train an 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[4]. The problem related to this encoding for the tasks with large vocabularies (as is often the case) is the huge size of the resulting NN. We address this problem following Bengio et al. (2003), by developing a *distributed representation* for each word.[5] an NN LM is able to learn jointly the distributed representation of each word into a continuous space and the conditional probability estimates of Equation (1). This procedure allows the NN LM to smooth to unseen sequences of words.

Figure 1 illustrates the architecture of the feed-forward NN used to estimate the NN LM:

- The input is composed of words $w_{i-n+1}, \dots, w_{i-1}$ from Equation (1). For example, the input words are w_{i-3}, w_{i-2}, and w_{i-1} for a 4-gram. Each word is represented using a local encoding at each L_j set of neurons.
- P is the projection layer of the input words, formed by $P_1, \dots, P_{n-1}$ subsets of projection units. The subset of projection units P_j represents the distributed encoding of input neurons at L_j (corresponding to the word at input position j). The weights of this projection layer are shared, that is, the weights from each local encoding of input L_j to the corresponding subset of projection units P_j are the same for all input words. After training, the codification layer is removed from the network by pre-computing a table of size $|\Omega|$ which serves as a distributed encoding. The codification of each word is computed as follows:

  $$P_j = L_j^T \cdot W_{l,p} + b \tag{2}$$

  where L_j^T is the transposed of the vector L_j, representing the local codification of the corresponding word $w_{i-n+1+j}$, $W_{l,p}$ is the matrix of NN weights from each input word to the corresponding projection units subset (shared for each input word), b is the vector of biases of each projection units subset (shared for each subset), P_j is a vector that represents the distributed encoding of the corresponding word.
- H denotes the hidden layer, which computes:

  $$H = tanh(P^T \cdot W_{p,h} + c) \tag{3}$$

  where P^T is the transposed of the projection layer vector (concatenation of $P_1, P_2, \dots, P_{n-1}$), $W_{p,h}$ is the matrix of NN weights from the projection layer to the hidden layer, c is the vector of the hidden layer biases, and $tanh(.)$ is the component-wise hyperbolic tangent activation function.

4. A word locally encoded needs a vector of $|\Omega|$ neurons, where exists a one-to-one mapping between neurons and words, so the neuron which represents the word is activated with 1, and the rest of neurons are 0s.
5. In a distributed representation, the words are mapped (or projected) into a continuous space, using a number of neurons much smaller than $|\Omega|$.

- The output layer O has $|\Omega|$ units, one for each word of the vocabulary. O is calculated as follows:

$$A = H^T \cdot W_{h,o} + d \quad (4)$$

$$O = \frac{exp(A)}{\sum_{j=1}^{|\Omega|} exp(a_j)} \quad (5)$$

where $W_{h,o}$ is the matrix of weights from hidden layer to output layer, d is the vector of output layer biases, A is the vector of activation values computed before applying the softmax normalization, and a_j is the component j of A.

The cross-entropy error function has been used for all the training experiments, adding a L2 regularization term to all weights, except the bias:

$$E = D \cdot \log(O) + \epsilon \sum_{w_i \in \mathbf{W}} \frac{w_i^2}{2} \quad (6)$$

where D is the vector of desired outputs, $\mathbf{W}$ is the union of weight matrixes $W_{l,p}, W_{p,h}, W_{h,o}$, and E is the computed error. The NN estimates the posterior probability of each word w_i of the vocabulary given its history, i.e., $p(w_i|w_{i-n+1} \dots w_{i-1})$.

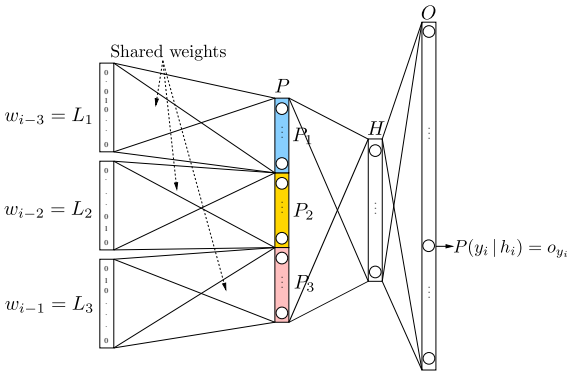

Figure 1: Architecture of the continuous space NN LM for 4-grams, where $h_i = w_{i-n+1} \dots w_{i-1}$.

To illustrate the huge size of the NNs, we computed the number of weights of the largest NN LM, which contained 224 neurons in the projection layer and 200 neurons in the hidden layer. In the case of the Italian-to-English translation, the NN that includes all vocabulary words that appear more than twice has 914,905 weights. For the Arabic-to-English task, the NN that has 56 neurons in the projection layer operates with 2,266,803 weights. Nevertheless, a standard 4-gram trained on the same corpora has 20,817,600 parameters.

3.2 Continuous space LM experiments

The strategy for the NN LM experiments was to use the Italian-to-English BTEC corpus in preliminary experiments, while a larger amount of effort has been dedicated to the Arabic-to-English NIST

task. The underlying idea was to prove that NN LMs can be used in large-scale MT trials focusing on the issues of practical usefulness of the continuous space LMs.

We considered two key parameters for the continuous space NN LMs:

- A *shortlist size* is defined by the K most frequent words, or by a word frequency threshold Θ that implies that every word occurring less than Θ times in the training corpus is discarded.
- An *n-gram order* limits a word history to the n preceding words. N-gram of orders 3 and 4 were considered in the Italian-to-English experiments; 4−, 5− and 6-gram configurations, as well as interpolation of high- and low-order n-grams, were tested when translating from Arabic.

The Θ values were set to 2, 3 and 4 for the Italian-to-English translation, which correspond to 4105, 3093, and 2498 words in the NN LM vocabulary, respectively. When translating from Arabic into English we took a different approach: we used the 10 K most frequent words from the full training vocabulary. We refer to Table 1 for actual vocabulary size values.

Italian-to-English

Θ	**# of words in English training set**
4	2,498
3	3,093
2	4,105

Arabic-to-English

Θ	**# of words in English training set**
N/A	10,000

Table 1: Number of words in the reduced training corpora. Note that the training set used for the Italian-to-English translation contains 10.2K unique words, and vocabulary of the Arabic-to-English is 157.6K words.

An additional neuron is added in the input and the output layers to take into account the out-of-shortlist words, called the OSL neuron. During training each out-of-shortlist word is replaced by the OSL identifier. In the testing step, when computation of the probability of an out-of-shortlist word is needed, the activation of the OSL neuron is combined with a simple standard unigram model computed over all the out-of-shortlist words. This probability is computed as follows:

$$p(w_i|w_{i-n+1}\dots w_{i-1}) = \begin{cases} O_{w_i} \text{ iff } w_i \text{ is in the shortlist;} \\ O_{OSL} \cdot p(w_i|OSL) \text{ iff otherwise,} \end{cases} \quad (7)$$

where O_{w_i} is the output neuron activation related to w_i, O_{OSL} is the OSL neuron activation, and $p(w_i|OSL)$ is the standard unigram probability computed over the out-of-shortlist words. Finally, each NN LM is the combination of four NNs, one for each size of the projection layer (see Section 3.1).

To define the $p(w_i|OSL)$ value we followed a trade-off between two alternative strategies presented and compared in Emami and Mangu (2007). The first approach implies a standard n-gram calculation, substituting $p(w_i|OSL)$ by $p(w_i|OSL, w_{i-n+1}\dots w_{i-1})$, in the same way as described in Schwenk (2007). According to the second strategy, the $p(w_i|OSL)$ value is set to 0. Since the obtained results were indistinguishable, we followed a compromise strategy, using a unigram probability to model $p(w_i|OSL)$.

When reestimating the weight coefficients for the new log-linear model with the NN LM, different starting points were tried, and the best set of weights resulted from the *100 BLEU + 4 NIST* criteria.

3.3 Differences between our NN LM and Schwenk's system

To the best of our knowledge, the study found in this paper is the first attempt to present an NN LM different from the work by Schwenk.

Armed with Zipf's law, we estimate the posterior probability of the K most frequent words from the vocabulary without significant loss of generality. The posterior probabilities of more rare words are estimated introducing an extra neuron and using a standard unigram model.

By contrast, the model previously presented in Schwenk et al. (2006) computes the posterior probabilities of out-of-shortlist words using a standard n-gram model. It requires the estimation of the contribution of all shortlist words in the standard n-gram model. Besides, when all the vocabulary is used at the NN LM input, the codification of less frequent words cannot be well learned (Hai-Son et al. 2010).

4. Experiments

4.1 Data

The experiment results were obtained using two corpora, which are different in size of the training material (see brief statistics in Table 2). The first one is the Italian-to-English BTEC corpus (Takezawa et al. 2002), which is a collection of spoken dialogue data. The second corpus under consideration is the 37M-word extraction from the Arabic-to-English NIST[6] corpus (news domain). In both experiments, the development and test sets include more than one ground truth reference (each reference is a different translation), which normally helps in the automatic evaluation process because the error is measured over the best of these references. Note that statistics are shown for source (Italian and Arabic) portions of the development and test bilingual corpora.

Set	Language	#Sentences	#Words	Vocab. Size	Average Sent. Length	References
Italian-to-English BTEC corpus						
Train	Italian	24.5K	166.3K	10.2K	6.5	-
Train	English	24.5K	155.4K	7.3K	6.1	-
Dev	Italian	489	5.2K	1.2K	6.5	7
Test	Italian	500	6.0K	1.3K	6.9	7
Arabic-to-English NIST corpus (1.2M-line extraction)						
Train	Arabic	1.2M	37.4M	186.9K	31.2	-
Train	English	1.2M	37.4M	157.6K	31.2	-
Dev	Arabic	2,075	62.7K	10.1K	30.2	4
Test	Arabic	2,040	61.6K	9.9K	30.2	4

Table 2: Basic statistics of training, development and test data.

4.2 Data preprocessing

For the Italian-to-English system, the preprocessing step consisted of tokenization, tagging, lemmatization, and separation of contractions for the Italian part, as described in Crego et al. (2006).

Regarding Arabic-to-English translation, we used a similar approach to that shown in Habash and Sadat (2006). The MADA+TOKAN[7] (Roth et al. 2008) system was utilized for disambiguation and tokenization. For disambiguation only diacritic unigram statistics were employed. For tokenization

6. National Institute of Standards and Technology
7. `www1.ccls.columbia.edu/~cadim/MADA.html`

we used the D3 scheme with the `-TAGBIES` option that splits the following set of clitics: w+, f+, b+, k+, l+, Al+ and pronominal clitics. The `-TAGBIES` option produces Bies POS tags on all taggable tokens.

In the first step of the N-best list extraction, the parameter N was set to 1000, limiting the size of the list of possible translations generated from the MARIE decoder output lattice.

4.3 System configurations

To provide a reasonable comparison with the NN LM experiments, we considered regular n-gram LMs for rescoring in the same way as the NN LMs (integrating the n-gram LM in the rescoring step). We call this system configuration *Baseline 1*.

As an alternative configuration, we considered the inclusion of the regular LM as a feature in the set of functions combined in a log-linear way during decoding (we call this system *Dec*). Results shown by the *Dec* system correspond to the performance of a standard n-gram-based SMT.

As a secondary baseline (*Baseline 2*) we used the continuous space system with default parameters, as described in Schwenk (2010). The package we used is available online: `http://www-lium.univ-lemans.fr/fr/content/cslm`.

In every NN LM experiment, the N-best list size was set to 1000.

Modified Kneser-Ney discounting was chosen to compute smoothed n-gram LMs since it had demonstrated the best results in terms of perplexity and the final translation score (BLEU) measured on the concatenation of the reference translations (development dataset). We compared the original Kneser-Ney discounting with the Good-Turing and Chen-Goodman (uninterpolated and interpolated versions) discounting algorithms (Chen and Goodman 1999). Application of the modified Kneser-Ney technique demonstrated significant improvement in perplexity ($\approx 12\%$) and translation quality according to the BLEU score ($\approx 3.9\%$) in comparison with alternative smoothing algorithms.

In order to achieve a good generalization and performance, we trained four NNs that are linearly combined to build the final NN LM. Each NN had a different projection layer size (128, 160, 192, 224 for the Italian-to-English task, and 160, 192, 224, 256 for the Arabic-to-English task) and the same hidden layer size (set to 200 neurons).

The number of parameters of each NN LM depends on the size of the layer. These numbers are taken from preliminary works done with these two tasks, and based in the literature. The combination of different projection layers seems to be better than optimizing the size of this layer (which is very time consuming). The hidden layer does not have an important effect on the performance of the model, but has a big computational impact (Schwenk 2007).

Automatic evaluation conditions were case-sensitive and included punctuation marks. All the automatic scores are calculated for 7 (BTEC experiments) and 4 (NIST experiments) reference translations.

4.3.1 Italian-to-English experiments

Table 3 shows BLEU, NIST, and METEOR scores for the systems with 3- and 4-gram NN LMs being an integrated component of a combined SMT system with reduced amount of training material. Both *Baseline 1* and *Dec* systems were trained and tested using 4-gram target-side LMs. Best scores are placed in cells filled with grey.

4.3.2 Arabic-to-English experiments

In the Arabic-to-English experiments we could use higher-order NN LMs than in the Italian-to-English task due to the larger amount of training data. Different baseline systems have been considered:

- *Baseline 1 (4), (5), and (6)* are the systems that employ regular LMs of corresponding order in the rescoring step,

System		Dev			Test		
		BLEU	NIST	METEOR	BLEU	NIST	METEOR
Baseline 1		39.2	8.5	73.0	33.5	7.7	70.3
Dec		39.4	8.6	73.2	33.6	7.7	70.9
NN LMs							
Baseline 2		42.2	8.9	74.0	34.9	7.9	71.2
NN LM $\Theta > 4$	3-gram	41.7	8.8	73.4	34.3	7.8	70.2
	4-gram	40.9	8.6	73.0	34.4	7.8	70.2
NN LM $\Theta > 3$	3-gram	41.6	8.7	73.1	34.4	7.8	70.7
	4-gram	42.0	8.8	73.8	34.4	7.8	70.4
NN LM $\Theta > 2$	3-gram	41.8	8.8	73.5	34.3	7.8	70.4
	4-gram	42.3	8.8	74.0	34.9	7.8	71.2

Table 3: Evaluation scores on the development and test datasets for the Italian-to-English BTEC translation.

System	Dev			Test		
	BLEU	NIST	METEOR	BLEU	NIST	METEOR
Baseline 1 (4)	46.1	10.1	64.5	38.1	9.6	60.3
Baseline 1 (5)	45.7	10.1	64.3	38.1	9.6	60.4
Baseline 1 (6)	45.3	10.0	64.7	37.9	9.6	60.5
Baseline 1 (4+5+6)	45.0	9.9	64.1	37.9	9.6	60.6
Dec (6)	45.1	9.9	64.1	37.9	9.6	60.5
NN LMs						
Baseline 2	46.6	10.2	64.5	38.6	9.8	60.6
4-gram	46.5	10.2	64.6	38.4	9.7	60.6
5-gram	46.5	10.2	64.4	38.6	9.7	60.3
6-gram	46.6	10.2	64.4	38.5	9.7	60.3
$4+5+6$-grams	46.6	10.2	64.3	38.5	9.7	60.5
NN LMs excluding SRI LM models						
4-gram	46.3	10.1	64.2	38.0	9.6	60.5
5-gram	46.5	10.3	64.2	38.3	9.6	60.4
6-gram	46.4	10.2	64.5	38.4	9.6	60.4
$4+5+6$-grams	46.5	10.3	64.5	38.2	9.6	60.6

Table 4: Evaluation scores on the development and test datasets for the Arabic-to-English NIST translation.

- *Baseline 1 (4+5+6)* combines 4-, 5-, and 6-grams in one N-best list,
- *Dec (6)* provides the decoder with access to the 6-gram standard LM without additional rescoring.

Along with independent NN LMs (*"4-, 5-, and 6-grams"*), we trained the network interpolating high- and low-order n-grams (*"4+5+6-grams"*). To isolate the impact of NN LMs to translation

quality we rescored the N-best lists excluding the scores generated by regular n-gram LMs (*"NN LMs excluding SRI LM models"*).

The best system configurations are highlighted in both aforementioned tables.

4.4 Perplexity analysis

The output sentences from a SMT system are built by aggregating word sequences that have a high-scoring combination of probabilities provided by the bilingual tuple translation model and a set of feature models, including LM. Therefore, there is no a clear correlation of the impact of the LM perplexity on the assembled translation. However, perplexity is a measure of a predictive power of an LM, which can be used to compare how well an LM can predict the next word in a previously unseen piece of text.

Table 5 shows perplexity values for stand-alone LMs measured on the merged set of translation references of the test corpora for the Italian-to-English BTEC and the Arabic-to-English NIST tasks.

Language Model	Perplexity
Italian-to-English BTEC task	
Conventional 3-gram	156
Conventional 4-gram	155
NN LM 3-gram, $\Theta > 4$	131
NN LM 4-gram, $\Theta > 4$	130
NN LM 3-gram, $\Theta > 3$	132
NN LM 4-gram, $\Theta > 3$	128
NN LM 3-gram, $\Theta > 2$	130
NN LM 4-gram, $\Theta > 2$	130
Arabic-to-English NIST task	
Conventional 4-gram	132
Conventional 5-gram	151
Conventional 6-gram	176
NN LM 4-gram	185
NN LM 5-gram	175
NN LM 6-gram	173
NN LM interpolation 4, 5, and 6-grams	167

Table 5: Perplexity results for different language models.

The architectural difference of target-side (English) NN LMs for the Italian-to-English and the Arabic-to-English translation tasks lies in the distinct algorithm of not-in-the-vocabulary (UNK) words processing.

For the Italian-to-English translation, the perplexity values calculated on the basis of SRI LM models are higher than the ones for the NN LMs. Note that the neural network computes the n-gram probabilities of a subset of the task vocabulary, computing out-of-shortlist words probabilities by the combination of the OSL neuron multiplied by a unigram model, which affects the perplexity of the model.

For the Arabic-to-English task, the output of the neural network covers the 10 K most frequent words only, while the task vocabulary is of 157.6 K size. The perplexity loss due to the out-of-shortlist words is more important in this task. It implies that the perplexity of the NN LMs calculated using this new neuron is higher than the perplexity of the SRI LM models.

4.5 Analysis of Italian-to-English results

The sentence-based BLEU scores[8] show that the share of sentences that improved when NN LMs were integrated into the SMT pipeline for the best performing integrated system (4-*gram* $\Theta > 2$) is 46%. At the same time the BLEU scores for 12% of the sentences became worse and for the remaining segments remained unchanged.

As it can be observed, a considerable improvement has been obtained using an NN LM for the Italian-to-English translation. For the development dataset, the BLEU score for the NN LM experiments is higher than the one for the baseline system for all NN LM systems. The best-performing 4-*gram* $\Theta > 2$ NN LM system allows a gain up to 1.3 BLEU points for the test set over the system that includes a conventional n-gram LM as a feature in the decoder (*DEC*); and a gain of about 1.4 BLEU points for the test dataset over the system that uses the regular LM for rescoring (*Baseline*).

All the aforementioned differences are statistically significant for a 95% confidence interval and 1,000 resamples using the bootstrap resampling method (Koehn 2004). The upper-bound statistical significance threshold (BLEU score calculated on the test dataset translated with the *Baseline* system) lies at 34.0 BLEU points.

Analysis of NIST scores for the Italian-to-English systems shows that the baseline results for the test set are exceeded by all the NN LM systems. Concerning the METEOR score, only the 4-*gram,* $\Theta > 4$ system provides a better LM generalization. The performance shown by our best system is statistically indistinguishable from the results shown by Schwenk's system for BLEU and METEOR scores.

Correlation of automatic and subjective human evaluation metrics (fluency and adequacy) is one of the main topics in the area of MT evaluation. As was reported in Paul (2006) for small BTEC translation tasks, fluency correlates best with BLEU, while adequacy correlates best with METEOR. The NIST metric has a moderate correlation with both subjective human evaluation metrics. Taking the aforementioned observations into consideration, our work demonstrates the potential for the application of NN LMs to SMT systems to improve translation fluency, while adequacy remains the same. The positive impact of higher-order n-grams is not clear, and this is possibly due to the relatively short sentences provided within the BTEC corpus. Another possible issue is that higher-order n-gram order only slightly decreases translation quality, yet at the same time, it introduces noisier translations.

An example of a typical sentence from the Italian-to-English BTEC corpus is shown in Figure 2.

4.6 Analysis of Arabic-to-English results

For the best-performing NN LM system (5-gram NN LMs) in terms of BLEU, 34% of sentences were improved in comparison with the *Baseline 2* system, 10% decreased their performance and for the rest of the dataset no changes were observed.

For the Arabic-to-English translation, both BLEU and NIST scores calculated on the development dataset are improved when the NN LM is applied in comparison with the performance shown by baseline engines, while the METEOR values generated by the NN LM configurations vary around the scores produced by the system integrated with a conventional n-gram LM.

Considering the test data translation scores, the difference between BLEU scores shown by the best NN LM system and the best system from the baseline population is 0.5 BLEU points, that is above the statistical significance threshold for this task (± 0.5). At the same time, results achieved when rescoring N-best lists including NN LMs, but excluding standard n-gram LMs, are not statistically distinguishable neither from baselines, nor from NN LM systems.

The system configuration providing the better BLEU score corresponds to the 5-gram LMs. Incorporating NN LMs into this n-gram-based SMT system allows gaining up to 0.7 BLEU points for

8. In general settings, sentence-based BLEU scores do not make much sense due to the accumulative nature of BLEU.

Source	Oggi abbiamo a scelta insalata ai frutti di mare insalata di patate e insalata mista.
Refs	Today we have a choice of seafood salad potato salad and wild vegetables salad. We are serving seafood salad potato salad and wild vegetables salad today. As for today's salad you can enjoy seafood potato and wild vegetables. For salad we have seafood potato and wild vegetables today. Today's selections are the seafood salad potato salad and wild vegetables salad. For today we have the seafood salad potato salad and wild vegetables salad. For today you can choose to have the seafood salad the potato salad or the wild vegetables salad.
Baseline	Today *we have selection at* the seafood salad potato salad and mixed salad.
3-**gram** $\Theta = 5$	Today *we have to choose from* the seafood salad potato salad and mixed salad.
4-**gram** $\Theta = 5$	Today we have selection at the seafood salad potato salad and mixed salad.
3-**gram** $\Theta = 3$	Today *we have to choose from* the seafood salad potato salad and mixed salad.
4-**gram** $\Theta = 3$	Today *we have to choose from* the seafood salad potato salad and mixed salad.

Figure 2: Example of the Italian-to-English translation. The Italian expression *"Oggi abbiamo a scelta"* is translated by the baseline system as *"Today we have selection at"*, whereas three of four NN LM systems provide a more fluent translation *"Today we have to choose from"*.

Source	w AElnt wkAlp AlAnbA' AlAmArAtyp En wSwl AlEAhl AlArdny mn dwn twDyH brnAmj Aw mdp zyArp h .
Refs	The emirates news agency announced the arrival of the jordanian monarch without specifying either the programme or the duration of his visit . The emirates news agency announced the arrival of the jordanian king , without giving the details of his program , or the duration of his visit . The emirates news agency announced the jordanian monarch 's arrival withoutnoting his schedule or the duration of his stay . The emirates news agency announced the jordanian monarch 's visit without giving further details about its purpose or duration .
Baseline 6-**gram**	The emirates agency announced king access without clarifying programme or duration visit .
Dec	The emirates news agency said *the jordanian monarch* access without clarifying programme or the duration of his visit .
4-**gram**	The news agency announced *the jordanian monarch* access without clarification of the programme or the duration of the visit .
5-**gram**	The news agency announced *the arrival* of *the jordanian monarch* without clarification of the programme or the duration of the visit .
6-**gram**	The news agency announced *the jordanian monarch* access without clarifying programme or the duration of the visit .
$4 + 5 + 6$	The news agency announced *the arrival* of *the jordanian monarch* without clarification of the programme or the duration of the visit .

Figure 3: Example of the Arabic-to-English translation. All NN LMs (along with *Dec* system) manage to generate the correct translation "the jordanian monarch". Only 5- and 4+5+6-gram models can produce the correct translation of the Arabic word *"wSwl"*, which is *"the arrival"*. Other systems translate it incorrectly as *"access"*.

the test set over *Dec* and around 0.5 BLEU points over the best baseline configurations. Increase of n-gram order to 6 does not lead to further performance improvement, neither does the interpolation of 4, 5, and 6-grams.

Obtained results show that adding both NN LMs and regular NN LMs within a discriminative rescoring framework provides the deliverance of slightly improved but consistent translation quality in comparison with the systems that do not consider standard LMs.

For this task, the Schwenk system performs slightly better than our NN LMs in terms of BLEU and NIST scores and it is as good as the latter considering METEOR.

Figure 3 illustrates an example[9] of one of the sentences from the Arabic-to-English NIST.

5. Discussion and conclusions

The architecture of a SMT system implies that, the smaller the amount of available training data is, the worse is the performance of a translation system.

In this paper, we have shown the following:

1. The robustness of the NN LMs, even for a highly limited training corpus. The in-domain NN LM provides a significantly better generalization of the target language, better smoothed SMT output, and enhanced improvement in the automatically evaluated translation scores. The NN LM turns out to be beneficial even if it is trained on an excerpt of most frequent words from the vocabulary. We also claim that for small translation tasks, integration of NN LM improves translation fluency, while adequacy remains the same. The empirical proof of this claim is planned to be done in the near future.

2. A proof of the claim that an NN LM approach is scalable even at the modern level of technology development. We have demonstrated that the technique of using the NN LM only for the set of the K most frequent words, while the probabilities of the less frequent words are estimated with the use of an extra neuron and unigram probabilities, leads to minor improvements in translation quality for large-vocabulary tasks.

Comparison with existing systems based on continuous space language model shows that our approach performs practically as good as the known Schwenk's system in terms of BLEU (correlated with fluency, at least for small translation tasks) and slightly better considering METEOR (correlated with adequacy) (Paul 2006).

A main disadvantage of the continuous space LM is its very high computational cost during training. While traditional n-gram LMs can be trained in a few minutes using the SRI LM toolkit, it can take some days to estimate a continuous space LM for a large-vocabulary task. A possible solution to this problem can be either the application of fast-training techniques (lattice regrouping and the utilization of specialized NN libraries with an ability of parallel calculation) or involving powerful (and expensive) computing resources. These high computational costs cause that all research efforts have been done in a decoupled system, based on N-best rescoring. Recent research has yielded some promising results efficiently integrating the NN LM in the decoder (Zamora-Martínez et al. 2009, Zamora-Martínez et al. 2010).

The three most urgent tasks we are planning to undertake to increase the credibility of the NN LM integrated into an SMT framework are:

1. To experiment with a higher amount of training data, probably focusing on more distant language pairs.

2. To run a human evaluation campaign, based on adequacy/fluency scoring, to confirm the results of automatic evaluation.

9. The Arabic example is provided in Buckwalter transliteration (Buckwalter 1994).

3. To check the applicability of the described mechanisms for hierarchical SMT systems, like the one described in Chiang (2005) and Chiang (2007).

References

Arisoy, E., T.N. Sainath, B. Kingsbury, and B. Ramabhadran (2012), Deep neural network language models, *Proceedings of NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, Montreal, Canada, pp. 20–28.

Banerjee, S. and A. Lavie (2005), METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72.

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

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

Buckwalter, T. (1994), Issues in Arabic orthography and morphology analysis, *Proceedings of COLING 2004*, Geneva, Switzerland, pp. 31–34.

Castro, M. J. and F. Prat (2003), New directions in connectionist language modeling, *Computational Methods in Neural Modeling*, Springer-Verlag.

Chen, S. F. and J. Goodman (1999), An empirical study of smoothing techniques for language modeling, *Computer Speech and Language* **4** (13), pp. 359–394.

Chiang, D. (2005), A hierarchical phrase-based model for statistical machine translation, *Proceedings of the Association for Computational Linguistics (ACL) 2005*, pp. 263–270.

Chiang, D. (2007), Hierarchical phrase-based translation, *Computational Linguistics* **2** (33), pp. 201–228.

Crego, J. M., A. de Gispert, P. Lambert, M. Khalilov, M. Costa-jussà, J. B. Mariño, R. Banchs, and J. A. R. Fonollosa (2006), The TALP Ngram-based SMT System for IWSLT 2006, *Proceedings of IWSLT 2006*, pp. 116–122.

Crego, J. M. and J. B. Mariño (2007), Improving statistical MT by coupling reordering and decoding, *Machine Translation* **20(3)**, pp. 199–215.

Crego, J. M., J. B. Mariño, and A. de Gispert (2005), An Ngram-based statistical machine translation decoder, *Proceedings of INTERSPEECH05.*

Doddington, G. (2002), Automatic evaluation of machine translation quality using n-grams co-occurrence statistics, *In HLT 2002 (Second Conference on Human Language Technology)*, pp. 128–132.

Emami, A. and L. Mangu (2007), Empirical study of neural network language models for Arabic speech recognition, *Proceedings of IEEE Automatic Speech Recognition and Understanding Workhsop (ASRU 2007)*, pp. 147–152.

Habash, N. and F. Sadat (2006), Arabic preprocessing schemes for statistical machine translation, *Proceedings of the Human Language Technology Conference of the NAACL*, pp. 49–52.

Hai-Son, L., A. Alluzen, G. Wisniewski, and F. Yvon (2010), Training continuous space language models: Some practical issues, *Proceedings of EMNLP*, pp. 778–788.

James, F. (2000), Modified Kneser-Ney smoothing of n-gram models, *Technical report*, Research Institute for Advanced Computer Science (RIACS).

Khalilov, M., A. H. Hernández, M. R. Costa-jussà, J. M. Crego, C. A. Henríquez, P. Lambert, J. A. R. Fonollosa, J. B. Mariño, and R. Banchs (2008), The TALP-UPC Ngram-based statistical machine translation system for ACL-WMT 2008, *Proceedings of the ACL 2008 Third Workshop on Statistical Machine Translation (WMT'08)*, pp. 127–131.

Koehn, Ph. (2004), Statistical significance tests for machine translation evaluation, *Proceedings of Empirical Methods in Natural Language Processing (EMNLP) 2004*, pp. 388–395.

Koehn, Ph., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst (2007), Moses: open-source toolkit for statistical machine translation, *Proceedings of the Association for Computational Linguistics (ACL) 2007*, pp. 177–180.

Lambert, P., M. R. Costa-jussà, J. M. Crego, M. Khalilov, J. Mariño, R. E. Banchs, J. A. R. Fonollosa, and H. Shwenk (2007), The TALP Ngram-based SMT system for IWSLT 2007, *Proceedings of the International Workshop on Spoken Language Translation (IWSLT07)*, pp. 169–174.

Lavergne, T., A. Allauzen, Hai-Son Le, and F. Yvon (2011), LIMSI's experiments in domain adaptation for IWSLT11, *Proceedings of IWSLT 2011*, San Francisco, CA, USA, pp. 62–67.

Mariño, J. B., R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-jussà (2006), N-gram based machine translation, *Computational Linguistics* **32** (4), pp. 527–549, ACL.

Nelder, J.A. and R. Mead (1965), A simplex method for function minimization, *The Computer organization* **7**, pp. 308–313.

Och, F. and H. Ney (2000), Improved statistical alignment models, *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 440–447.

Papineni, K., S. Roukos, T. Ward, and W. Zhu (2002), BLEU: a method for automatic evaluation of machine translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL) 2002*, pp. 311–318.

Park, J., X. Liu, M.J.F. Gales, and P.C. Woodland (2010), Improved neural network based language modelling and adaptation, *Proceedings of INTERSPEECH 2010*.

Paul, M. (2006), Overview of the IWSLT 2006 Evaluation Campaign, *Proceedings of IWSLT06*, pp. 1–15.

Roth, R., O. Rambow, N. Habash, M. Diab, and C. Rudin (2008), Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking, *Proceedings of Association for Computational Linguistics (ACL)*, Columbus, Ohio.

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

Schwenk, H. (2010), Continuous-space language models for statistical machine translation, *The Prague Bulletin of Mathematical Linguistics* (93), pp. 137–146.

Schwenk, H., M. R. Costa-jussà, and J. A. R. Fonollosa (2006), Continuous space language models for the IWSLT 2006 task, *Proceedings of IWSLT 2006*, pp. 166–173.

Schwenk, H., M. R. Costa-jussà, and J. A. R. Fonollosa (2007), Smooth bilingual translation, *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, Prague, Czech Republic, pp. 430–438.

Stolcke, A. (2002), SRILM: an extensible language modeling toolkit, *Proceedings of the Int. Conf. on Spoken Language Processing*, pp. 901–904.

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

Xu, P. and F. Jelinek (2004), Random forest in language modeling, *Proceedings of EMNLP 2004*, pp. 325–332.

Zamora-Martínez, F., M. J. Castro-Bleda, and H. Schwenk (2010), N-gram-based machine translation enhanced with neural networks for the French-English BTEC-IWSLT'10 task, *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pp. 45–52.

Zamora-Martínez, F., M.J. Castro-Bleda, and S. España-Boquera (2009), Fast evaluation of connectionist language models, *in* Cabestany, Joan, Francisco Sandoval, Alberto Prieto, and Juan M. Corchado, editors, *Proceedings of the 10th International Work-Conference on Artificial Neural Networks, IWANN 2009*, Vol. 5517 of *LNCS*, Springer, pp. 33–40.

Zipf, G. K. (1949), *Human Behavior and the Principle of Least-Effort*, Addison-Wesley.