PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 108 JUNE 2017 283–294

Neural Networks Classifier for Data Selection in Statistical Machine Translation

Álvaro Peris, Mara Chinea-Ríos, Francisco Casacuberta

Pattern Recognition and Human Language Technology Research Center, Universitat Politècnica de València

Abstract

Corpora are precious resources, as they allow for a proper estimation of statistical machine translation models. Data selection is a variant of the domain adaptation field, aimed to extract those sentences from an out-of-domain corpus that are the most useful to translate a different target domain. We address the data selection problem in statistical machine translation as a classification task. We present a new method, based on neural networks, able to deal with monolingual and bilingual corpora. Empirical results show that our data selection method provides slightly better translation quality, compared to a state-of-the-art method (cross-entropy), requiring substantially less data. Moreover, the results obtained are coherent across different language pairs, demonstrating the robustness of our proposal.

1. Introduction

The performance of a statistical machine translation (SMT) system is dependent on the quantity and quality of the available training data. Typically, SMT systems are trained with all available data, assuming that the more data used to train the system, the better. Nevertheless, it is critical that such data is related to the task at hand. Translation quality is negatively affected when there is a lack of domain-specific training data (Callison-Burch et al., 2007; Koehn, 2010). In addition, growing the amount of data available is only feasible to a certain extent. The aim of data selection (DS) is to properly select for training a subset of sentence pairs from a large sentence pool, so that the translation quality achieved in the target domain is improved.

DS techniques extract monolingual or bilingual data that are similar to the in-domain corpus based on some criteria, either at monolingual or bilingual level. Such

 Corresponding author: lvapeab@prhlt.upv.es
Cite as: Álvaro Peris, Mara Chinea-Ríos, Francisco Casacuberta. Neural Networks Classifier for Data Selection in Statistical Machine Translation. The Prague Bulletin of Mathematical Linguistics No. 108, 2017, pp. 283–294.
doi: 10.1515/pralin-2017-0027.

selection is incorporated into the training data. The similarity metric varies depending on each technique. Cross-entropy (CE) difference is a typical and well-established ranking function (Moore and Lewis, 2010; Axelrod et al., 2011; Mansour et al., 2011; Schwenk et al., 2012; Rousseau, 2013). CE-based methods train n-gram language models on the in-domain corpus to select similar sentences from the out-of-domain corpus according to their CE difference.

On the other hand, distributed representation of words have proliferated spectacularly during the last years in the research community. Neural networks provide powerful tools for processing text, achieving success in text classification (Kim, 2014), machine translation (Sutskever et al., 2014; Bahdanau et al., 2015) or domain adaptation (Joty et al., 2017). Related to the DS field, Duh et al. (2013) leveraged neural language models to perform DS, reporting substantial gains over conventional n-gram language models.

Recently, convolutional neural networks (CNN) (LeCun et al., 1998) have also been used in the domain adaptation field (Chen and Huang, 2016; Chen et al., 2016). In these works, the authors used a similar strategy to the one proposed in Section 3, but in a different domain adaptation case—close to a transductive learning scenario: they have no in-domain training corpus, only a large out-of-domain pool and small sets of translation instances. Their goal was to select from the out-of-domain corpus, the more suitable samples for translating their in-domain corpora.

This paper tackles DS by taking advantage of neural networks as sentence classifiers, with the ultimate goal of obtaining corpora subsets that improve translation quality. In order to make systems scalable, such subsets should be as reduced as possible. Therefore, our goal is twofold: we want to select sentences subsets with the least size possible that improve translation quality.

The main contributions of this paper are:

- We tackle the DS problem for SMT as a classification task, employing CNNs and bidirectional long short-term memory (BLSTM) networks.
- We conduct a wide experimentation, using monolingual and bilingual corpora. The results show that our method outperforms a state-of-the-art DS technique in terms of translation quality and selection sizes.
- We show that both CNNs and BLSTM networks provide a similar performance for the task at hand.
- In order to make results reproducible, we release the source code of our method. Corpora are also publicly available.

The paper is structured as follows. Section 2, presents our neural DS method. We introduce two architectures, for taking into account a monolingual or a bilingual corpus. Section 3 presents a semi-supervised algorithm for training our classifiers. Next, Section 4 describes the experimental framework, detailing and discussing the results obtained. are detailed and discussed. Finally, Section 5 concludes the work, tracing the future lines of research.

2. Data selection

The goal of DS methods consists in selecting a subset S of sentences from an out-of-domain pool of sentences G, based on an in-domain corpus I. The objective is to enhance the performance of a SMT system trained using this selection. Note that, the lesser the size of S is, the easier is to extend the original SMT system.Therefore, the selection S must represent a trade-off between size and translation improvement.

2.1. Data selection using cross-entropy

As mentioned in Section 1, a well-established DS method consists in scoring the sentences from the out-of-domain corpus (G) by their CE difference (Moore and Lewis, 2010). For selecting S, this technique relates the CE given by a language model trained on the in-domain corpus I, together with an out-of-domain language model, computing a score for a sentence $\mathbf{x}$:

$$c(\mathbf{x}) = H_I(\mathbf{x}) - H_G(\mathbf{x}) \tag{1}$$

where H_I and H_G are the in-domain and out-of-domain CE of sentence $\mathbf{x}$, respectively.

Note that this method is defined in terms of I, as defined by the original authors. Even though it would also be feasible to define this method in terms of S, such redefinition lies beyond the scope of this paper, since our purpose is only to use this method only for comparison purposes.

In Axelrod et al. (2011), the authors propose a extension to this monolingual CE method, so that it is able to deal with bilingual information. To this end, they sum the CE difference for each side of the corpus, both source and target. Let I_s and G_s be the in-domain source corpus and the out-of-domain source corpus respectively, and I_t and G_t be the in-domain and out-of-domain target corpora. Then, the CE difference between a source sentence $\mathbf{x}$ and a target sentence $\mathbf{y}$ is defined as:

$$c(\mathbf{x}, \mathbf{y}) = [H_{I_s}(\mathbf{x}) - H_{G_s}(\mathbf{x})] + [H_{I_t}(\mathbf{y}) - H_{G_t}(\mathbf{y})] \tag{2}$$

2.2. Data selection using neural networks

In this work, we tackle the DS problem as a classification task. Let us consider a classifier model M that assigns a probability $p_M(I \mid \mathbf{x})$ to a given sentence $\mathbf{x}$, depending whether $\mathbf{x}$ belongs to the in-domain corpus I or not.

In this case, to obtain the selection S, one could just apply M to each sentence from the out-of-domain pool G and select the most probable ones.

We propose to use a neural classifier, exploring CNN and BLSTM networks as sentence encoders. As shown in Fig. 1 (left), the input sentence is fed to our system following a one-hot codification scheme and is projected to a continuous space by means of a word-embedding matrix. Next, the sequence of word embeddings is processed either by a CNN or a BLSTM network. After this, we stack one or more fully-connected

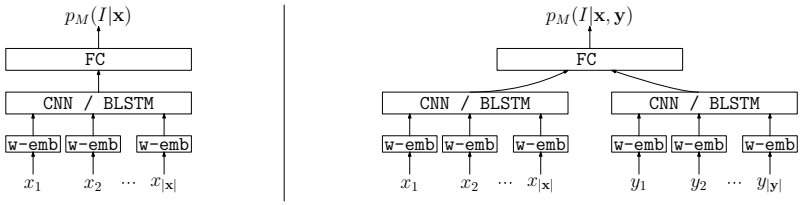

Figure 1: General architecture of the proposed classifiers. The monolingual model is shown at the left while the bilingual model is shown at the right. `w-emb` *stands for word-embedding and* `FC` *for fully-connected layer.*

(FC) layers. Finally, we can apply a softmax function, if we wish to obtain normalized probabilities. All elements can be jointly trained by maximum likelihood.

This reasoning can be extended in order to be applicable to a bilingual corpus. Therefore, if we have the source sentence $\mathbf{x}$ and its corresponding translation $\mathbf{y}$, we can model the probability $\mathrm{p_M(I \mid \mathbf{x}, \mathbf{y})}$. For doing this, we used two networks, one for the source language and another one for the target language. We concatenated their outputs and apply FC layers, as in the previous case, computing an unique score for each bilingual pair. Fig. 1 (right) shows this architecture.

Convolutional neural networks. CNNs have proven their representation capacity, not only in computer vision tasks (Szegedy et al., 2015), but also representing text (Kalchbrenner and Blunsom, 2013; Kim, 2014). In this work, we used the non-static CNN proposed by Kim (2014). This CNN consists in the application of a set of filters to windows of different length. These filters apply a non-linear function (e.g. ReLU). Next, a max-pooling operation is applied to the set of convolutional filters. As result, the CNN obtains a feature vector representing the input sentence.

Recurrent neural networks. In recurrent neural networks, connections form a directed cycle. This allows the network to maintain an internal state and be effective sequence modelers. Moreover, bidirectional networks (Schuster and Paliwal, 1997) have two independent recurrent layers, one processing the input sequence in a forward manner and other processing it a backward manner. Therefore, they allow to exploit the full context at each time-step. Gated units, such as LSTM (Hochreiter and Schmidhuber, 1997; Gers et al., 2000), mitigate the vanishing gradient problem and hence, they are able to properly model long sequences. BLSTM networks can be used for encoding a sentence by concatenating the last hidden state of the forward and backward LSTM layers. This provides a compact representation of the sentence, which accounts for relationships in both time directions.

3. Semi-supervised selection

Properly training these neural classifiers may be a challenging task, since the in-domain data is scarce. Hence, for training them, we follow a semi-supervised iterative protocol (Yarowsky, 1995).

Input: P_0 (positive samples),
N_0 (negative samples),
G_0 (out-of-domain corpus),
l (selection size),
r (training granularity)
Output: P_i (selection of size l)
begin
  $i = 0$
  while $|P_i| \leq l$ **do**
    $M_i \leftarrow$ `Train model on` $\{P_i \cup N_i\}$
    $S_i \leftarrow$ `Classify` G_i `with` M_i
    $P_{i+1} \leftarrow \{P_i \cup$ `get_top`$(S_i, r)\}$
    $N_{i+1} \leftarrow \{N_i \cup$ `get_bottom`$(S_i, r)\}$
    $G_{i+1} \leftarrow \{G_i -$`get_top`$(S_i, r) -$ `get_bottom`$(S_i, r)\}$
    $i++$
  end
  return P_i
end

Algorithm 1: Semi-supervised selection. The functions `get_top` and `get_bottom` select the top-r and the bottom-r scoring sentences from a scored set. The algorithm returns a selection consisting of l sentences.

Algorithm 1 shows this semi-supervised training procedure. Since the data selection is a binary classification problem, we need a set of positive and negative training samples. We start from an initial set of positive samples P_0 and a set of negative samples N_0. At each iteration $i \geq 0$, we train a model with the current sets of data (P_i, N_i). Next, we classify all sentences belonging to the out-of-domain pool (G_i). We extract a number r of top-scoring sentences and include them into the set of positive samples, producing a new set P_{i+1}. Analogously, the r bottom-scoring sentences are included into a new negative samples set N_{i+1}. Hence, at each iteration, we remove $2r$ samples from the out-of-domain set, producing the pool G_{i+1}. Then, a new iteration starts. This is repeated until the selection P_i reaches the desired size (l).

We set our in-domain corpus I as P_0. We randomly extract $|\ I\ |$ sentences from G for constructing N_0. The initial out-of-domain pool G_0 is defined as $\{G - N_0\}$.

4. Experiments in SMT

In this section, we empirically evaluate the DS strategy proposed in Section 2. We conducted experiments on different language pairs for evaluating whether the conclusions drawn from one single language pair hold in further scenarios.

4.1. Corpora

Two corpora are involved within the DS task: an out-of-domain corpus G and an in-domain corpus I. DS selects only a portion of the out-of-domain corpus, and leverages that subset together with the in-domain data to train a, hopefully improved, SMT system. We used the publicly available Europarl (Koehn, 2005) and EMEA (Tiedemann, 2009) corpora as out-of-domain and in-domain data, respectively. As in-domain test sets, we used the Medical-Test and Medical-Mert corpora, partitions established in the 2014 Workshop on Statistical Machine Translation[1]. We focused on the English (En), French (Fr) and German (De) language pairs, conducting experiments in all directions. Table 1 shows the corpora figures.

	EMEA		**Medical-Mert**		**Medical-Test**		**Europarl**	
	$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$
En	1.0M	98k	501	979	1.0k	1.8k	2.0M	157k
Fr		112k		1.0k		26.9k		215k
En	1.1M	99k	500	979	1.0k	1.9k	1.9M	153k
De		141k		874		1.7k		290k

Table 1: Corpora main figures. EMEA is the in-domain corpus, Medical-Test is the evaluation data and Medical-Mert is the development set. Europarl is the out-of-domain corpus. $|S|$ stands for number of sentences and $|V|$ for vocabulary size. M denotes millions of elements and k thousands.

4.2. Experimental setup

All neural models were initialized using word-embedding matrices from word2vec, obtained using the skip-gram model from Mikolov et al. (2013) and trained on part of Google News dataset in the case of English and on Wikipedia in the case of French and German. Word-embedding matrices were fine-tuned during the semi-supervised selection protocol. The size of the word-embeddings was 300.

Following Kim (2014), we used filter windows of lengths 3, 4, 5 with 100 features maps each for the CNN classifier. In order to have a similar number of parameters

[1] http://www.statmt.org/wmt14/medical-task/

than in the CNN (20 million approximately), we used 300 units in each LSTM layer. 2 FC layers of size 200 and 100 were introduced after the CNN and BLSTM (Section 2.2).

For training the CNN classifier, we used Adadelta (Zeiler, 2012) with its default parameters. The BLSTM network was trained with Adam (Kingma and Ba, 2014), with a learning rate of 10^{-4}. During training, we applied Gaussian noise to the weights ($\sigma = 0.01$). All neural models[2] were implemented using the Theano (Theano Development Team, 2016) and Keras libraries. The number of sentences selected at each iteration (r) was chosen trading off speed and granularity ($r = 50,000$).

All SMT experiments were carried out using the open-source phrase-based SMT toolkit Moses (Koehn et al., 2007). The language model used was a 5-gram, with modified Kneser-Ney smoothing (Kneser and Ney, 1995), built with the SRILM toolkit (Stolcke, 2002). The phrase table was generated by means of symmetrised word alignments obtained with GIZA++ (Och and Ney, 2003). The log-lineal combination weights were optimized using MERT (minimum error rate training) (Och, 2003). In order to minimize the random nature of MERT and purposing to provide robustness to the results, every result of this paper constitutes the average of 10 repetitions. In the tables, 95% confidence intervals of these repetitions are shown.

The final translation quality was evaluated by means of BLEU (Papineni et al., 2002). Nevertheless, since we are in a DS scope, the amount of data required for training each system also becomes a fundamental evaluation metric.

The SMT systems were trained using the selection provided by the proposed methods together with the in-domain corpus. We compared the selection methods with two baseline systems. The first one consists in training the SMT system only with in-domain data. We refer to this setup with the name of `bsln-emea`. The second baseline was obtained training with all available data (i.e., in-domain and out-of-domain). We will refer to this setup as `bsln-all`. In addition, we also included results of a purely random sentence selection without replacement.

4.3. Experimental results

Table 2 shows the best results obtained with our DS method using the two neural network architectures proposed (CNN and BLSTM) and the CE method for each language pair.

In En-Fr and En-De, Fr-En, translation quality using DS improves over `bsln-all`, but using significantly less data (20%, 23% and 26% of the total amount of out-of-domain data, respectively). In the case of De-En, translation quality results are similar, but also reducing the amount of data required: only a 23%. According to these results, we can state that our DS strategy is able to deliver similar quality than using all the data, but only with a rough quarter of the data.

[2]Source code available at `https://github.com/lvapeab/sentence-selectioNN`.

	En-Fr		Fr-En	
Strategy	BLEU	#Sentences	BLEU	#Sentences
`bsln-emea`	28.6 ± 0.2	1.0M	29.9 ± 0.2	1.0M
`bsln-all`	29.4 ± 0.1	1.0M+1.5M	32.4 ± 0.1	1.0M+1.5M
Random	29.4 ± 0.4	1.0M+500k	32.3 ± 0.3	1.0M+500k
CE	29.8 ± 0.1	1.0M+450k	31.8 ± 0.1	1.0M+600k
BLSTM	$\mathbf{29.9 \pm 0.3}$	1.0M+**300**k	32.3 ± 0.1	1.0M+500k
CNN	29.8 ± 0.2	1.0M+450k	$\mathbf{32.3 \pm 0.2}$	1.0M+**350**k
	De-En		En-De	
Strategy	BLEU	#Sentences	BLEU	#Sentences
`bsln-emea`	23.7 ± 0.2	1.0M	15.6 ± 0.1	1.0M
`bsln-all`	26.2 ± 0.3	1.0M+1.5M	16.6 ± 0.2	1.0M+1.5M
Random	25.5 ± 0.1	1.0M+600k	16.8 ± 0.1	1.0M+550k
CE	25.5 ± 0.3	1.0M+600k	16.8 ± 0.2	1.0M+500k
BLSTM	25.9 ± 0.1	1.0M+500k	$\mathbf{17.1 \pm 0.2}$	1.0M+**400**k
CNN	$\mathbf{25.9 \pm 0.1}$	1.0M+**400**k	16.9 ± 0.1	1.0M+350k

Table 2: Summary of best results obtained. Columns denote, from left to right: selection strategy, BLEU, number of sentences, given in terms of the in-domain corpus size, and $(+)$ *selected sentences.*

All proposed DS methods are mostly able to improve over random selection but in some cases differences are not significant. It should be noted that beating random is very hard, since all DS methods, including random, will eventually converge to the same point: adding all the data available. The key difference is the amount of data needed for achieving the same translation quality.

Results obtained in terms of BLEU with our DS method are slightly better than the ones obtained with CE difference. However, CE difference requires significantly more sentences to reach comparable translation quality.

Finally, CNN and BLSTM networks seem to perform similarly. Therefore, we conclude that both architectures are good options for this task.

Table 3 shows the best results obtained with our bilingual data selection method using both neural architectures proposed (Bili-CNN and Bili-BLSTM) and bilingual CE (Bili-CE) method for each language pair. Again, the DS selection techniques beat all baselines in terms of BLEU, requiring less data to train the SMT system.

Compared to the monolingual methods, our bilingual DS techniques provide similar results. Nevertheless, in all cases the bilingual methods are able perform better selections at the early stages of the process, as illustrated in Figure 2. As we steadily select more sentences, monolingual and bilingual methods eventually converge to

	En-Fr		Fr-En	
Strategy	BLEU	#Sentences	BLEU	#Sentences
Bili-CE	30.2 ± 0.2	1.0M+350k	32.5 ± 0.1	1.0M+450k
Bili-BLSTM	$\mathbf{30.2 \pm 0.2}$	1.0M+**300k**	32.3 ± 0.1	1.0M+450k
Bili-CNN	30.1 ± 0.3	1.0M+300k	$\mathbf{32.6 \pm 0.2}$	1.0M+**500k**
	De-En		En-De	
Strategy	BLEU	#Sentences	BLEU	#Sentences
Bili-CE	25.9 ± 0.2	1.0M+350k	17.0 ± 0.2	1.0M+500k
Bili-BLSTM	26.0 ± 0.1	1.0M+500k	$\mathbf{17.1 \pm 0.2}$	1.0M+**250k**
Bili-CNN	25.8 ± 0.1	1.0M+200k	17.0 ± 0.1	1.0M+350k

Table 3: Summary of bilingual results obtained. Columns denote, from left to right: selection strategy, BLEU, number of sentences, given in terms of the in-domain corpus size, and $(+)$ selected sentences.

similar results. We can see that adding sentences selected by means of DS techniques improves over the baselines from the very beginning. Selecting at a bilingual level is specially effective in small selections: while the monolingual method requires 150k sentences for beating the `bsln-all` baseline, the bilingual methods only require 50k. Here we show only the En-Fr language pair due to space restriction, but this behavior is consistent across all languages.

5. Conclusion and future work

We developed a DS method, based on sentence classification techniques. The uses CNNs or BLSTM networks for computing a sentence representation. We thoroughly evaluated it over four language pairs. Our method yielded better translation performance than the cross-entropy DS technique, requiring a minor amount of data. Additionally, we found that both CNN and BLSTM networks performed similarly, thus being both suitable sentence encoders.

At the light of the monolingual results, we expected higher gains of performance when considering the both sides of the corpora. It should be tested if a different combination strategy of the classifiers is able to exploit parallel corpora to their full. Moreover, we should also compare the performance of classical classifiers, such as support vector machines (SVM) or logistical regression. We also noted that the De-En language pair had a different behavior than other language pairs. We should study the DS process when applied to inflected languages.

In this work, we chose the initial set of negative samples (N_0) following a random criterion. In the future, we should investigate if a more informed technique (e.g. per-

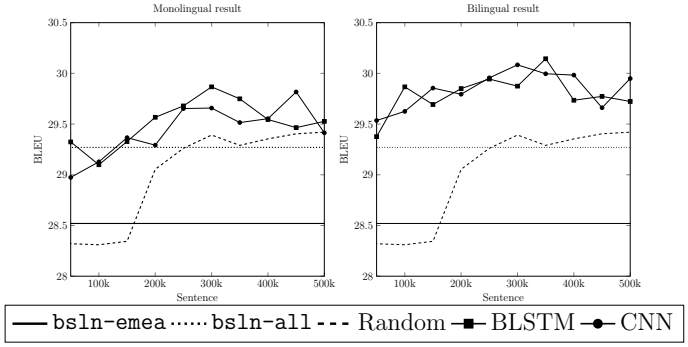

Figure 2: Effect of adding sentences over the BLEU score using the different DS techniques (with monolingual and bilingual form) and random selection techniques for the En-Fr language pair. Horizontal lines represent the scores when using just the in-domain training corpus(`bsln-emea`) and all the data available(`bsln-all`).

plexity or the invitation model from Hoang and Sima'an (2014)) helps the selection system by providing a more suitable N_0.

In addition, we aim to delve into the usage of semi-supervised training strategies for the classifier. Ladder networks (Rasmus et al., 2015) seem a promising tool. We should investigate how to include them in our pipeline. We should also explore one-shot learning strategies in a scenario where only the text to translate is available.

Finally, we should also test our data selection method within the neural machine translation (NMT) technology. NMT systems rely on the usage of large amount of data, but it should be investigated whether the inclusion of in-domain data effectively helps the system. Moreover, as by product of the NMT training, we could use the NMT encoder for pre-initializing our classifier, hoping a boost in the system performance.

Acknowledgements

The research leading to these results has received funding from the Generalitat Valenciana under grant PROMETEOII/2014/030 and the FPI (2014) grant by Universitat Politècnica de València. We also acknowledge NVIDIA for the donation of a GPU used in this work.

Bibliography

Axelrod, Amittai, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proc. of EMNLP*, pages 355–362, 2011.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473*, 2015.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. (Meta-) evaluation of machine translation. In *Proc. of WMT*, pages 136–158, 2007.

Chen, Boxing and Fei Huang. Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. *Proc. of CoNLL*, pages 314–324, 2016.

Chen, Boxing, Roland Kuhn, George Foster, Colin Cherry, and Fei Huang. Bilingual Methods for Adaptive Training Data Selection for Machine Translation. *Proc. of AMTA*, pages 93–103, 2016.

Duh, Kevin, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. Adaptation Data Selection using Neural Language Models: Experiments in Machine Translation. In *Proc. of ACL*, pages 678–683, 2013.

Gers, Felix A, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.

Hoang, Cuong and Khalil Sima'an. Latent Domain Translation Models in Mix-of-Domains Haystack, 2014.

Hochreiter, Sepp and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9 (8):1735–1780, 1997.

Joty, Shafiq, Nadir Durrani, Hassan Sajjad, and Ahmed Abdelali. Domain adaptation using neural network joint model. *Computer Speech & Language*, In Press, 2017.

Kalchbrenner, Nal and Phil Blunsom. Recurrent Continuous Translation Models. In *Proc. of EMNLP*, pages 1700–1709, 2013.

Kim, Yoon. Convolutional Neural Networks for Sentence Classification. In *Proc. of EMNLP*, pages 1746–1751, 2014.

Kingma, Diederik P. and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2014.

Kneser, Reinhard and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proc. of ICASSP*, pages 181–184, 1995.

Koehn, Philipp. Europarl: A parallel corpus for statistical machine translation. In *Proc. of MT Summit*, pages 79–86, 2005.

Koehn, Philipp. *Statistical machine translation*. Cambridge University Press, 2010.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL*, pages 177–180, 2007.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.

Mansour, Saab, Joern Wuebker, and Hermann Ney. Combining translation and language model scoring for domain-specific data filtering. In *Proc. of IWSLT*, pages 222–229, 2011.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS*, pages 3111–3119, 2013.

Moore, Robert C and William Lewis. Intelligent selection of language model training data. In *Proc. of ACL*, pages 220–224, 2010.

Och, Franz Josef. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pages 160–167, 2003.

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

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318, 2002.

Rasmus, Antti, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Proc. of NIPS*, pages 3546–3554, 2015.

Rousseau, Anthony. XenC: An Open-Source Tool for Data Selection in Natural Language Processing. *The Prague Bulletin of Mathematical Linguistics*, 100:73–82, 2013.

Schuster, Mike and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

Schwenk, Holger, Anthony Rousseau, and Mohammed Attik. Large, pruned or continuous space language models on a GPU for statistical machine translation. In *Proc. of NAACL-HLT*, pages 11–19, 2012.

Stolcke, Andreas. SRILM - an Extensible Language Modeling Toolkit. In *Proc. of ICSLP*, pages 901–904, 2002.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. In *Proc. of NIPS*, pages 3104–3112, 2014.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. of CVRP*, pages 1–9, 2015.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv:1605.02688*, 2016.

Tiedemann, Jörg. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In *Proc. of RANLP*, pages 237–248, 2009.

Yarowsky, David. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proc. of ACL*, pages 189–196, 1995.

Zeiler, Matthew D. ADADELTA: An Adaptive Learning Rate Method. *arXiv:1212.5701*, 2012.

Address for correspondence:
Álvaro Peris
lvapeab@prhlt.upv.es
Pattern Recognition and Human Language Technology Research Center,
Universitat Politècnica de València,
Camino de Vera s/n, 46022 Valencia, SPAIN.