Vector sentences representation for data selection in statistical machine translation

Mara Chinea-Rios[a,*], Germán Sanchis-Trilles[b], Francisco Casacuberta[a]

[a] *Pattern Recognition and Human Language Technology Research Center, Universitat Politècnica de València, Valencia, 46022, Spain*
[b] *Sciling, Carrer del Riu 321, Pinedo, Valencia, 46012, Spain*

Abstract

One of the most popular approaches to *machine translation* consists in formulating the problem as a pattern recognition approach. Under this perspective, bilingual corpora are precious resources, as they allow for a proper estimation of the underlying models. In this framework, selecting the best possible corpus is critical, and data selection aims to find the best subset of the bilingual sentences from an available pool of sentences such that the final translation quality is improved. In this paper, we present a new data selection technique that leverages a continuous vector-space representation of sentences. Experimental results report improvements compared not only with a system trained only with in-domain data, but also compared with a system trained on all the available data. Finally, we compared our proposal with other state-of-the-art data selection techniques (Cross-entropy selection and Infrequent ngrams recovery) in two different scenarios, obtaining very promising results with our proposal: our data selection strategy is able to yield results that are at least as good as the best-performing strategy for each scenario. The empirical results reported are coherent across different language pairs.

Keywords: statistical machine translation, data selection, continuous vector-space representation, cross-entropy, infrequent ngrams recovery

*Corresponding author
Email address: machirio@prhlt.upv.es (Mara Chinea-Rios)

1. Introduction

Machine Translation (MT) is a specific sub-field of Natural Language Processing (NLP). MT studies the way in which automatic systems should be developed so that they are able to translate a certain sentence in a source language into a sentence in a given target language, such that source and target sentences preserve the exact same meaning, while ensuring that the target sentence is well-formed in the corresponding target language.

Bilingual corpora are precious resources in computational linguistics and they constitute the possibility of performing another kind of machine translation: this is the case of Statistical Machine Translation (SMT), which advanced the state of the art in MT radically. The goal is to create mathematical models that can describe the translation process accurately, by estimating mathematical models that leverage bilingual training data.

The performance of an SMT system is dependent on the quantity and quality of the available training data. Typically, SMT systems are trained with all the available data, assuming that the more data used to train the system, the better. This assumption is backed by evidence that scaling to ever larger data shows continued improvements in quality, even when one trains models over billions of [1]. In the SMT context, n-grams refers to sequences of n consecutive words. However, growing the amount of data available is only feasible to a certain extent. In fact, translation quality is negatively affected when there is not enough training data for the specific domain to be tackled in production conditions [2, 3]. Hence, it is necessary to adapt the underlying models so that they are able work with the data to be dealt with. This problem, known as *domain adaptation*, is a very common problem in SMT, and the aim is to improve the performance of an SMT system trained on out-of-domain data by using limited amounts of in-domain data.

Domain adaptation methods can be split into two broad categories: *1)* domain adaptation methods that tackle the problem at the corpus level, for example, by weighting, selecting or joining the training corpora, and *2)* domain

adaptation methods that have an influence at the model level, by adapting directly the translation or language models.

Data selection (DS) is a domain adaptation method that fits under the first category. The underlying intuition implies selecting for training the best subset of sentence pairs from an available pool, so that the translation quality achieved in the target domain is improved. The current paper tackles DS by taking advantage of vector space representations of sentences, feeding on the most recent work on distributed representations of sentences [4, 5], with the ultimate goal of obtaining corpus subsets that minimize the bilingual training corpus size, while improving translation quality. In this work, we propose a new DS technique called Continuous Vector-Space Representation of Sentences for Data Selection (CRSDS), with the aim of selecting the best subset of sentences from an available pool, using a vector space representation of sentences.

The main contributions of this paper involve the necessary steps required to assess the novel CRSDS strategy, i.e.:

- We *describe* the algorithmic foundation of our CRSDS technique, which leverages a continuous space representation of sentences and words (Section 4).
- We *evaluate* the CRSDS technique, obtaining translation results that improve baseline translation quality (Section 6.2).
- We *compare* the CRSDS technique with some DS methods, considering different practical application scenarios (Sections 5, 6.3 and 6.4).

This paper is structured as follows. Section 2 describes the SMT framework. Section 3 summarises the related work. Section 4 presents our DS method using continuous vector-space representations. Section 5 describes the DS methods with which we compare our method. In Section 6, the experimental design and results are detailed. Finally, the main results of the work and future work are discussed in Section 7.

2. Statistical Machine Translation

One important breakthrough in SMT was provided by the use of log-linear models for modelling the translation process, proposed in [6] and reviewed in [3]. In SMT, log-linear models are defined as follows: given an input sentence $\mathbf{x}$ from a certain source language, the purpose is to find an output sentence $\mathbf{y}$ in a certain target language such that:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, p(\mathbf{y} \mid \mathbf{x}) = \underset{\mathbf{y}}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{x}, \mathbf{y}) \tag{1}$$

where λ_m is the weight assigned to $h_m(\mathbf{x}, \mathbf{y})$ and $h_m(\mathbf{x}, \mathbf{y})$ is a score function representing an important feature for the translation of $\mathbf{x}$ into $\mathbf{y}$, as for example the language model of the target language, a reordering model, or several translation models. M is the number of models (or features).

One of the most widely-used instances of log-linear models in SMT are phrase-based (PB) models [7, 3]. The basic idea of PB models is to segment $\mathbf{x}$ into phrases, and then translate each source phrase into a target phrase. For this purpose, phrase-tables are produced, in which a source phrase is listed together with several target phrases and the probability of translating the former into the latter.

Once the bilingual phrases have been extracted from a bilingual corpus, the features h_m can already be computed. At this point it is necessary to obtain appropriate values for the weight-vectors λ_m, this process is called *tuning*. The weights λ_m are normally optimised with the use of a development set. The most popular approach for adjusting λ_m is the one proposed in [8], commonly referred to as *minimum error rate training* (MERT).

Both feature values h_m and weight vectors λ_m are then leveraged in the decoding process, which typically implements a beam-search algorithm [3] to perform the actual translation of input sentence. The problem here is to find the best candidate hypothesis $\mathbf{y}^*$ according to the equation 1. The decoding problem is NP-complete [9], given that examining the complete search space (all possible translations of a given input sentence) is computationally very costly. Hence, the

decoding process often implements heuristic pruning strategies [10, 11], which however do not guarantee that optimal translation will be found.

Evaluation in SMT is a very controversial problem [3], [12]. An obvious method for evaluating SMT output is to manually evaluate whether a given translation is correct. However, human evaluation is way too costly for experimentation purposes, which entails the need of resorting to automatic evaluation metrics. Even though automatic metrics in SMT are under constant debate in the community, BLEU (Bilingual Evaluation Understudy) [12] is the most popular evaluation metric. BLEU measures the precision of unigrams, bigrams, trigrams, and four-grams with respect to a set of reference translations, with a penalty for too short sentences. Since BLEU measures precision, the higher the BLEU score, the better. In this paper, we will be evaluating translation output with BLEU, even though other metrics also exist, such as TER (Translation Edit Rate) [13] and METEOR [14]).

As it can be understood from the above discussion, phrases lie within the core of modern PB SMT systems. Given that phrases are extracted from parallel data, improving the quality of such training data is crucial towards improving translation quality. Hence, the purpose of this work is to devise an algorithm with the final objective of improving the quality of the bilingual data fed as training set to the PB model, by means of an appropriate DS strategy.

3. Related work

As anticipated during the introduction, data selection aims to obtain the best subset of a generic pool of data. Then, this set of data is concatenated with the in-domain training data, and such concatenation is then used for training the SMT system.

In this work, we will refer to the available pool of generic-domain sentences as *out-of-domain* corpus because we assume that it belongs to a different domain than the one to be translated. Similarly, we refer to the corpus belonging to the specific domain of the text to translated as *in-domain* corpus.

State-of-the-art DS approaches rely on the idea of choosing those sentence pairs in the out-of-domain training corpus that are in some way similar to an in-domain training corpus in terms of some different metrics.

Different works use perplexity-related DS strategies [15, 16, 17, 18]. In NLP, perplexity is a measurement of how well a model predicts a sample. A low perplexity indicates the models is good at predicting the sample. In this research direction, sentences in a out-of-domain corpus are ranked by their perplexity score according to an in-domain language model, and only the top percentage with lowest perplexity scores are retained as training data. This method, proposed by [15], is extremely easy to apply: first train an in-domain LM, then score each sentence in the out-of-domain, and select the highest ranked. In [16], the authors re-implemented the perplexity-based method, with the modification of using the cross-entropy of a given sentence instead of its perplexity. All these paper [19, 20, 21, 22] used this method and the good result has become a de-facto standard in the SMT research community. We apply this criterion as comparison with our DS technique.

Two different approaches are presented in [23]: one based on approximating the probability of an in-domain corpus and another one based on infrequent n-gram recovery. On the one hand, the technique based on approximating the probability relies on preserving the probability distribution of the task domain by wisely selecting the bilingual pairs to be used, excluding sentences that distort the actual probability. On the other hand, the second technique presented (the best-performing one) is based on the notion of infrequent n-gram, and will be explained in detail in section 5.

Other works have applied information retrieval methods for DS [24], in order to produce different sub-models which are then weighted. The baseline was defined as the result obtained by training only with the corpus that shares the same domain with the test. They claim that they are able to improve the baseline translation quality by adding new sentences retrieved with their method. However, they do not provide a comparison with a model trained on all the corpora available.

More recently, in [25] leveraged neural language models to perform DS, reporting substantial gains over conventional n-gram language model-based DS.

4. Continuous vector-space representation of sentence for DS

Here we describe our CRSDS method for SMT. For defining our strategy, the following steps are required:

1. A CVR of words (Section 4.2)
2. A CVR of sentences (using step 1) (Section 4.3)
3. A selection algorithm as such (using step 2) (Section 4.4)

With the purpose of simplifying notation, we start by defining the notion of similarity corpus in the next section.

4.1. Similarity corpus

The core idea of every DS method is to select a subset of the out-of-domain data that is considered to be the most relevant for translating a given set of data, which we will name in this work *similarity corpus* S. Ideally, S will be the text to be translated (T), and the DS method will ensure that the resulting subset of the training data is the best possible subset for translating T [23]. Nevertheless, in scenarios where a system is set for on-the-fly translation, such data T is not available in advance. For this reason, it is often the case that an in-domain set I (considered to be very similar, or at least belonging to the same domain as T) is used instead [24, 19]. Since in this paper we will define our approach independently of whether I or T is used, our data selection method will be defined in terms of S, and the experimental results will instantiate S to either I or T. Note that there is an important piece of information in I which is lacking in T: the target side of the bilingual data. In contrast, T contains the true data to be translated, albeit obviously without the sentence to be produced. Hence, DS approaches that intend to use I or T independently will need to be designed to use only source language data, i.e., not require the target sentence data.

4.2. Continuous vector-space representation of words

CVR of words have been widely used in a variety of NLP applications. These representations have recently demonstrated promising results across a variety of tasks [26, 27, 28, 29, 30], such as speech recognition, part-of-speech tagging, sentiment classification and identification and machine translation.

The idea of representing words in vector space was originally proposed by [31, 32]. The limitation of these proposals were that computational requirements quickly became unpractical for growing vocabulary sizes $|V|$. However, work performed recently in [33, 34, 4, 35] made it possible to overcome such drawback, while still relying on neural network language models, in which words are represented as high dimensional real valued vectors. These model have the purpose that words with similar meanings will map to similar vectors. The basic idea is to represent each word w_i in the vocabulary V, $w_i \in V$, with a real-valued vector of some fixed dimension D, i.e., $f(w_i) \in R^D \; \forall i = 1, \ldots, |V|$, capturing the similarity (lexical, semantic and syntactic) between the words.

Two approaches are proposed in [4], namely, the *Continuous Bag of Words Model* (CBOW) and the *Continuous Skip-Gram Model*. CBOW forces the neural net to predict the current word by means of the surrounding words, and Skip-Gram forces the neural net to predict surrounding words using the current word. These two approaches were compared to previously existing approaches, such as the ones proposed in [33], and [34], obtaining a considerably better performance in terms of training time. In addition, experimental results also demonstrated that the Skip-Gram model offers better performance on average, excelling especially at the semantic level [4]. These results were confirmed in our own preliminary work, and hence we used the Skip-Gram approach to generate our distributed representations of word.

We used the word2vec[1] toolkit to obtain the representations of words. The toolkit takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary V from the training corpus and then learns the

[1] `https://code.google.com/archive/p/word2vec/`

CVR of the words.

However, a problem that arises when using CVR of words is how to represent a whole sentence (or document) with a continuous vector. Following the idiosyncrasy described in the previous paragraph (i.e., semantically close words are also close in their CVR), we present in the next section the different sentence representations employed in the present work.

4.3. Continuous vector-space representation of sentences

Numerous works have attempted to extend the CVR of words to the sentence or phrase level (just to name a few, [36, 37, 29, 38, 39, 5]). In the present work, we used two different CVRs of sentences, which we will denote as $F(\mathbf{x})$ (or, in some cases and to simplify notation, $F_{\mathbf{x}}$):

1. The first one is the most intuitive approach, which relies on using a weighted arithmetic mean of all the words in the document or sentence (as proposed by [37, 40] :

$$F_{\mathbf{x}} = F(\mathbf{x}) = \frac{\sum_{w \in \mathbf{x}} N_{\mathbf{x}}(w) f(w)}{\sum_{w \in \mathbf{x}} N_{\mathbf{x}}(w)} \tag{2}$$

   where w is a word that appears in sentence $\mathbf{x}$, $f(w)$ is the CVR of w, obtained as described above, and $N_{\mathbf{x}}$ is the count of w in sentence $\mathbf{x}$. We will refer to this representation by `Mean-vec`.

2. A more sophisticated approach is presented by [5]. The authors adapted the continuous Skip-Gram model [4] to generate representative vectors of sentences or documents. *Document vectors* follow the Skip-Gram architecture to train a special vector $F_{\mathbf{x}}$ representing the sentence or document. The formalization of this technique goes beyond the scope of the current paper, but the reader is referred to the original paper in [5] for further information. We will refer to this representation by `Document-vec`[2].

[2] `http://radimrehurek.com/gensim/models/doc2vec`

4.4. CRSDS technique

In this section, we will describe principal contribution of this paper, namely, the CRSDS method which leverages the vector-space representation of sentences, detailed above. Since the objective of DS is to increase the informativeness of the in-domain training corpus, it seems important to choose out-of-domain sentences that provide information considered relevant with respect to the similarity corpus S.

Algorithm 1 shows the procedure. Here, G is the out-domain-corpus, $\mathbf{x}$ is an out-of-domain sentence ($\mathbf{x} \in G$), $F_{\mathbf{x}}$ is the CVR of $\mathbf{x}$, and $|G|$ is the number of sentences in G. Then, our objective is to select data from G such that it is the most suitable for translating data belonging to the similarity corpus S. For this purpose, we define $F_{\mathbf{s}}$ as the CVR of a sentence $\mathbf{s} \in S$.

Data: $F_{\mathbf{x}},\ \mathbf{x} \in G$; and $F_{\mathbf{s}},\ \mathbf{s} \in S$; threshold τ
Result: `Selected-corpus`

```
forall sentences s in S do
    forall sentences x in G do
        score(F_s, F_x) = sim_i(F_s, F_x, τ)
        if score(F_s, F_x) ≥ τ then
            add x to Selected-corpus
        end
    end
end
```

Algorithm 1: Pseudo-code for our DS technique (Section 4.4)

Algorithm 1 introduces $sim_i(\cdot, \cdot)$, which will be defined in Section 4.4.1.

4.4.1. Similarity functions

The most simple approach would be to implement a mechanism by which a sentence $\mathbf{x}$ would only be selected if its similarity score $cos(F_{\mathbf{s}}, F_{\mathbf{x}}) \geq \tau$, with τ

a certain threshold to be established empirically, i.e:

$$sim_0(F_{\mathbf{s}}, F_{\mathbf{x}}, \tau) = \begin{cases} cos(F_{\mathbf{s}}, F_{\mathbf{x}}) & \text{if } cos(F_{\mathbf{s}}, F_{\mathbf{x}}) \geq \tau \\ 0 & \text{if } cos(F_{\mathbf{s}}, F_{\mathbf{x}}) < \tau \end{cases} \tag{3}$$

As a function over $cos(\cdot, \cdot)$, the cosine similarity between two different sentence vectors:

$$cos(F_{\mathbf{s}}, F_{\mathbf{x}}) = \frac{F_{\mathbf{s}} \cdot F_{\mathbf{x}}}{\|F_{\mathbf{s}}\| \cdot \|F_{\mathbf{x}}\|} \tag{4}$$

Note that it would have been possible to use any other similarity metric. Here, the purpose of similarity function $sim_i(\cdot, \cdot)$ is to allow a projection from the original similarity metric, so as to allow higher flexibility. Note that the *best* value for $cos(\cdot, \cdot)$ is 1, and the *worst* value for $cos(\cdot, \cdot)$ is 0.

Nevertheless, this approach proved empirically to not be very useful: certain, very specific, sentences in S yield much higher similarity scores, dominating the ranking when establishing τ and leading to other sentences in S not getting the chance to promote any sentences in G at all, i.e., a small number of sentences in S account for the wide majority of sentences selected. This is problematic, since the final set selected in such case is only suitable for translating a very small subset of S.

Hence, we developed three different similarity functions $sim_i(\cdot), i \in \{1, 2, 3\}$, for the metric $cos(F_{\mathbf{s}}, F_{\mathbf{x}})$ with the purpose of solving this issue. Let us first define $G_{\mathbf{s},\tau} = \{\mathbf{x} \mid \forall \mathbf{x} \in G : sim_0(F_{\mathbf{s}}, F_{\mathbf{x}}) > \tau\}$. Then, the similarity functions used are defined as follows:

sim_1 The purpose of this first approach is to limit the amount of sentences $\mathbf{x} \in G$ that can be promoted by a certain sentence $\mathbf{s} \in S$. Let μ be the empirical average of $|G_{\mathbf{s},\tau}|$, i.e., $\mu = \sum_{\mathbf{s} \in S} |G_{\mathbf{s},\tau}|/|S|$, and σ the corresponding standard deviation of $|G_{\mathbf{s},\tau}|$. Since $cos(F_{\mathbf{s}}, F_{\mathbf{x}})$ establishes a natural ordering in G for each $\mathbf{s} \in S$, let us define $G'_{\mathbf{s},\tau}$ as the set of sentences with highest $cos(F_{\mathbf{s}}, F_{\mathbf{x}})$ value, restricted to $|G'_{\mathbf{s},\tau}| \leq \mu + 2\sigma$. Then, we define sim_1 as

follows:

$$sim_1(F_{\mathbf{s}}, F_{\mathbf{x}}, \tau) = \begin{cases} cos(F_{\mathbf{s}}, F_{\mathbf{x}}) & \text{if } \mathbf{x} \in G'_{\mathbf{s},\tau} \\ 0 & \text{if } \mathbf{x} \notin G'_{\mathbf{s},\tau} \end{cases} \quad (5)$$

sim_2 In this case, the purpose is to promote those sentences in G that are the most similar to the whole similarity corpus S. We implemented this intuitive concept as the arithmetic mean of $cos(\cdot, \cdot)$ for all sentences $\mathbf{s} \in S$, i.e.:

$$sim_2(F_{\mathbf{s}}, F_{\mathbf{x}}, \tau) = \frac{\sum_{\mathbf{s} \in S} cos(F_{\mathbf{s}}, F_{\mathbf{x}})}{|G_{\mathbf{s},\tau}|} \quad (6)$$

sim_3 This proposal is dramatically different from the previous ones, in that $cos(F_{\mathbf{s}}, F_{\mathbf{x}})$ is not employed as such. Instead, we computed a CVR of the whole corpus S, F_S, assuming S as the concatenation of all its sentences, and applied the threshold selection in line 4 of Algorithm 1 on such score:

$$sim_3(F_{\mathbf{s}}, F_{\mathbf{x}}, \tau) = cos(F_S, F_{\mathbf{x}}) \quad (7)$$

Notation has been slightly abused since S is not in the parameter list of sim_3, but has been omitted for clarity.

5. Comparision of data selection methods

In this section we present two standard DS methods. The first one, proposed by [16], is based in cross-entropy. Having been used in many different works [19, 20, 21], it has become a de-facto standard in the SMT research community. The second strategy we used is infrequent n-grams recovery. Presented in [23], it was the one to obtain the best results in their work, achieving significant improvements. Both strategies depend on the n-grams (i.e., sequences of n contiguous words) that compose the corpus considered, be it for building a language model (cross-entropy) or for determining which n-grams are infrequent (infrequent n-grams). Previous work [41] analysed the effect of varying the order of the n-grams, considering 2-grams (cross-entropy) and 5-grams (infrequent n-grams).

5.1. Cross-entropy method

As mentioned in Section 3, one established DS method consists in scoring the sentences in the out-of-domain corpus by their perplexity [15]. [16] use cross-entropy rather than perplexity, even though they are both monotonically related. The cross-entropy $H_C(\mathbf{x})$ of a given sentence $\mathbf{x} = \{x_1, \ldots, x_{|\mathbf{x}|}\}$, according to a given language model p estimated on corpus C, is typically estimated as [20]:

$$H_C(\mathbf{x}) = -\sum_{i=1}^{|\mathbf{x}|} \frac{1}{|\mathbf{x}|} \log p(x_i \mid x_1, \ldots, x_{i-1}) \tag{8}$$

Then, let I be an in-domain corpus, and G be an out-of-domain corpus from which we draw sentence $\mathbf{x}$. The cross-entropy score of $\mathbf{x}$ is defined as:

$$c(\mathbf{x}) = H_I(\mathbf{x}) - H_G(\mathbf{x}) \tag{9}$$

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 re-definition lies beyond the scope of this paper, since our purpose is only to use this method only for comparison purposes.

5.2. Infrequent ngrams recovery

The main idea underlying the infrequent n-grams recovery strategy [23] consists in increasing the information of the in-domain corpus by adding evidence for those n-grams (i.e., sequences of n consecutive words) that have been seldom observed in the in-domain corpus. This evidence is obtained by selecting sentences from the out-of-domain corpus. The n-grams that have never been seen or have been seen just a few times are called *infrequent n-grams*. An n-gram is considered infrequent when it appears less times than a given infrequency threshold t. Therefore, the idea is to select from the out-of-domain corpus the sentences which contain the most infrequent n-grams in the source sentences to be translated.

Let X be the set of n-grams that appear in the sentences to be translated and $\mathbf{m}$ one of them; let be $R(\mathbf{m})$ the counts of $\mathbf{m}$ in a given source sentence $\mathbf{x}$

of the out-of-domain corpus, and $C(\mathbf{m})$ the counts of $\mathbf{m}$ in the source language in-domain corpus. Then, the infrequency score $i(\mathbf{x})$ is defined as:

$$i(\mathbf{x}) = \sum_{\mathbf{m} \in X} \min(1, R(\mathbf{m})) \max(0, t - C(\mathbf{m})) \tag{10}$$

Then, the sentences in the out-of-domain corpus are scored using Equation 10 and given infrequency threshold t. This being done, the sentence $\mathbf{x}^*$ with the highest score $i(\mathbf{x}^*)$ is selected in each iteration. $\mathbf{x}^*$ is added to the in-domain corpus and is removed from the out-of-domain sentences. The counts of the n-grams $C(\mathbf{m})$ are updated with the counts $R(\mathbf{m})$ within $\mathbf{x}^*$ and therefore the scores of the out-of-domain corpus are updated. Note that t will determine the maximum amount of sentences that can be selected, since when all the n-grams within X reach the t frequency no more sentences will be extracted from the out-of-domain corpus.

6. Experiments

In this section, we describe the experimental framework employed to assess the performance of the data selection method described in Section 4. Then, we show the results for CRSDS strategy, followed by a comparative with two data selection methods (cross-entropy method and infrequent ngrams recovery).

For comparing the different DS methods, we explored the effect of varying empirically the selection constraint (e.g., the maximum number of selected sentences in Section 5.1 or infrequency threshold t in Section5.2, or τ in Section 4.4). These preliminary experiments were conducted on different corpora, not related to the task at hand in this paper. By doing so, we obtained different subsets of the selected out-of-domain corpus. Then, an SMT system is trained on each selected subset and tested on the test corpus. This provides several comparison points between the DS methods. In this setting, the different selection methods are compared based on how many sentences are required in order to reach the best BLEU score.

6.1. Experimental setup

We evaluated empirically the DS methods described in Section 4 and Section 5. As explained above, SMT systems need large corpora for training the underlying statistical models. Two corpora are dealt with in the DS task: an out-of-domain corpus G and an in-domain corpus (I in Section 4). 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.

For the out-of-domain corpus, we used the Europarl[3] corpus [42]. The Europarl corpus is composed of translations of the proceedings of the European parliament. As in-domain data, we used the EMEA corpus[4] corpus [43], which is available in 22 languages and contains documents from the European Medicines Agency. In order to make the results reported in this paper comparable with other works, standard partitions of the corpus will be used. The Medical-Test and Medical-Mert corpora are partitions established in the 2014 Workshop on Statistical Machine Translation (WMT) [5] [44] of the Association for Computational Linguistic. We focused on the English-French (En-Fr), German-English (De-En) and English-German (En-De) language pairs. We conducted experiments with different language pairs with the purpose of evaluating whether the conclusions drawn from one single language pair hold in further scenarios. The main figures of the corpora used are shown in Tables 1 and 2.

All experiments were carried out using the open-source phrase-based SMT toolkit Moses [45]. The language model used was a 5-gram, standard in SMT research, with modified Kneser-Ney smoothing [46], built with the SRILM toolkit [47]. The phrase table was generated by means of symmetrised word alignments obtained with GIZA++ [48]. The decoder features a statistical log-linear model including a phrase-based translation model, a language model, a distortion model and word and phrase penalties. The log-lineal combination weights

[3] www.statmt.org/europarl/

[4] www.statmt.org/wmt14/medical-task/

[5] www.statmt.org/wmt14/

Table 1: In-domain corpora main figures. (EMEA-Domain) is the in-domain corpus, (Medical-Test) is the evaluation data and (Medical-Mert) is development set used for adjusting λ in Equation 1. In this table, M denotes millions of elements and k thousands of elements, $|S|$ stands for number of sentences, $|W|$ for number of words (tokens) and $|V|$ for vocabulary size (types).

Corpus		$\|S\|$	$\|W\|$	$\|V\|$
EMEA-Domain	EN	1.0M	12.1M	98.1k
	FR		14.1M	112k
Medical-Test	EN	1000	21.4k	1.8k
	FR		26.9k	1.9k
Medical-Mert	EN	501	9.9k	979
	FR		11.6k	1.0k
Medical-Domain	DE	1.1M	10.9M	141k
	EN		12.9M	98.8k
Medical-Test	DE	1000	18.2k	1.7k
	EN		19.2k	1.9k
Medical-Mert	DE	500	8.6k	874
	EN		9.2k	979

Table 2: Out-of-domain corpus main figures (same abbreviations as in Table 1).

Corpus		$\|S\|$	$\|W\|$	$\|V\|$
Europarl	EN	2.0M	50.2M	157k
	FR		52.5M	215k
Europarl	DE	1.9M	44.6M	290k
	EN		47.8M	153k

λ (Equation 1) were optimized using MERT (minimum error rate training) [8]. Since MERT requires a random initialisation of λ that often leads to different

Table 3: Translation results using our DS method, in different configurations. `Mean` and `Doc` are the two different CVR methods, $sim(\cdot)$ denotes the three different similarity functions, $\#Sent$ for number of sentences, which are given in terms of the in-domain corpus size, and (+) the number of sentences selected.

	EN-FR		FR-EN		DE-EN	
Strategy	BLEU	# Sent	BLEU	# Sent	BLEU	# Sent
`bsln-emea`	28.6±0.2	1.0M	23.7±0.2	1.0M	15.6±0.1	1.0M
`bsln-all`	29.4±0.1	1.0M+1.5M	26.2±0.2	1.0M+1.5M	16.6±0.2	1.0M+1.5M
`Mean-`sim_0	29.4±0.2	1.0M+500k	25.6±0.3	1.0M+600k	16.6±0.2	1.0M+600k
`Mean-`sim_1	29.4±0.2	1.0M+347k	25.6±0.2	1.0M+439k	**16.9±0.2**	1.0M+**357k**
`Mean-`sim_2	29.6±0.3	1.0M+472k	25.7±0.2	1.0M+328k	16.7±0.2	1.0M+347k
`Mean-`sim_3	29.6±0.3	1.0M+137k	25.8±0.2	1.0M+394k	16.8±0.2	1.0M+496k
`Doc-`sim_0	29.6±0.3	1.0M+560k	25.8±0.1	1.0M+500k	16.8±0.2	1.0M+593k
`Doc-`sim_1	29.6±0.2	1.0M+284k	25.9±0.1	1.0M+365k	16.9±0.3	1.0M+440k
`Doc-`sim_2	29.7±0.2	1.0M+380k	**26.1±0.2**	1.0M+**403k**	16.9±0.3	1.0M+410k
`Doc-`sim_3	**29.8±0.2**	1.0M+**41k**	25.9±0.4	1.0M+406k	16.9±0.1	1.0M+440k

local optima being reached, every point in each plot of this paper constitutes the average of 10 repetitions with the purpose of providing robustness to the results. In the tables reporting translation quality, 95% confidence intervals of these repetitions are shown, but are omitted from the plots for purpose of clarity.

We compared the selection methods with two baseline systems. The first one was obtained by training the SMT system with in-domain training data (EMEA-Domain data). We will refer to this setup with the name of `bsln-emea`. A second baseline experiment has been carried out with the concatenation of the Europarl corpus and EMEA training data (i.e., all the data available). We will refer to this setup as `bsln-all`. In addition, we also included results for a purely random sentence selection without replacement. In the plots, each point corresponding to random selection represents the average of 5 repetitions.

In this work, SMT output will be evaluated by means of BLEU (BiLingual

Evaluation Understudy) [12]. BLEU is not an error rate, i.e. the higher the BLEU score, the better.

The word2vec toolkit (Section 4.2) has different parameters that need to be adjusted in during the training process. We adjusted two parameters: vector dimension v_size and n_c is the minimum number of times a given word needs to appear in the training data for its corresponding vector to be built. The values for $v\_size = 200$ and $n_c = 1$ were fixed for all the experiments reported in this paper.

6.2. Experimental results

As a first step to empirical evaluation for CRSDS technique, we analysed the performance of the two different vector representations of sentences (Section 4.3; `Mean-vec` and `Document-vec`), since these two methods have a great impact on the vectors obtained, and are bound to have an important impact on the data selection technique, and finally in the translation quality. In addition, we also studied the performance of the three similarity functions proposed in Section 4.4.1.

Table 3 shows the best results obtained with the different CVR methods, using the four different functions sim_i (see Section 4.4.1) and for each language pair. The values shows the best result for each strategy in terms of BLEU, and comparing the size of selected corpora. Note that translation quality remains very much similar, since the purpose of the extent to which the different DS strategies are able to reduce the amount of training data required, without any significant loss in translation quality. In this case, the similarity corpus S considered was the source test data T.

Several conclusions can be drawn:

- Translation quality using DS significantly improves over baseline (`bsln-emea`) translation quality.
- In EN-FR and EN-DE, translation quality using DS improves over `bsln-all`, but using a significantly less data (3% and 23%, respectively). In the case

of DE-EN, translation quality results are similar, but using only 27% of the data. Hence, we can safely state that our DS strategy is always able to deliver similar quality than using all the data, but only with a rough quarter of the data.

- `Document-vec` yields slightly better translation quality than `Mean-vec`. Although differences are not statistically significant, this might mean that `Document-vec` entails a better estimation of the sentence CVR.

- Lastly, sim_1, sim_2 or sim_3 seem to perform similarly. However, sim_0 does require significantly more sentences to reach comparable translation quality. Hence, sim_3 should be preferred: it is the cheapest in computational terms because it only requires one comparison with each $\mathbf{s} \in S$.

6.3. Comparative with cross-entropy selection

Once the effect of the different parameters in CRSDS method was analysed, we now pursue to compare our DS method with the cross-entropy method (Section 6.1). Results in Figure 1 show the effect of adding sentences to the in-domain corpus. Here, the similarity corpus is the in-domain set (i.e., $S = I$). We only show cross-entropy results using 2-grams, which was the best result according to previous work [41]. N-gram selection is not considered here because in this scenario the test data is not available. For CRSDS method, we tested both CVR methods (`Document-vec` and `Mean-vec`, combined with sim_3). Several conclusions can be drawn:

- All DS methods are mostly able to improve over random selection, specially when low amounts of data are added; in those cases where random yields better results, differences are not significant. This is reasonable, since all DS methods including random will eventually converge to the same point: adding all the data available. Even though these results should be expected, previous works (reported in Section 3) revealed that beating random was very hard.

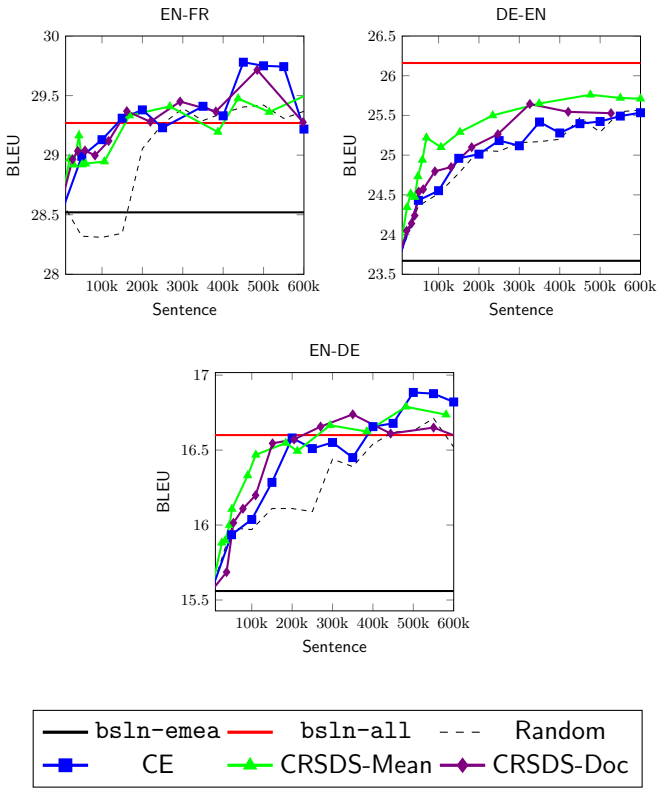

Figure 1: Effect on BLEU of adding sentences by means of CRSDS, cross-entropy, and random DS. Horizontal lines represent `bsln-emea` and `bsln-all`. The similarity corpus is the in-domain set (S=I)

- Results obtained with CRSDS method are slightly better (or similar) than the ones obtained with cross-entropy.

6.4. Comparative with infrequent ngrams recovery

We now pursue to compare CRSDS method with the infrequent ngrams method in Section 5.2. As exposed in section 5.2, this method requires the source Test corpus to be available for computing the *infrequent n-grams* list. For this reason, in this comparative the similarity corpus used was the source Test corpus (i.e., $S = T$). The results in Figure Fig. 2 show the effect of adding sentences to the in-domain corpus. In the case of CRSDS method, the same

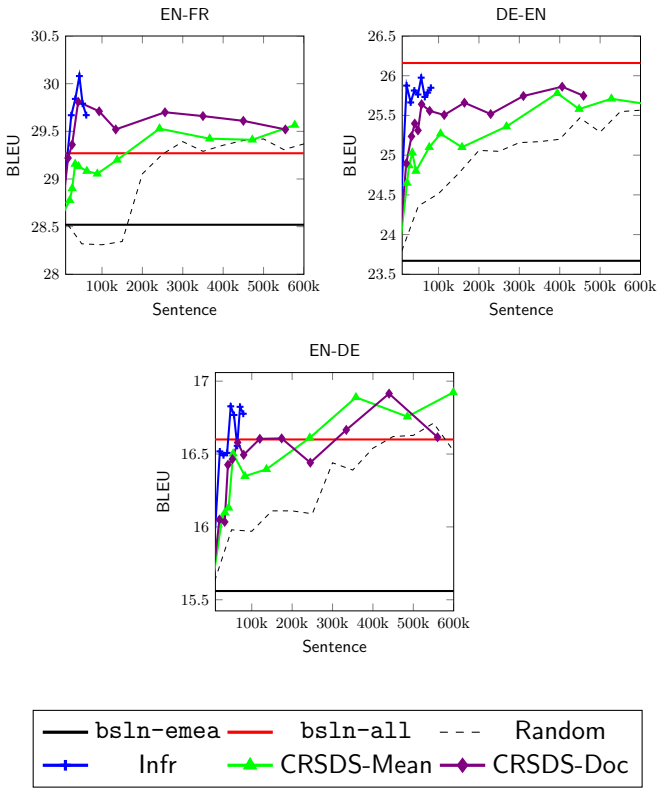

Figure 2: Effect on BLEU of adding sentences using CRSDS, infrequent n-grams recovery, and random DS. Horizontal lines represent the score the `bsln-emea` and `bsln-all` system. The similarity corpus is, in this case, the source test data (S=T).

approach as in previous section was used. Several conclusions can be drawn:

- Results show that, also in the case of $S = T$, DS yields better results than `bsln-emea`.
- The results achieved by CRSDS method are similar (i.e., not statistically different) from the results achieved by infrequent n-gram recovery, in all the languages studied, albeit requiring more sentences.
- The results achieved by CRSDS strategy leveraging the `Document-vec` representation are consistently better than those achieved by the cross-entropy method.

- Note that, for equal amount of sentences added, translation quality with CRSDS method is significantly better when $S = T$ as compared to $S = I$ (Figure 1). We understand that this happens because we use the Test corpus entails a better selection out-of-domain sentence.

6.4.1. Combination with infrequent ngrams recovery

Table 4: Summary of the best combination results obtained for each language. *#Sent* stands for number of sentences, wich is given in terms of the in-domain corpus size, and (+) for the number of sentences selected.

	EN-FR		FR-EN		DE-EN	
Strategy	BLEU	# Sent	BLEU	# Sent	BLEU	# Sent
`bsln-emea`	28.6±0.2	1.0M	23.7±0.2	1.0M	15.6±0.1	1.0M
`bsln-all`	29.4±0.1	1.0M+1.5M	26.2±0.2	1.0M+1.5M	16.6±0.2	1.0M+1.5M
`Random`	29.4±0.4	1.0M+500k	25.5±0.1	1.0M+600k	16.7±0.3	1.0M+600k
`Infr`	30.2±0.2	1.0M+44k	26.0±0.2	1.0M+57k	16.8±0.2	1.0M+71k
`CRSDS`	29.8±0.2	1.0M+41k	25.9±0.3	1.0M+400k	16.9±0.1	1.0M+440k
`CRSDS+Infr`	**30.0±0.1**	1.0M+**14k**	**25.9±0.2**	1.0M+**37k**	**16.7±0.2**	1.0M+**27k**

In this section, we present the experimental results obtained through a re-selection process, in which we use CRSDS method to obtain a first selected corpus, which is then fed as out-of-domain corpus G to the infrequent n-grams method. The ultimate purpose is to combine the advantages of both methods, i.e., reducing as much as possible the number of sentences added, while improving translation quality at the same time.

Table 4 shows the results obtained. Interestingly, the combined DS method is able to yield very similar translation quality than each of the DS methods individually, but with a much lower amount of sentences. Specifically, the combination is able to reach the same translation quality by adding as few as 1% of the out of domain corpus for EN-FR, 2.5% for DE-EN and 1.6% for EN-DE. We consider this specially relevant, since it proves that DS has a very important

potential for reducing the computational resources required for training SMT systems.

6.5. Summary of the results

Table 5: Summary of the best results obtained with each setup. *#Sent* for number of sentences wich are given in terms of the in-domain corpus size, and (+) the number of sentences selected.

	EN-FR		FR-EN		DE-EN	
Strategy	BLEU	# Sent	BLEU	# Sent	BLEU	# Sent
`bsln-emea`	28.6±0.2	1.0M	23.7±0.2	1.0M	15.6±0.1	1.0M
`bsln-all`	29.4±0.1	1.0M+1.5M	26.2±0.2	1.0M+1.5M	16.6±0.2	1.0M+1.5M
`Random`	29.4±0.4	1.0M+500k	25.5±0.1	1.0M+600k	16.7±0.3	1.0M+600k
`CE`	29.8±0.1	1.0M+450k	25.5±0.3	1.0M+600k	16.8±0.2	1.0M+500k
`CRSDS`	29.7±0.2	1.0M+485k	25.8±0.2	1.0M+470k	16.7±0.2	1.0M+350k
`Infr`	30.2±0.2	1.0M+44k	26.0±0.2	1.0M+57k	16.8±0.2	1.0M+71k
`Cvr`	29.8±0.2	1.0M+41k	25.9±0.3	1.0M+400k	16.9±0.1	1.0M+440k
`CRSDS+Infr`	**30.0±0.1**	1.0M+**14k**	**25.9±0.2**	1.0M+**37k**	**16.7±0.2**	1.0M+**27k**

Table 5 shows the best results obtained with our strategy and the other two techniques for each language pair (EN-FR, DE-EN, EN-DE), with the corresponding similarity corpus instantiations. As shown, our CRSDS method is able to yield competitive results in both scenarios considered. We understand that is important, since it proves the usefulness of our proposal, with respect to the other techniques: it provides a single state-of-the-art approach to DS selection, regardless of the scenario.

6.6. Example translations

Translation examples are shown in Table 6. In the first example, our method is not able to obtain the % symbol, as provided in the reference. This is not only casual, since our method increases the coverage of *percent* with translations with *pour cent*, hence leading the system to avoid the use of the % symbol, present

in the in-domain corpus (as evidenced by the `Bsl` system). Nevertheless, note that this is not an actual mistake in translation terms, but will be penalised by BLEU (which measures n-gram precision). In addition, all the systems present the same lexical choice error with word (*développer*). However, this is so because this is the most likely translation in our data, both in-domain and out-of-domain. In the second example, our CRSDS method its the only one system able to obtain the reference translation. This happens because we are able to add the appropriate information for translating *aortic stenosis* (*aortic sténosis*) and *as a* (*comme une*), and do not introduce incorrect information, as is the case with infrequent n-grams (*définie* is replaced by *défini*).

Table 6: Translation examples with the SMT systems built: Src (source sentence), Bsl (baseline), All (all the data available), Infr (Infrequent n-grams), Entr (Cross-entropy), `CRSDS` (Continuouns vector-space representation of sentence for data selection) and Ref (reference).

`Src`	5 percent of people with ulcerative colitis develop cancer .
`Bsl`	5 % des personnes avec colite ulcreuse *de développer* un cancer .
`All`	5 *pour cent* des personnes avec colite ulcéreuse *développer* un cancer .
`Infr`	5 % des personnes avec colite ulcéreuse de *développer* un cancer .
`CE`	5 *pour cent* des personnes avec colite ulcéreuse *de développer* un cancer .
`CRSDS`	5 *pour cent* des personnes avec colite ulcéreuse *développer* un cancer .
`Ref`	5 % des personnes souffrant de colite ulcéreuse sont atteintes de cancer .
`Src`	an aortic stenosis is defined as a reduction of the surface .
`Bsl`	une *aortic sténosis* est définie *par une* réduction de la surface .
`All`	une sténose aortique est définie *par une* réduction de la surface .
`Infr`	un *aortic sténosis* est *défini* comme une réduction de la surface .
`CE`	une sténose aortique est définie *par une* réduction de la surface .
`CRSDS`	une sténose aortique est définie comme une réduction de la surface .
`Ref`	une sténose aortique est définie comme une réduction de la surface .

7. Conclusion and future work

Data selection has been receiving an increasing amount of attention within the SMT research community. There are a lot of data selection methods based in different ideas. In this work, we presented a novel data selection method based on CVR of sentences or documents, which yield similar representations for semantically close sentences, called Continuous Vector-Space Representation of Sentence for Data Selection (CRSDS). In addition, we perform a comparison of our technique with two different state-of-the-art techniques, which are very common in the literature and follow two different scenarios: the cross-entropy method selects a subset given in-domain data, and in contrast the infrequent n-grams recovery selects a subset of the out-of-domain data that is considered to be most relevant for the data to be translated. When comparing our method, an important conclusion stands out: our method is able to yield similar or better quality than the state-of-the-art methods for each scenario. Combining the two techniques (our method and infrequent n-grams) in the second scenario yields again similar quality, but with much less data.

We understand that the results obtained indicate that our data selection technique is able to take advantage of the benefits claimed by CVR methods: being able to represent the semantic and syntactic relationship between words or sentences, it goes beyond their string-based representation and is able to tackle better data sparsity problems (e.g., where a given word or sentence is only seen once in the test data).

In future work, we will carry out new experiments with bigger and more diverse data sets. In addition, we will modify the original cross-entropy definition to set similarity corpus $S = T$. We also intend to combine the strategies proposed in more sophisticated ways.

Acknowledgments

Work supported by the Generalitat Valenciana under grant ALMAMATER (PrometeoII/2014/030) and the FPI (2014) grant by Universitat Politcnica de

Valncia.

References

[1] T. Brants, A. C. Popat, P. Xu, F. J. Och, J. Dean, Large language models in machine translation, in: Proc. of EMNLP/CoNLL, 2007, pp. 858–867.

[2] C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, J. Schroeder, (meta-) evaluation of machine translation, in: Proc. of WMT, 2007, pp. 136–158.

[3] P. Koehn, Statistical machine translation, Cambridge University Press, 2010.

[4] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, `arXiv:1301.3781` (2013).

[5] Q. V. Le, T. Mikolov, Distributed representations of sentences and documents, `arXiv:1405.4053` (2014).

[6] F. J. Och, H. Ney, Discriminative training and maximum entropy models for statistical machine translation, in: Proc. of ACL, 2002, pp. 295–302.

[7] P. Koehn, F. J. Och, D. Marcu, Statistical phrase-based translation, in: Proc. of NAACL, 2003, pp. 48–54.

[8] F. J. Och, Minimum error rate training in statistical machine translation, in: Proc. of ACL, 2003, pp. 160–167.

[9] R. Udupa, H. K. Maji, Computational complexity of statistical machine translation., in: Proc. of EACL, 2006, pp. 25–32.

[10] C. Tillmann, H. Ney, Word reordering and a dynamic programming beam search algorithm for statistical machine translation, Computational linguistics 29 (2003) 97–133.

[11] D. Ortiz-Martínez, Advances in fully-automatic and interactive phrase-based statistical machine translation, Ph.D. thesis, Universitat Politècnica de València (2011).

[12] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proc. of ACL, 2002, pp. 311–318.

[13] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul, A study of translation edit rate with targeted human annotation, in: Proc. of AMTA, 2006, pp. 223–231.

[14] S. Banerjee, A. Lavie, Meteor: An automatic metric for mt evaluation with improved correlation with human judgments, in: Proc. of the ACL Workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, 2005, pp. 65–72.

[15] J. Gao, J. Goodman, M. Li, K.-F. Lee, Toward a unified approach to statistical language modeling for chinese, ACM TALIP 1 (1) (2002) 3–33.

[16] R. C. Moore, W. Lewis, Intelligent selection of language model training data, in: Proc. of ACL, 2010, pp. 220–224.

[17] K. Yasuda, R. Zhang, H. Yamamoto, E. Sumita, Method of selecting training data to build a compact and efficient translation model., in: Proc. of IJCNLP, 2008, pp. 655–660.

[18] G. Foster, C. Goutte, R. Kuhn, Discriminative instance weighting for domain adaptation in statistical machine translation, in: Proc of EMNLP, 2010, pp. 451–459.

[19] A. Axelrod, X. He, J. Gao, Domain adaptation via pseudo in-domain data selection, in: Proc. of EMNLP, 2011, pp. 355–362.

[20] A. Rousseau, Xenc: An open-source tool for data selection in natural language processing, The Prague Bulletin of Mathematical Linguistics 100 (2013) 73–82.

[21] H. Schwenk, A. Rousseau, M. Attik, Large, pruned or continuous space language models on a gpu for statistical machine translation, in: Proc. of NAACL-HLT, 2012, pp. 11–19.

[22] S. Mansour, J. Wuebker, H. Ney, Combining translation and language model scoring for domain-specific data filtering, in: Proc. of IWSLT, 2011, pp. 222–229.

[23] G. Gascó, M.-A. Rocha, G. Sanchis-Trilles, J. Andrés-Ferrer, F. Casacuberta, Does more data always yield better translations?, in: Proc. of EACL, 2012, pp. 152–161.

[24] Y. Lü, J. Huang, Q. Liu, Improving statistical machine translation performance by training data selection and optimization, in: Proc. of EMNLP, 2007, pp. 343–350.

[25] K. Duh, G. Neubig, K. Sudoh, H. Tsukada, Adaptation data selection using neural language models: Experiments in machine translation., in: Proc. of ACL, 2013, pp. 678–683.

[26] H. Schwenk, Continuous space language models, Computer Speech & Language 21 (2007) 492–518.

[27] R. Collobert, J. Weston, A unified architecture for natural language processing, in: Proc. of ICML, 2008, pp. 160–167.

[28] X. Glorot, A. Bordes, Y. Bengio, Domain adaptation for large-scale sentiment classification: A deep learning approach, in: Proc. of the ICML, 2011, pp. 513–520.

[29] R. Socher, C. C. Lin, C. Manning, A. Y. Ng, Parsing natural scenes and natural language with recursive neural networks, in: Proc. of ICML, 2011, pp. 129–136.

[30] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, `arXiv:1409.1259` (2014).

[31] J. L. McClelland, D. E. Rumelhart, P. R. Group, et al., Parallel distributed processing, Vol. 2, Cambridge University Press, 1987.

[32] J. L. Elman, Finding structure in time, Cognitive science 14 (2) (1990) 179–211.

[33] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, JMLR 3 (2003) 1137–1155.

[34] T. Mikolov, M. Karafit, L. Burget, J. Eernocký, S. Khudanpur, Recurrent neural network based language model, in: Proc. of INTERSPEECH, 2010, pp. 1045–1048.

[35] R. Paulus, R. Socher, C. D. Manning, Global belief recursive neural networks, in: Proc. of NIPS, 2014, pp. 2888–2896.

[36] J. Mitchell, M. Lapata, Composition in distributional models of semantics, Cognitive Science 34 (8) (2010) 1388–1429.

[37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proc. of NIPS, 2013, pp. 3111–3119.

[38] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, C. D. Manning, Semi-supervised recursive autoencoders for predicting sentiment distributions, in: Proc. of EMNLP, 2011, pp. 151–161.

[39] Y. Kim, Convolutional neural networks for sentence classification, `arXiv:1408.5882` (2014).

[40] M. Kågebäck, O. Mogren, N. Tahmasebi, D. Dubhashi, Extractive summarization using continuous vector space models, in: Proc. of the Workshop on Continuous Vector Space Models and their Compositionality, 2014, pp. 31–39.

[41] M. Chinea-Rios, G. Sanchis-Trilles, F. Casacuberta, Bilingual sentence selection strategies: comparative and combination in statistical machine translation systems, in: Proc. of the VII JTH and III Iberian SLTech Workshop, 2014, pp. 227–236.

[42] P. Koehn, Europarl: A parallel corpus for statistical machine translation, in: Proc. of MT Summit, 2005, pp. 79–86.

[43] J. Tiedemann, News from opus - a collection of multilingual parallel corpora with tools and interfaces, in: Proc. of RANLP, 2009, pp. 237–248.

[44] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, L. Specia (Eds.), Proceedings of the Ninth Workshop on SMT, ACL, 2014.

[45] P. Koehn, 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, E. Herbst, Moses: open source toolkit for statistical machine translation, in: Proc. of ACL, 2007, pp. 177–180.

[46] R. Kneser, H. Ney, Improved backing-off for m-gram language modeling, in: Proc. of ICASSP, 1995, pp. 181–184.

[47] A. Stolcke, SRILM - an extensible language modeling toolkit, in: Proc. of ICSLP, 2002, pp. 901–904.

[48] F. J. Och, H. Ney, A systematic comparison of various statistical alignment models, Computational Linguistics 29 (1) (2003) 19–51.