Creating the best development corpus for Statistical Machine Translation systems

Mara Chinea-Rios[1] **Germán Sanchis-Trilles**[2] **Francisco Casacuberta**[1]
[1] Pattern Recognition and Human Language Technology Research Center
Universitat Politècnica de València, València, Spain
{machirio, fcn}@prhlt.upv.es
[2] Sciling, València, Spain
sanchis@sciling.es

Abstract

We propose and study three different novel approaches for tackling the problem of development set selection in Statistical Machine Translation. We focus on a scenario where a machine translation system is leveraged for translating a specific test set, without further data from the domain at hand. Such test set stems from a real application of machine translation, where the texts of a specific e-commerce were to be translated. For developing our development-set selection techniques, we first conducted experiments in a controlled scenario, where labelled data from different domains was available, and evaluated the techniques both with classification and translation quality metrics. Then, the best-performing techniques were evaluated on the e-commerce data at hand, yielding consistent improvements across two language directions.

1 Introduction

Tuning is a critical step in every system that presents a weighted combination of features. By adjusting the weights so that they best fit the target distribution, this process typically yields important improvements on the performance of the system developed. However, selecting an appropriate development set is key for this process to reach its goal.

In Statistical Machine Translation (SMT), the tuning step implies optimizing the log-linear weights $\{\lambda_1 \dots \lambda_m \dots \lambda_M\}$ of a discriminative model that implements a weighted combination of features $\{h_1 \dots h_m \dots h_M\}$, considered relevant in the translation process:

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

with $\mathbf{x}$ and $\mathbf{y}$ being the source and target sentences.

Such optimization has become de-facto standard in SMT, thanks to the wide-spread adoption of tuning algorithms such as Minimum Error Rate Training (MERT) (Och, 2003) or the Margin Infused Relaxed Algorithm (MIRA) (Cherry and Foster, 2012). The purpose of these algorithms is to adjust the log-linear weights such that the model distribution best fits the target distribution, or the target metric by which the system is evaluated.

Given that the amount of weights λ_m is typically around 10 or 20, the size of the development corpus required for tuning is typically in the range of hundreds or a few thousands of sentences. However, such corpus is typically required to be disjoint from the training corpus, used to estimate the features h_m, and its selection is critical, having an important impact on the system's performance if the development set of choice is too different from the test set at hand (Koehn, 2010).

The Data Selection (DS) task is stated as the problem of selecting the best sub-corpus of sentences from an available pool of sentences, with which to train a machine learning system. This paper deals with DS, but here the aim is to select, out of an available pool of sentences, the best development corpus for a given test set, for the purpose of log-linear weight optimization in SMT.

We study our development DS techniques in two different tasks. In the first case, the purpose is to

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 99–108
Alacant, Spain, May 2018.

analyse the behaviour of our techniques in a controlled scenario where the data is labelled according to domain. The goal is to study our methods' capacity of correctly predicting the domain labels, in addition to the translation quality achieved. In the second scenario, we evaluate the techniques presented in a real task, where a specific test set belonging to the texts of a real e-commerce site is provided, without domain labels.

The main contributions of this paper involve the necessary steps required to assess our novel development set selection techniques:

- We propose three different development DS (DDS) techniques: LD-DDS computes the Levenshtein Distance between the candidate sentences and the test sentences (Section 3); TF-DDS is based on the *t*erm frequency – inverse document frequency, which can be seen as a way of computing a numeric representation for a sentence (Section 4.1); lastly, CVR-DDS leverages a vector-space representation of sentences, relying on word the embeddings by (Mikolov et al., 2013) (Section 4.2).
- We study our DDS techniques in a controlled scenario, where domain labels are available (Section 5.2).
- We validate our DDS techniques in a real e-commerce translation task, with results that improve over random selection (Section 5.3).

This paper is structured as follows. Sections 3 and 4 present our different DDS methods. Section 5 presents the experiments: in Section 5.2, we present the analysis derived from the controlled experiment; Section 5.3 presents the results achieved with the real e-commerce task. Section 2 summarises related work. Conclusions and future work are discussed in Section 6.

2 Related works

The work presented here is close in concept to the domain adaptation scenario. Domain adaptation in SMT systems received considerable attention from the research community. Different domain adaptation techniques, including data selection, mixture models, etc., have been developed for different scenarios. A wide variety of data selection methods have been used over the years, where the main principle is to measure the similarity of sentences from the out-of-domain corpus to some in-domain corpus, either the development or the (source side of the) test set. Such similarity is often based on information theory metrics, like perplexity or cross entropy. In the last years, perplexity-based, or cross-entropy based, methods have become more common (Moore and Lewis, 2010; Axelrod et al., 2011; Rousseau, 2013; Schwenk et al., 2012; Mansour et al., 2011). Cross-entropy difference is a typical and well-established ranking function. Techniques based on information retrieval have also been widely used for data selection (Hildebrand et al., 2005; Lü et al., 2007). Furthermore, (Duh et al., 2013) leveraged neural language models to perform DS, reporting substantial gains over conventional n-gram language model-based DS. Finally, many researchers have used convolutional neural networks (CNN) in the domain adaptation field (Chen and Huang, 2016; Chen et al., 2016; Peris et al., 2016).

All the above DS approaches assume that the selection corpus is used to train or combine the SMT models. However, previous research on selecting the appropriate development corpus also exists. Such research can be split into two categories: in the first category, a development set is chosen, from among several "closed" development sets, based on the test set at hand (transductive learning) (Li et al., 2010; Zheng et al., 2010; Liu et al., 2012). The second category deals with the problem without knowing the test set beforehand, but knowing the domain of the test set (inductive learning). Previous work on development data selection for unknown test sets includes (Hui et al., 2010; Song et al., 2014). Note that the work presented here has an important difference with both transductive and inductive learning: even though it is closer to the transductive learning setting, all these works are based on selecting the most adequate development corpus from a collection of "closed" development corpora, with the purpose of choosing the one that belongs to the test set domain. In our case, we want to construct a specific development corpus for a given test corpus, without knowing the domain of the test set.

3 Levenshtein Distance DDS

The first DDS technique proposed involves computing the edit distance (Levenshtein Distance) between a candidate sentence and the closest sentence in the test set. Here, the intuition is to consider that a given sentence to be included in the development set D is a good candidate if it is not

too far away from the sentences in the test set T, as measured by the Levenshtein Distance. We will refer to this technique as LD-DDS.

The Levenshtein Distance (LD) (Levenshtein, 1966) is a string metric for measuring the difference between two sequences (words or sentences). The LD between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to make them match.

Algorithm 1 shows the procedure. Here, P is the pool of sentences available, $[\mathbf{x}_p, \mathbf{y}_p]$ is an out-of-domain sentence pair ($[\mathbf{x}_p, \mathbf{y}_p] \in P$), and $|P|$ is the number of sentences in P. Then, our objective is to select data from P such that it is the most suitable for translating data belonging to the test corpus T (composed only of source sentences).

Data: pool P; test data T; threshold τ
Result: Development corpus D
forall $\mathbf{t}$ *in* T **do**
  forall $[\mathbf{x}_p, \mathbf{y}_p]$ *in* P **do**
    if $LD(\mathbf{t}, \mathbf{x}_p) \leq \tau$ **then**
      if $[\mathbf{x}_p, \mathbf{y}_p] \notin D$ **then**
        add $[\mathbf{x}_p, \mathbf{y}_p]$ to D
      end
    end
  end
end

Algorithm 1: Pseudo-code for LD-DDS.

Algorithm 1 introduces the $LD(\cdot, \cdot)$ function, which computes the LD between two given sentences. Note that threshold τ establishes the size of the development corpus, and will need to be fixed empirically (Section 5.2).

4 DDS with vector-space representations

Here, we present two other DDS selection techniques, where the common point is that they both leverage a continuous vector-space representation of the sentences involved. First, we will describe our technique in abstract terms, and then we will present two different candidates for obtaining a continuous vector-space representation $F(\mathbf{x})$ (or $F_\mathbf{x}$ for short) of a given sentence $\mathbf{x}$.

Here, the intuition is to select as candidate sentences those whose vector-space representation is similar to those in the test set, assuming that similar sentences will have similar vectors.

The advantage of having a continuous vector-space representation of the test sentences is that a centroid can be computed, which can be assumed to be a sort of prototype of the sentences present in the test set. Note it was not possible to compute such centroid in the case of LD-DDS (Section 3).

Perhaps the best way to explaining this intuition is graphically, as shown in Figure1. This figure is a graphical example of the idea that we follow in this section, where sentences are represented in a two-dimensional vector-space. Here, blue points are the representation of the test sentences and red points represent the vectors of the sentences of the available pool of sentences, from which the development set is to be selected. Assuming that similar sentences will have a similar vector-space representation, the vectors of the test corpus will be very closer to each other, but the vectors for the general pool of sentences will be more disperse. The idea in our method is to draw a circle boundary, containing all test-sentences within it, and (hopefully) only a few of the sentences in the candidate pool. The radius of this circumference (or hyper-sphere in a multi-dimensional vector-space) is established as the distance between the centroid of the test set, and the furthest of the test sentences.

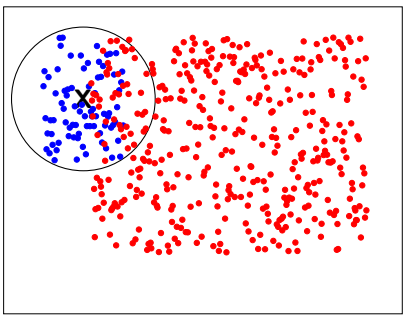

Figure 1: Graphical representation of the intuition behind our vector-space selection techniques. Red points represent the development sentence vectors, blue points represent the test sentence vectors. X is the centroid for the test vectors and the circumference represents the boundary obtained.

Algorithm 2 shows the procedure. Here, P is the pool of candidate sentences, $[\mathbf{x}_p, \mathbf{y}_p]$ is a candidate sentence pair, with $[\mathbf{x}_p, \mathbf{y}_p] \in P$, $F_\mathbf{x}$ is the vector-space representation of $\mathbf{x}$, and $|P|$ is the number of sentences in P. Then, our objective is to select data from P such that it is the most suitable for translating data belonging to the source test data T. For this purpose, we define $F_\mathbf{t}$ as the vector-space representation of a sentence $\mathbf{t} \in T$.

Algorithm 2 introduces several functions:

- $centroid(\cdot)$: calculates centroid $F_T = \{F_{T1} \dots F_{Tz} \dots F_{TZ}\}$ for test corpus T, as-

Data: Pool P; test data T
Result: Development corpus D
$F_T = centroid(T)$; $\rho = inf$
forall $\mathbf{t}$ *in* T **do**
  if $cos(F_{\mathbf{t}}, F_T) \leq \rho$ **then**
    $\rho = cos(F_{\mathbf{t}}, F_T)$
  end
end
forall $[\mathbf{x}_p, \mathbf{y}_p]$ *in* P **do**
  if $cos(F_{\mathbf{x}_p}, F_T) \geq \rho$ **then**
    add $[\mathbf{x}_p, \mathbf{y}_p]$ to D
  end
end

Algorithm 2: Pseudo-code for DDS leveraging vector-space representations of sentences.

suming a Z-dimensional vector-space:

$$F_{Tz} = \frac{1}{|T|} \sum_{t}^{|T|} F_{tz} \tag{2}$$

- $cos(\cdot, \cdot)$: the cosine similarity between two different vectors, e.g.:

$$cos(F_{\mathbf{t}}, F_T) = \frac{F_{\mathbf{t}} \cdot F_T}{\|F_{\mathbf{t}}\| \cdot \|F_T\|} \tag{3}$$

In addition, ρ represents the radius of the circumference, which is computed in lines 2 to 6 (the first **forall** loop) in Algorithm 2.

Once the selection algorithm has been established, we need to define how to represent sentences in a Z-dimensional space. Using vector-space representation for textual data (word, sentence or document) is not a new idea and has been widely employed in a variety of NLP applications. These representations have recently demonstrated promising results across a variety of tasks.

In this paper, we used two different approaches for representing sentences in a continuous vector-space: the popular *term frequency – inverse document frequency* (TF-IDF), and sentence embeddings (Mikolov et al., 2013). The basic idea is to represent a sentence $\mathbf{x}$ with a real-valued vector of some fixed dimension Z, i.e., $F(\mathbf{x}) \in R^Z$ that is able to capture similarity (lexical, semantic or syntactic) between a given pair of sentences.

4.1 TF-IDF representation

The TF-IDF (Term Frequency and Inverse Document Frequency) values can be used to create vector representations of sentence or documents. Using this kind of representation in a common vector-space is called vector space model (Salton et al., 1975), which is not only used in information retrieval but also in a variety of other research fields like machine learning (i.e. clustering, classification, information retrieval).

Each sentence $\mathbf{x} \in P$ is represented as a vector $F_{\mathbf{x}} = (F_{\mathbf{x}1}, \ldots, F_{\mathbf{x}k}, \ldots, F_{\mathbf{x}|V|})$, where $|V|$ is the size of the vocabulary V. Then, each $F_{\mathbf{x}k}$ is calculated as follows:

$$F_{\mathbf{x}k} = tf_{\mathbf{x}k} \cdot log(idf_k) \tag{4}$$

where $tf_{\mathbf{x}k}$ is the Term Frequency (TF), computed as the raw frequency of word $\mathbf{x}_k$ in a sentence, i.e. the number of times that word $\mathbf{x}_k$ occurs in sentence $\mathbf{x}$. idf_k is the Inverse Document Frequency (IDF), which is a measure of how much information word $\mathbf{x}_k$ provides, i.e., whether the term is common or rare across corpus P, computed as:

$$idf_k = \frac{|P|}{|\{\mathbf{x} \in P : \mathbf{x}_k \in \mathbf{x}\}|} \tag{5}$$

where $|P|$ is the number of sentences in corpus P, and $|\{\mathbf{x} \in P : \mathbf{x}_k \in \mathbf{x}\}|$ is number of sentences of P where the word $\mathbf{x}_k$ appears.

We will refer to the DDS technique that derives from using TF-IDF in Algorithm 2 as TF-DDS.

4.2 Continuous vector-space representation

The idea of representing words or sentence in a continuous vector-space employing neuronal networks was initially proposed by (Hinton, 1986; Elman, 1990). Continuous vector-space representations (CVR) of words or sentences have been widely leveraged in a variety of natural language applications and demonstrated promising results across a variety of tasks, such as speech recognition, part-of-speech tagging, sentiment classification and identification and machine translation just to name a few; (Schwenk et al., 2012; Glorot et al., 2011; Socher et al., 2011; Cho et al., 2014; Chinea-Rios et al., 2016).

In this paper, we use a sophisticated CVR for obtaining the representation of the sentences dealt with in our DDS method. Specifically, in (Le and Mikolov, 2014), the authors presented a CVR sentence approach. The authors adapted the continuous Skip-Gram model (Mikolov et al., 2013) to generate representative vectors of sentences or documents. *Document vectors* follow the Skip-Gram architecture to train a particular vector $F_{\mathbf{x}}$

representing the sentence or document. This work leverages the propose by (Le and Mikolov, 2014). We will refer to this representation by CVR[1], and to the DDS technique derived from using CVR in Algorithm 2 as CVR-DDS.

5 Experiments

In this section, we describe the experimental framework employed to assess the performance of the DDS methods described in Sections 3 and 4. For this purpose, we studied their behaviour in two separate tasks: a controlled scenario with labelled data, and a real e-commerce translation task. We will first detail the experimental setup employed, which is common to both tasks, and then we will report on each one of the tasks and their results.

5.1 Experimental setup

All 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 employing symmetrised word alignments obtained with GIZA++ (Och and Ney, 2003). The log-lineal combination weights λ were optimized using MERT (Minimum Error Rate Training) (Och, 2003). Since MERT requires a random initialisation of λ that often leads to different local optima being reached, every result in this paper constitutes the average of 10 repetitions.

To study to which extent weight optimization could yield improvements in translation quality, and hence obtain an upper bound for the performance of our DDS techniques, we will also report results with a so-called *oracle*, in which tuning was performed directly using the test set. Note that this setting is not realistic, but is useful to understand how much room for improvement there is by only choosing the development set wisely.

In addition to *oracle*, two more comparative results will also be provided: *baseline*, that is obtained by a translation system where tuning was performed on the original out-of-domain data; and *in-domain*, where tuning is performed using an in-domain development set, and is hence a good reference for comparison purposes if we were to assume that such development set is not available.

Translation quality will be measured as:

- BLEU (BiLingual Evaluation Understudy) (Papineni et al., 2002) measures n-gram precision with respect to a reference set, with a penalty for sentences that are too short.
- TER (Translation Error Rate) (Snover et al., 2006) is an error metric that computes the minimum number of edits (including swaps) required to modify the system hypotheses so that they match the reference.
- METEOR (Lavie, 2014) is a precision metric that includes stemmed and synonym matches when measuring the similarity between the system's hypotheses and the references.

For the case of CVR-DDS (Section 4.2), two meta-parameters need to be fixed: $Z = 200$, the dimension of the vector-space, and $n_c = 1$, the minimum number of times a given word needs to appear in the training data for its corresponding vector to be built. These values were fixed according to preliminary research, and maintained for all the experiments reported in this paper.

5.2 Controlled scenario results

First, we conducted an assessment of our DDS methods (LD-DDS, TF-DDS, and CVR-DDS) by analyzing their performance in a controlled scenario, where domain labels were readily available. The purpose was to study to which extent the DSS techniques proposed were able to correctly classify development sentences according to some common feature, as for instance domain, by providing a test set belonging to that specific domain.

We resorted to the domain adaptation task from the Johns Hopkins Summer Workshop 2012 (Carpuat et al., 2012), where the task was to adapt French→English models. The training corpus provided originated in the parliamentary domain (Canadian Hansards). Development and test corpora included the medical domain (referred to as EMEA), the general news domain (NEWS), the press domain (PRESS), and the subtitle domain (SUBS). Statistics are provided in Table 1.

In this scenario, the development data extracted by our DDS techniques was obtained from a set where all four domain-specific development sets were merged. The *baseline* system was tuned on the Hansards development data, and the *in-domain* system was tuned on the domain-specific development data of each domain, respectively.

[1] http://radimrehurek.com/gensim/doc2vec

Table 1: Corpora used in the controlled scenario. (Dev-in) is the in-domain development set, (Test) is the evaluation set, (Training) is the training corpus and (Dev-bsln) is the baseline development set. M stands for millions and k thousands of elements; $|S|$ stands for number of sentences and $|V|$ for vocabulary size.

		EMEA		**NEWS**		**PRESS**		**SUBS**			**HANSARD**	
		$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$		$\|S\|$	$\|V\|$
EN	Dev-in	2022	2285	2043	3682	1990	4232	2972	1755	Training	8.1M	186.6k
FR			2563		3828		4583		1879			191.5k
EN	Test	2045	2061	2489	4404	1982	4259	3306	1980	Dev-bsln	1367	24.1k
FR			2274		4759		4551		2032			24.9k

Table 2: Precision, recall and F_1 scores for LD-DDS, TF-DDS and CVR-DDS in the controlled scenario.

		EN-FR			**FR-EN**		
Domain	System	Precision	Recall	F_1	Precision	Recall	F_1
EMEA	LD-DDS	0.35	0.33	0.34	0.37	0.32	0.34
	TF-DDS	0.16	0.32	0.21	0.16	0.32	0.21
	CVR-DDS	**0.64**	**0.47**	**0.54**	**0.74**	**0.45**	**0.56**
NEWS	LD-DDS	0.10	0.12	0.11	0.08	0.12	0.09
	TF-DDS	0.24	0.28	0.25	**0.25**	**0.60**	**0.35**
	CVR-DDS	**0.16**	**0.53**	**0.25**	0.17	0.54	0.25
PRESS	LD-DDS	0.01	0.01	0.01	0.01	0.02	0.02
	TF-DDS	0.32	0.46	0.38	0.21	0.60	0.31
	CVR-DDS	**0.38**	**0.52**	**0.47**	**0.36**	**0.47**	**0.41**
SUBS	LD-DDS	0.77	0.39	0.51	**0.81**	**0.43**	**0.56**
	TF-DDS	0.74	0.38	0.50	0.38	0.43	0.39
	CVR-DDS	**0.79**	**0.39**	**0.52**	0.74	0.39	0.51
Total	LD-DDS	0.24	0.46	0.31	0.26	0.27	0.27
	TF-DDS	0.35	0.32	0.33	0.24	0.46	0.32
	CVR-DDS	**0.37**	**0.46**	**0.41**	**0.37**	**0.45**	**0.40**

5.2.1 Precision, Recall and F_1-score

We analysed the ability of our DDS methods to recover the domain labels by providing the corresponding test set. We measured precision, recall and the F_1 measureResults are shown in Table 2, where the last row, *total*, shows precision, recall and F_1 across all domains in a 4-class confusion matrix (i.e., not the average). Several things should be noted:

- Selecting sentences using CVR-DDS obtained significantly better results than TF-DDS and LD-DDS, except for SUBS, where all methods obtained very similar results.
- The best classification quality was obtained in SUBS domain. We understand that this is because this domain has the largest test corpus, and hence yields better estimations.
- In the case of NEWS, our DDS methods obtained the worst values of precision and recall, which implies that they were not able to retrieve the correct development sentences. This seems to signal that it is not an adequate corpus for research on adaptation, as already observed in related work (Haddow and Koehn, 2012; Irvine et al., 2013).
- Finally, the results obtained for the three different methods are coherent across different language pairs (EN-FR and FR-EN).

Note that the result of LD-DDS depends on threshold τ. In Table 2 we only reported the best results obtained, which might slightly bias the results in favour of LD-DDS. However, given that LD-DDS is even so not the best DDS technique (neither in terms of classification metrics, nor in terms of translation quality), we report these results for the sake of assessing its potential.

5.2.2 SMT results

Once the quality of the selected development corpus was analysed, we now pursue establish to which extent classification metrics relate to translation quality, measuring the performance of the DDS methods in terms of BLEU (Table 3). Results with METEOR and TER presented similar

Table 3: Translation results in the controlled scenario. $|S|$ denotes number of sentences.

		EMEA		NEWS		PRESS		SUBS	
	System	$\|S\|$	BLEU	$\|S\|$	BLEU	$\|S\|$	BLEU	$\|S\|$	BLEU
EN-FR	*baseline*	1367	22.9	1367	21.4	1367	21.9	1367	16.6
	in-domain	1784	24.8	1467	23.9	1255	23.9	2940	18.3
	LD-DDS	1657	24.0	1772	22.5	2225	20.9	1568	18.2
	TF-DDS	1778	22.9	1718	23.5	1832	21.6	1543	18.0
	CVR-DDS	1295	24.8	3592	23.7	1724	23.8	1436	18.4
	oracle	1842	26.7	1782	24.7	1227	24.6	3281	19.1
FR-EN	*baseline*	1367	22.6	1367	21.5	1367	20.8	1367	12.3
	in-domain	1784	23.8	1467	23.0	1255	21.1	2940	18.9
	LD-DDS	1532	20.2	2418	20.6	2218	17.1	1549	18.5
	TF-DDS	3550	23.9	3563	22.6	3589	20.2	3496	14.9
	CVR-DDS	1067	24.4	4254	22.7	3754	20.9	1543	18.6
	oracle	1842	26.1	1782	23.6	1227	22.0	3281	19.5

conclusions and are omitted in this case for clarity purposes. Several conclusions can be drawn:

- All DDS methods are mostly able to improve over *baseline* across the different domains and language pairs. This seems reasonable, given that the *baseline* results were obtained using an out-of-domain development corpus for tuning purposes.
- CVR-DDS yields better translation quality than LD-DDS and TF-DDS. This seems to signal that CVR-DDS achieves a better representation of the sentences involved. However, results involving the SUBS domain yield very similar results across all three DDS methods.
- Lastly, translation quality results between CVR-DDS and *in-domain* are not significantly different. We understand that this is important since it proves the utility of our development DS method, which is able to recover a development set which is at least as well-suited for the task as the development set originally designed for that task.

5.3 Real scenario results

After analyzing the behaviour of our DDS techniques in a controlled scenario, we pursued to evaluate them in a real-world task, where no development set was readily available. For this purpose, we confronted the system with a set of sentences obtained from a real e-commerce.

For this purpose, we gathered the data from one of our customers, *Cachitos de Plata*[2], where no appropriate development set was readily available. As training data, we explored the use of three different corpora available in the Workshop on Statistical Machine Translation [3] (WMT): *1)* The Europarl (EURO) corpus, which is composed of translations of the proceedings of the European parliament; *2)* The United Nations (UN) corpus, which consists of official records and other documents of the United Nations belonging to the public domain; *3)* The Common Crawl corpus (COMMON) which was collected from web sources. Statistics of these corpora are provided in Table 4. In this case, our DDS methods were set to sample from the pool of development data available from the different years of the WMT task (*Dev* row in Table 4), and the *baseline* system was tuned according to the 2015 development data (*Dev-bsln*).

In this case, and given that no in-domain development set is available, we also considered random sampling a set of sentences from the available pool of data, in addition to *baseline* and *oracle*. We will refer to this baseline as *random*. Here, 2500 sentences were randomly sampled from the available pool of development data, without repetition. The results reported show the average of 5 repetitions of the sampling, where confidence intervals were never greater than 0.2 points (in the corresponding translation quality metric).

Results in Table 5 show the results in terms of BLEU, METEOR and TER, and development set size. In this case, we omitted both LD-DDS and TF-DDS for clarity purposes and because the results were consistent with those reported in Section 5.2. We also omitted the results obtained when using Europarl as training set, given that BLEU

[2] `http://www.cachitosdeplata.com`. In case of acceptance, this data set will be published free for research purposes, for the purpose of replicability and further research.

[3] `http://www.statmt.org/wmt16`

Table 4: Corpora main figures for real e-Commerce task. (Dev) is the pool development set, (Test) is the evaluation data, (Training) is the training corpus and (Dev-bsln) is the development corpus. Same abbreviations as in Table1.

		e-Commerce			**EURO**		**UN**		**COMMON**	
		$\|S\|$	$\|V\|$		$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$	$\|S\|$	$\|V\|$
EN ES	Test	886	874 976	Training	1.5M	88.2k 133.7k	11.2M	1.7M 893.2k	1.8M	1.9M 613.8k
EN ES	Dev	16.4k	26.0k 31.7k	Dev-bsln	2600	3691 3925	2600	3691 3925	2600	3691 3925

scores with this corpus were around 9.00 points. Several conclusion can be drawn:

- CVR-DDS achieves consistent improvements over the *baseline* translation quality, in all three metrics considered.
- CVR-DDS achieves consistent improvements over the *random* translation quality, in all three metrics, across both language pairs, and with much fewer sentences. Note that it is typically assumed that such random baseline is very tough to beat in DS and active learning research (Ananthakrishnan et al., 2010; Ambati et al., 2010), and, furthermore, improvements are statistically significant.
- Training with UN and COMMON leads to very different results. We assume this is because COMMON, even though being a smaller corpus, is more related to the domain at hand: the Commoncrawl data is crawled from the web, and in this case we are dealing with web data.

6 Conclusions

In this paper, we have presented different techniques for building a test-specific development corpus, leveraged for optimizing the log-linear weights of the SMT system. We proposed three new development data selection methods: LD-DDS, TF-DDS, and CVR-DDS. We analysed the performance of these methods in a controlled scenario, where domain labels are available, and evaluated the methods in a real translation task where e-commerce data was to be translated, without a development set being readily available. The empirical results show that CVR-DDS, which leverages a continuous vector-space representation of the sentences, is able to improve over baseline translation quality, and provide a development set that leads to similar translation quality as than the one obtained whenever an in-domain development set is readily available. In addition, the results obtained with CVR-DDS consistently and significantly improve over those obtained with a random sampling baseline, across different languages.

In the future, we will further investigate the selection of development corpus, since there is more room for improvements, as reported by the *oracle* setting. We also intend to test our methods on other domains and test data so as to establish their robustness. Finally, we are providing the e-commerce corpus *Cachitos de Plata*, used as test data, free for research purposes.

Acknowledgements: The research leading to these results were partially supported by projects CoMUN-HaT-TIN2015-70924-C2-1-R (MINECO/FEDER) and PROMETEO/2018/004.

References

Ambati, V., Vogel, S., and Carbonell, J. G. (2010). Active learning and crowd-sourcing for machine translation. In *Proc. of the LREC*, pages 2169–2174.

Ananthakrishnan, S., Prasad, R., Stallard, D., and Natarajan, P. (2010). A semi-supervised batch-mode active learning strategy for improved statistical machine translation. In *Proc. of the CoNLL*, pages 126–134.

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proc. of the EMNLP*, pages 355–362.

Carpuat, M., Daumé III, H., Fraser, A., Quirk, C., Braune, F., Clifton, A., et al. (2012). Domain adaptation in machine translation: Final report. In *2012 Johns Hopkins summer workshop final report*.

Chen, B. and Huang, F. (2016). Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proc. of the CoNLL*, pages 314–324.

Chen, B., Kuhn, R., Foster, G., Cherry, C., and Huang, F. (2016). Bilingual methods for adap-

Table 5: Translation results for real e-commerce scenario.

		EN-ES				ES-EN			
Training	System	$\|S\|$	BLEU	METEOR	TER	$\|S\|$	BLEU	METEOR	TER
UN	bsln	2600	13.8	42.2	67.3	2600	17.4	24.9	60.8
	random	2500	12.5	40.9	64.7	2500	18.2	27.4	60.9
	CVR-DDS	1681	15.5	42.8	64.4	1750	18.6	27.9	60.2
	oracle	886	19.3	45.6	58.3	886	21.0	28.8	58.0
COMMON	bsln	2600	21.3	49.1	57.0	2600	24.1	32.9	52.7
	random	2500	21.9	49.7	56.6	2500	22.1	33.1	52.2
	CVR-DDS	2346	22.8	50.6	56.5	1704	25.6	34.4	51.3
	oracle	886	31.1	55.7	53.3	886	33.0	37.4	43.5

tive training data selection for machine translation. In *Proc. of the AMTA*, pages 93–103.

Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proc. of the NAACL*, pages 427–436.

Chinea-Rios, M., Sanchis-Trilles, G., and Casacuberta, F. (2016). Bilingual data selection using a continuous vector-space representation. In *Proc. of the SPR+SSPR*, pages 95–106.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv e-prints*.

Duh, K., Neubig, G., Sudoh, K., and Tsukada, H. (2013). Adaptation data selection using neural language models: Experiments in machine translation. In *Proc. of the ACL*, pages 678–683.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proc. of the ICML*, pages 513–520.

Haddow, B. and Koehn, P. (2012). Analysing the effect of out-of-domain data on smt systems. In *Proc. of the WMT*, pages 422–432.

Hildebrand, A. S., Eck, M., Vogel, S., and Waibel, A. (2005). Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proc. of the EAMT*, pages 133–142.

Hinton, G. E. (1986). Learning distributed representations of concepts. In *Proc. of the CogSci*, pages 12–24.

Hui, C., Zhao, H., Song, Y., and Lu, B.-L. (2010). An empirical study on development set selection strategy for machine translation learning. In *Proc. of the WMT*, pages 67–71.

Irvine, A., Morgan, J., Carpuat, M., Daumé III, H., and Munteanu, D. (2013). Measuring machine translation errors in new domains. *Transactions of the Association for Computational Linguistics*, 1:429–440.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proc. of the ICASSP*, pages 181–184.

Koehn, P. (2010). *Statistical machine translation*. Cambridge University Press.

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

Lavie, M. D. A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proc. of the ACL*, page 376.

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

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(20-29):707–710.

Li, M., Zhao, Y., Zhang, D., and Zhou, M. (2010). Adaptive development data selection for log-linear model in statistical machine translation. In *Proc. of the ACL*, pages 662–670.

Liu, L., Cao, H., Watanabe, T., Zhao, T., Yu, M., and Zhu, C. (2012). Locally training the log-linear model for smt. In *Proc. of the EMNLP*, pages 402–411.

Lü, Y., Huang, J., and Liu, Q. (2007). Improving statistical machine translation performance

by training data selection and optimization. In *Proc. of the EMNLP*, pages 343–350.

Mansour, S., Wuebker, J., and Ney, H. (2011). Combining translation and language model scoring for domain-specific data filtering. In *Proc. of the IWSLT*, pages 222–229.

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

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proc. of the ACL*, pages 220–224.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proc. of the ACL*, pages 160–167.

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

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proc. of the ACL*, pages 311–318.

Peris, Á., Chinea-Rios, M., and Casacuberta, F. (2016). Neural networks classifier for data selection in statistical machine translation. `arXiv:1612.05555`.

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

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Schwenk, H., Rousseau, A., and Attik, M. (2012). Large, pruned or continuous space language models on a gpu for statistical machine translation. In *Proc. of the NAACL-HLT*, pages 11–19.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proc. of the AMTA*, pages 223–231.

Socher, R., Lin, C. C., Manning, C., and Ng, A. Y. (2011). Parsing natural scenes and natural language with recursive neural networks. In *Proc. of the ICML*, pages 129–136.

Song, X., Specia, L., and Cohn, T. (2014). Data selection for discriminative training in statistical machine translation. In *Proc. of the EAMT*, pages 45–53.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proc. of the ICSLP*, pages 901–904.

Zheng, Z., He, Z., Meng, Y., and Yu, H. (2010). Domain adaptation for statistical machine translation in development corpus selection. In *Proc. of the IUCS*, pages 2–7.