




ARTICLE

Improved feature decay algorithms for statistical machine translation

Alberto Poncelas* , Gideon Maillette de Buy Wenniger  and Andy Way 

ADAPT Centre, Dublin City University, Glasnevin, Dublin 9, Ireland

*Corresponding author. E-mail: alberto.poncelas@adaptcentre.ie

(Received 17 April 2018; revised 18 May 2020; accepted 19 May 2020; first published online 22 September 2020)

Abstract

In machine-learning applications, data selection is of crucial importance if good runtime performance is to be achieved. In a scenario where the test set is accessible when the model is being built, training instances can be selected so they are the most relevant for the test set. Feature Decay Algorithms (FDA) are a technique for data selection that has demonstrated excellent performance in a number of tasks. This method maximizes the diversity of the n -grams in the training set by devaluing those ones that have already been included. We focus on this method to undertake deeper research on how to select better training data instances. We give an overview of FDA and propose improvements in terms of speed and quality. Using German-to-English parallel data, first we create a novel approach that decreases the execution time of FDA when multiple computation units are available. In addition, we obtain improvements on translation quality by extending FDA using information from the parallel corpus that is generally ignored.

Keywords: Machine translation; Data selection; Statistical methods

1. Introduction

In recent years, the amount of data available for language processing has increased significantly. It is now possible to find vast amounts of data for use as training data in Machine Learning. The field of Statistical Machine Translation (SMT) is no exception to this phenomenon. However, it has been shown in Ozdowska and Way (2009), Gascó *et al.* (2012) that having more data does not always lead to better results. In fact, performance can increase by limiting the training data to a smaller but more relevant set. In addition, inducing a general Machine Translation (MT) model from a large set of training instances can be time consuming. This is why the use of data selection techniques has become a common step in the creation of an MT pipeline.

However, reducing the training data to a subset of relevant sentences can be an ambiguous task. The criteria followed for selecting these sentences can range from the broad (e.g. data belonging to a domain) to the more particular (e.g. those similar to the document we want to translate). In this paper, we address the latter problem. We use data selection algorithms that consider the document to be translated, which is assumed to be available at training time, to build MT models that are more adapted to such a test set. We refer to these methods that use the test set as transductive data selection.

The method for building transductive models that we explore in our paper is Feature Decay Algorithms (FDA). Original FDA and its variants (Poncelas, Way, and Toral 2016; Poncelas, Maillette de Buy Wenniger, and Way 2017) are data selection techniques that use the information of the test set to select sentences from a parallel corpus used for training an MT model. Another

characteristic of these methods is that they are context-dependent data selection techniques. This means that selecting new sentences is an iterative process, whereby at each step not only the test set is considered but the information given by previously selected sentences is exploited as well.

While the default configuration of FDA proposed in Biçici and Yuret (2011) has demonstrated excellent performance^a in a number of tasks, there is room for improvement. This paper covers two main points: (i) we propose a faster FDA alternative that benefits from parallelization and (ii) improve the performance of FDA by adding information inferred from the parallel corpora. We provide a number of contributions in this paper, including

- a comparison of transductive data selection techniques;
- a variation of FDA that boosts execution speed;
- a bilingual model of FDA that also considers n -grams in the target language to avoid selecting noisy sentence pairs.
- an evaluation of the techniques presented in the work using Neural Machine Translation (NMT).

Consequently, this paper is divided into different sections addressing each of the issues mentioned above.

First, in Section 2, we briefly describe data selection algorithms and also introduce FDA.

The proposal for boosting the speed of FDA is explained in Section 3. We propose MRFDA, a variant of FDA that allows parallel computation, assuming multiple computation units available. This variant executes faster and obtains better results in some experiments.

A strategy to improve the quality is presented in Section 4. While the features of FDA are extracted from the test set, in the source language, a new proposal is to use phrase pairs from the training set as features and build a bilingual FDA system. Crucially, these phrase pairs are extracted from the training data and selected using only the source side of the test set. This allows us to indirectly leverage translation-equivalence information from the parts of the training data that are relevant to the source of the sentences in the test set.

In Section 5, we apply the techniques presented here to the NMT approaches. Finally, Section 6 concludes and suggests some further research that can be carried out in this area.

2. Related work

The goals of data selection are diverse: remove noise, restrict the amount of training data, select in-domain data, etc. Among the different categories, we highlight the transductive data selection methods (Vapnik 1998), which aim to identify the most relevant data points for a model to predict a new unseen test set. In the MT field, transductive algorithms assume that the document to be translated S_{test} is available at training time so the best subset of sentences for training an MT model (to translate S_{test}) is retrieved (Poncelas 2019).

In this section, we present an overview of different data selection methods (Eetemadi *et al.* 2015). We classify them as nontransductive (if they do not require the information from the document to be translated S_{test} , Section 2.1) or transductive (Section 2.2).

A limited subset of sentences is selected from a set of sentences S , given a sentence-level scoring function that specifies an implicit sentence ranking. Selected sentences are added to a selected pool or labeled data L , which eventually will become the training data for a particular end-task (MT in our case).

^aIn this work, we will use the term “performance” (of FDA) to indicate the translation quality obtained by the models trained with sentences retrieved by FDA.

2.1 Nontransductive data selection methods

Nontransductive data selection methods are the most common. In this section, we provide a list of methods that extract the most relevant subset of sentences from a set of parallel sentences without using the information of the test set.

- *Length-based Functions*: Many approaches remove noisy data by comparing the length difference (Taghipour *et al.* 2010) or the length proportion (Khadivi and Ney 2005) between the source-side and target-side sentences.
- *Alignment-based Functions*: Taghipour *et al.* (2010) use sentence-alignment entropy to remove noisy data from the training set. Parallel sentences with relatively high entropy in comparison to the rest of the corpus are considered unreliable and are removed.
- *Language Model-Based Functions*: Moore and Lewis (2010) propose to use an in-domain language model LM_I and an out-of-domain language model LM_O to obtain sentences that are closer to in-domain data. They therefore define the cross-entropy difference (CED) as $H_I(s) - H_O(s)$ where $H_d(s)$ is the cross-entropy of the sentence s according to a language model LM_d in a domain d . Axelrod, He, and Gao (2011) extend the equation of CED by using language models in both the source-side and target-side languages, defining the bilingual CED as $(H_{I-src}(s) - H_{O-src}(s)) + (H_{I-trg}(s) - H_{O-trg}(s))$. Additionally, Hoang and Simaan (2014) propose an iterative algorithm based on Expectation-Maximization to estimate the probability of a sentence pair, from a mix-domain corpus, to be in- or out-domain.
- *N-gram Coverage*: In Eck, Vogel, and Waibel (2005a), the sentences that contain n -grams which are not in L are rewarded.
- *TF-IDF Coverage*: The proposal of Eck, Vogel, and Waibel (2005b) is to retrieve sentences that are the most different to selected pool L based on TF-IDF (Salton and Yang 1973) distance. Those sentences that differ the most to the selected pool L are the best candidates to be added to L as they are the most informative.
- *DWDS*: Density Weighted Diversity Sampling (DWDS) (Ambati, Vogel, and Carbonell 2011) scores a sentence based on: (i) how the words are distributed in both the selected and candidate pool and (ii) how many words are not already in the selected pool.
- *Log-probability Ratios*: Haffari, Roy, and Sarkar (2009) propose to select sentences that are common (high probability) in the candidate pool and rare in the selected pool.
- *Perplexity Ratios*: A similar approach to *Log-probability Ratios* method is proposed by Mandal *et al.* (2008), in which sentences are selected based on the perplexity ratio between candidate and selected pool.

2.2 Transductive data selection

In this section, we explain data selection algorithms that use the document to be translated to retrieve sentences (as FDA is a method central to this paper, we explain it in Section 2.3). We describe two methods: *Edit distance* methods, which consider the sentences in the test set individually (sentence-wise method) and *Infrequent n-gram Recover*, which, like FDA, consider the test set as a whole (document-wise method).

Sentence similarity. These methods retrieve sentences based on how similar the sentences in S are compared to a sentence s_{test} from the test set (Wang *et al.* 2014). For every sentence in the test set, the most similar sentences from the training data are retrieved.

Hildebrand *et al.* (2005) propose *TF-IDF distance*; more precisely, they use the cosine value between TF-IDF (Salton and Yang 1973) vectors as distance metric. For each sentence $s_{test} \in S_{test}$ in the test set, the top- $\frac{N}{|S_{test}|}$ sentences from S are selected. Although they are aware that the resulting set may contain duplicates, in their experiments they show that models in which duplicate sentences have been removed achieve slightly worse results.

Infrequent n-gram Recovery. Parcheta, Sanchis-Trilles, and Casacuberta (2018) propose extracting those sentences containing n -grams from the test set that are considered infrequent (Gascó et al. 2012), using simple counting over the candidate data S together with a cutoff maximum frequency. Consequently, frequent words such as stop words are ignored. A sentence s is scored according to number of infrequent n -grams shared with the set of sentences S_{test} of the test set. It is computed as in Equation (1):

$$score(s, L) = \sum_{ngr \in s} C_{\{s\}}(ngr) * \max(0, t - C_{S_I+L}(ngr)) \quad (1)$$

where $C_{\{s\}}(ngr)$ is the count of ngr in the sentence $s \in S$ to be scored, $C_{S_I+L}(ngr)$ is the count of ngr in the selected data L and an in-domain set S_I used for initialization, and t is the *infrequent n-grams cutoff frequency*, that is the maximum number of occurrences up to which an n -gram is considered infrequent. If the number of occurrences is above the threshold t , then ngr is considered frequent n -gram and is ignored (the component $\max(0, t - C_{S_I+L}(ngr))$ is 0) and not considered for scoring the sentence.

2.3 Feature Decay Algorithms

FDA (Biçici and Yuret 2011, 2015; Biçici, Liu, and Way 2015) is a method that tries to maximize the coverage of the sentences to be translated. It uses coverage by phrases of the known source-side n -grams of the test-set as an estimator of coverage of the required target-side, which is unknown. Furthermore, it assigns source-side test n -grams a value, that decreases the more the n -grams have already been included in earlier selected sentences. This ensures sufficient translation examples will be extracted for all relevant test set source n -grams.

Note that in this section, we are using the generic notation of f as feature, even if in this paper (and in the original work of Biçici and Yuret 2011) the only features used are n -grams from the test set.

The selected features are then extracted from the training set sentences. The score of a sentence s is the normalized sum of the value of its features. At each step, the sentence with the highest score is selected. Then the values of the features of the selected sentence are decreased as in (2):

$$decay(f, L) = init(f) \frac{d^{C_L(f)}}{(1 + C_L(f))^c} \quad (2)$$

where L is the selected pool and $C_L(f)$ is the count of the feature f in L . As the initialization function $init(f)$, Biçici and Yuret (2011) propose to use either 1 or the inverted frequency $\log(|S|/C_S(f))$. The variables d and c are input parameters: the decay factor d is in the range (0, 1) with a default value of 0.5, and the decay exponent c is in the range $[0, \infty)$ with a default value of 0.0.

As previously mentioned, the score of a sentence is the normalized sum of the values of the features, and it is computed as in Equation (3):

$$score(s, L) = \frac{\sum_{f \in F_s} init(f) d^{C_L(f)} (1 + C_L(f))^{-c}}{length(s)^{len_exp}} \quad (3)$$

where len_exp is a parameter that indicates the amount of influence of the sentence length on the score. Higher values of len_exp cause FDA to retrieve shorter sentences.

Table 1. Statistics of the data sets. $|S|$ is the number sentences, $|W|$ the number of words, and $|V|$ the size of the vocabulary

	$ S $	$ W _{(DE)}$	$ W _{(EM)}$	$ V _{(DE)}$	$ V _{(EM)}$
Training set	4.48M	110M	116M	2M	971K
Development set	5000	127K	129K	23K	16K
Test set	2169	44K	46.8K	9.9K	7.8K

While the decay function in Equation (2) was introduced in Biçici and Yuret (2015), the default values of these parameters have been set so it is equivalent to Equation (4), the decay function originally proposed in Biçici and Yuret (2011).

$$\text{decay}(f, L) = 0.5^{C_L(f)} \quad (4)$$

The score of a sentence s at a particular iteration is the sum of the values of $C_L(f)$ of the features present in s , normalized by the length of s . The score of a sentence, using default configuration in Equation (4) is computed as in (5):

$$\text{score}(s, L) = \frac{\sum_{f \in F_s} 0.5^{C_L(f)}}{\text{length}(s)^{\text{len_exp}}} \quad (5)$$

where F_s is the set of features present in s .

FDA is a context-dependent function (Eetemadi *et al.* 2015) as it uses the information of selected pool L when considering a new sentence to be added and has demonstrated good results to retrieve in-domain data (Silva *et al.* 2018). It is related to the work of Kirchhoff and Bilmes (2014) where the problem is examined from a submodular optimization perspective.

In each iteration, the sentence with the highest score is transferred to L , which causes the score of the rest of the candidate sentences (sharing the same n -grams with the test set) to decrease, as they depend on L .

2.4 Comparison of FDA to other transductive algorithms

In Table 2, we compare FDA against the transductive algorithms. In the table, we show the performance of the models built with: the complete training data (*AllData* column), sentences obtained using TF-IDF Transductive Method (*TF-IDF* column), Infrequent n -gram Recovery, and FDA.

The data sets used in the experiments are based on those used in Biçici *et al.* (2015) (cf. Table 1):

- *Languages*: German-to-English.
- *Training set*: The training data provided in the WMT 2015 (Bojar *et al.* 2015) translation task setting a maximum sentence length of 126 words (4.5M sentence pairs, 225M words).
- *Development set*: We use 5K randomly sampled sentences from development sets from previous years of WMT for tuning.
- *Language Model*: 8-gram Language Model (LM) built on the target language side of the selected data via the KenLM toolkit (Heafield 2011) using Kneser–Ney smoothing (Kneser and Ney 1995).
- *Test set*: Documents provided in the WMT 2015 Translation Task.

We train SMT systems on the selected data using the Moses toolkit (Koehn *et al.* 2007) with the standard features and using GIZA++ (Och and Ney 2003) for word alignment.

We include several evaluation metrics: BLEU (Papineni *et al.* 2002), NIST (Doddington 2002), TER (Snover *et al.* 2006), METEOR (Banerjee and Lavie 2005), and CHRF (Popovic 2015). These

Table 2. Comparison of transductive algorithms

	AllData	TF-IDF	Infrequent n -gram Recovery	FDA
100K lines				
BLEU	18.21	18.06	19.31*	19.42*
TER	66.88	63.30	63.03*	62.26*
METEOR	26.01	25.85	26.68*	26.76*
CHRF3	47.15	45.91	47.22	47.06
200K lines				
BLEU	18.21	18.64*	19.64*	19.63*
TER	66.88	62.38*	63.14*	63.27*
METEOR	26.01	26.60*	27.12*	27.08*
CHRF3	47.15	46.70	48.09	48.01
500K lines				
BLEU	18.21	18.61*	–	18.83*
TER	66.88	62.96*	–	64.44*
METEOR	26.01	26.68*	–	26.58*
CHRF3	47.15	46.92	–	47.68

scores give an estimation of the quality of the output of the experiments compared to a human-translated reference. In general, the higher the score is, the better the translation is estimated to be (except for TER, which is a translation error measure and so lower is better).

We show the scores as the mean of 4 MERT (Och 2003) executions. Results in bold indicate an improvement over the baseline (in most cases, default FDA). We have also computed the statistical significance, indicated with an asterisk, at level $p = 0.01$ using multeval (Clark *et al.* 2011) for BLEU, TER, and METEOR when compared to the baseline at level $p = 0.01$ using Bootstrap Resampling (Koehn 2004).

For comparability, it is important to use the same sized training sets across these experiments. For this reason, in the rest of the paper, we have chosen to use 100, 200, and 500K training sentences. Note that the different methods executed in this work retrieve a similar amount of words for datasets of the same size: 4–5M words for 100K sentences; 9–11M words for 200K sentences; and 25–28M words for 500K sentences.

For the Infrequent n -gram Recovery configuration we also use 3-grams. In the work of Parcheta *et al.* (2018), they do not set a default t in Equation (1), but execute several runs with the values of t ranging from 10 to 20. We also execute the method for the values 10, 20, 40, and 80 (the value is multiplied by 2 in each run). Executing the method with $t = 160$ causes the execution time to exceed 48 hours, so we use this as a stopping criterion. The higher the value of t (*infrequent n -grams cutoff frequency*), the more sentences are retrieved (as the decision for considering an n -gram infrequent is less strict). However, with the largest value of t executed ($t = 80$), the number of sentences retrieved is 229,913 as the remaining sentences have a score of 0.0. Therefore, a comparison of a model built with 500K sentences retrieved by Infrequent n -gram Recovery with $t = 80$ is not possible.

In Table 2, we indicate in bold those scores that outperform the baseline (the model built with all data) and indicate with an asterisk if these improvements are statistically significant at $p = 0.01$. The performance of the models built with data retrieved by transductive algorithms performs

better than using the complete training set. If we compare FDA to the rest of algorithms, we see that it performs better than *TF-IDF*. Although we do not indicate in the table, the performance of FDA is statistically significantly better (at $p = 0.01$) than TF-IDF models for all experiments according to METEOR and TER scores (and BLEU for 100K lines and 200K lines models). In contrast, the performance of the Infrequent n -gram recovery method is comparable to FDA. We computed the statistical significance and see that, at $p = 0.01$, none of the scores are statistically significantly better than FDA.

3. Improving speed of Feature Decay Algorithms: MapReduce FDA

The data selection techniques we are exploring in this work are context-dependent functions. These techniques work iteratively extracting sentences from the candidate pool and adding them to a set we call the *selected pool*. At each iteration, the information in the selected pool is used to decide which sentences will be selected next. Once the selected pool has the desired amount of sentences, we can use that set as training data.

Seeing that the criteria for selecting a sentence are dependent on the selected pool, that is dependent on all the previous iterations, it is difficult to parallelize. For this reason, in this work, we want to analyze the dependencies between the sentences so a context-dependent technique such as FDA can be executed in parallel when multiple computation units are available. This will imply the computation resources will not be underused and so the algorithm will be faster.

3.1 ParFDA

ParFDA (Biçici *et al.* 2015) tries to parallelize FDA by executing several independent FDA processes on partitions of the training data. Then the resulting selected data are merged. However, it is only an approximation since some dependencies between the sentences are lost. The strength of FDA is to use the information of the previous selected sentences to choose the next sentence. If the parallel corpus is split into different parts, the dependencies between sentences in those different parts are lost and so each FDA process does not have complete information of the selected pool to decide which sentence to select next. This can hurt performance especially if features are not uniformly distributed over sub-parts of the corpus.

3.2 MapReduce Feature Decay Algorithms

The proposal in this work is to use the MapReduce approach to perform a faster “FDA-like” data selection. MapReduce (Dean and Ghemawat 2008) is a programming model for processing large datasets. It is an abstraction that allows the data to be processed in parallel. It organizes the data as a set of key value (k, v) pairs which can be distributed among the different computation units. For processing the data, we can execute different instantiations of the map and reduce functions:

- map: $(k, v) \rightarrow (k_I, v_I)$
Converts a pair (k, v) into a pair with an intermediate key and an intermediate value (k_I, v_I) . Every intermediate key value pair is independent of the rest, so different pairs can be assigned to different computation units, and be processed in parallel.
- reduce: $(k_I, v_I) \rightarrow (k_I, [\dots, v_I, \dots])$
Shuffles the (k_I, v_I) pairs and groups them by the key k_I .

The map and reduce functions can be concatenated in a pipeline to perform more complex tasks.

The results are not strictly equivalent to FDA as we make some assumptions that simplify the model. The first approximation is to pretend that a sentence can have only one occurrence of each

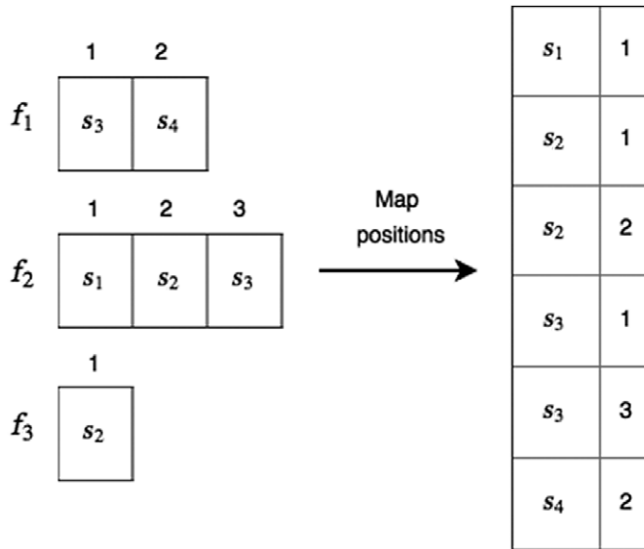


Figure 1. Map stage.

feature. Then, the value of $C_L(f)$ at the iteration when a sentence s is selected can be redefined as $pos(s, f)$, which indicates the position that sentence s has in the ranking of selected pool, conditioned to sentences containing the feature f . The FDA scoring method with default parameters in (5) of a sentence s can be formulated as Equation (6):

$$score(s) = \sum_{f \in F_s} \frac{0.5^{pos(s,f)}}{length(s)} \tag{6}$$

The computation of $score(s)$ in (6) can be divided into two tasks: (i) compute the values of $pos(s, f)$ for every feature in the sentence and (ii) add all the terms of the summation. In our work, the map stage will address the first task and is explained in Section 3.2.1. In the reduce stage (Section 3.2.2), the second task is performed and the top-N sentences are retrieved.

3.2.1 Map stage

The map stage aims to build a set of tuples $(s, pos(s, f))$ for every sentence s and feature f . Initially, a data structure (queried from an inverted index data) is built where each feature f_i maps the list of sentences where f_i is present. On the left side of Figure 1, we can see an example: feature f_1 is present in sentences s_3 , and s_4 ; feature f_2 is present in s_1 , s_2 , and s_3 ; and feature f_3 is present in s_2 .

Structuring the sentences as presented on the left side of Figure 1 (i.e. having a list of sentences for each feature) allows us to extract the tuples $(s, pos(s, f))$ in parallel. After ordering the sentences within a list (this will be explained later), the information can be organized as tuples $(s, f, pos(s, f))$. The name of the feature itself is not relevant to compute the score of Equation (6), so only pairs $(s, pos(s, f))$ are constructed.

At the end of the map stage (right side of Figure 1), the union of all tuples is retrieved. On the right side of Figure 1, we see the pair $(s_1, 1)$ as s_1 holds the position 1 in the list of the f_2 (the list on the left side of Figure 1). The sentence s_2 is in the list of f_2 (in position 2) and in the list of f_3 (in position 1), so we can find the pairs $(s_2, 2)$ and $(s_2, 1)$.

For ordering the sentences, two things must be considered in a sentence: the value $C_L(f)$ of the features and the length of the sentence. Longer sentences are more likely to contain more features,

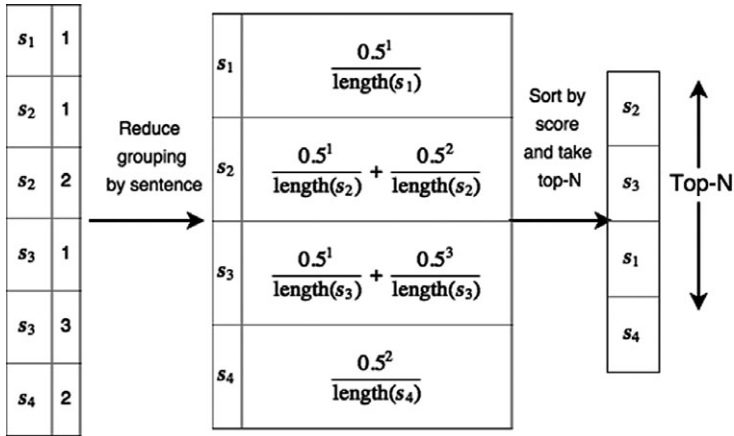


Figure 2. Reduce stage.

but the ratio of features per word may be smaller. It is necessary to find a balance between the value of the features and the length of the sentence.

Let assume two sentences s_1 and s_2 where the set of features present in s_2 is a subset of those present in s_1 ($|F_{s_2}| \subset |F_{s_1}|$). If both sentences were the same length, sentence s_1 would more valuable as it contains more features. It is better to select s_2 only when the length is much smaller. When should it be equally likely to be selected? In other words, when does the equality of scores occur as in Equation (7):

$$\sum_{f \in F_{s_1}} \frac{0.5^{\text{pos}(s,f)}}{\text{length}(s_1)} = \sum_{f \in F_{s_2}} \frac{0.5^{\text{pos}(s,f)}}{\text{length}(s_2)} \tag{7}$$

Considering the extreme case (and thus making another approximation of FDA) where all the positions are 1, we have the following approximation in Equation (8):

$$\frac{|F_{s_1}| * 0.5^1}{\text{length}(s_1)} = \frac{|F_{s_2}| * 0.5^1}{\text{length}(s_2)} \tag{8}$$

where $|F_{s_i}|$ means the number of features present in the sentence s_i . Therefore, the proportion of length when they are equally valuable is that in Equation (9):

$$\frac{\text{length}(s_1)}{|F_{s_1}|} = \frac{\text{length}(s_2)}{|F_{s_2}|} \tag{9}$$

Accordingly, we use $\text{length}(s)/|F_s|$ as a criterion to order (increasingly) the sentences.

3.2.2 Reduce stage

In the reduce stage, the pairs of $(s, \text{pos}(s, f))$ produced in the map stage are collected, grouping them by sentence. This means that each sentence maps a list of $\text{pos}(s, f)$, allowing us to compute the score of the sentence as in Equation (6). We show an example in the middle part of Figure 2. In the second row, a sentence such as s_2 collects the positions from the tuples $(s_2, 1)$ and $(s_2, 2)$ to compute $\frac{0.5^1}{\text{length}(s_2)} + \frac{0.5^2}{\text{length}(s_2)}$.

The list of tuples $(s, \text{score}(s))$ can be sorted by score and the top-N sentences are retrieved as the selected data.

Table 3. Percentage of common lines in FDA and ParFDA and FDA and MRFDA

Selected lines	Recall (3-grams)		Recall (7-grams)	
	ParFDA (%)	MRFDA (%)	ParFDA (%)	MRFDA (%)
100K	40	42	41	45
200K	51	51	52	54
300K	60	56	60	58
500K	72	63	73	64
800K	82	69	82	70

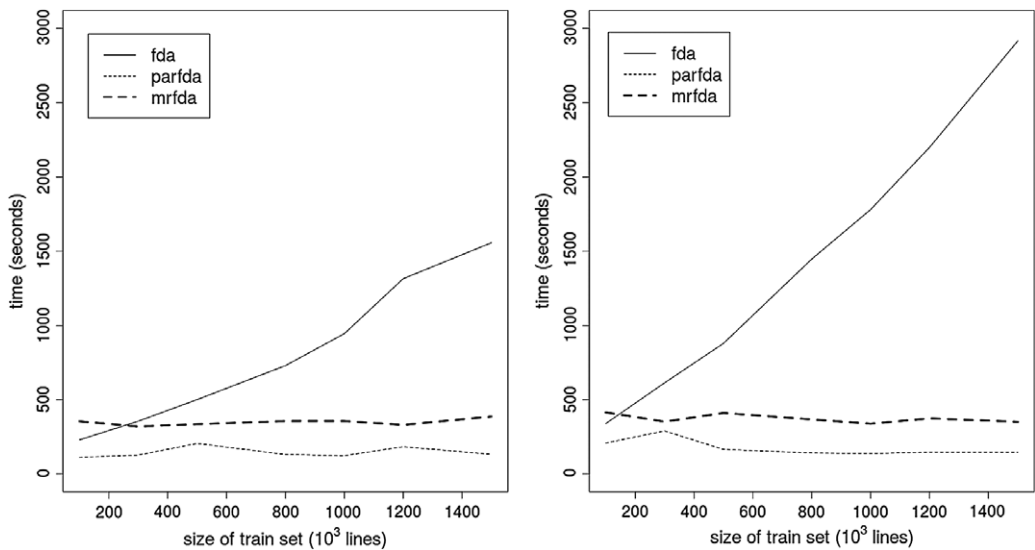


Figure 3. Time of execution of FDA, ParFDA, and MRFDA for different training sizes using 3-grams (left) and 7-grams (right) as features.

3.3 Experiments

In order to evaluate the performance of our proposal when there are multiple computation units available, we execute the three methods: FDA, ParFDA and our proposal, and FDA using MapReduce (MRFDA). The experiments were executed on a machine with 32 CPUs, so ParFDA was run by splitting the training set into 32 parts and executing FDA independently on each part.

In Figure 3, we present the execution times of different algorithms when selecting sets of sentences of different sizes. It shows that the execution time of FDA depends on the order of the *n*-gram used as feature and the size of the selected set. However, both ParFDA and MRFDA have execution times that are more constant, regardless the size of features or the size of the selected set of sentence pairs. We observe that the fastest algorithm is ParFDA. However, as we said in Section 3.1, ParFDA is only an approximation.

In order to understand how similar the output of MRFDA is to FDA, we present in Table 3, the percentage of lines ParFDA and MRFDA shares with FDA. We observe that for smaller sets of retrieved data the intersection is small (between 40% and 45%), but it rapidly increases to between 69–82% when more sentences are selected.

Table 4. Results of the executions of FDA, ParFDA, and MRFDA experiments using 3-grams as features

	FDA	ParFDA	MRFDA
100K lines			
BLEU	19.42	18.80	19.59*
TER	62.26	62.38	61.92**
METEOR	26.76	26.15	27.04**
CHRF3	47.06	46.58	47.39
200K lines			
BLEU	19.63	18.88	19.53*
TER	63.27	63.02	63.87
METEOR	27.08	26.35	27.02*
CHRF3	48.01	47.00	47.87
500K lines			
BLEU	18.83	18.26	19.15**
TER	64.44	64.66	65.44
METEOR	26.50	26.32	26.67**
CHRF3	47.68	47.31	47.91

We also show the evaluation metrics for FDA, ParFDA, and our work (MRFDA) using features of size 3 (Table 4) and size 7 (Table 5). Numbers in bold indicate that MRFDA has outperformed FDA for that particular metric. In addition, we indicate for BLEU, TER, and METEOR the statistically significant improvements at level $p = 0.01$ of MRFDA over ParFDA (one asterisk) and over both ParFDA and FDA (two asterisks).

In Tables 4 and 5, we see that despite being an approximation of FDA, the results obtained are comparable to those obtained for FDA. In most of the experiments, we can see evaluation scores where MRFDA improves over FDA (numbers in bold in the last columns). In addition, especially when selecting larger amounts of lines, the improvements are statistically significantly better, the results of both experiments (using 3-grams and 7-grams) agree that when 500K lines were selected, the improvements are statistically significant at level $p = 0.01$. For those experiments where FDA has better scores than MRFDA, we observe that none of them are statistically significant.

ParFDA obtains the lowest results. This can be understood from the fact that ParFDA splits the training set randomly into N parts, for some N , and performs selection on each part independently of the rest. Such independent selection over subsets of sentences is suboptimal for an algorithm that strives above all to select an adequate number of sentences for each n -gram across all sentences. None of the scores of column ParFDA are better than those of FDA or MRFDA. When comparing MRFDA to ParFDA, we observe in that in all the experiments, for at least two evaluation metrics, the improvements for MRFDA are statistically significantly better at level $p = 0.01$ than ParFDA.

Using higher order n -grams as features boosts translation quality. This effect is present when comparing Tables 4 and 5 for FDA. ParFDA, in contrast, has a random component so this effect does not always apply. In the experiment of ParFDA where 100K lines were selected, we see a decrease in the evaluation scores in Table 5. In MRFDA, while increasing the order of the n -gram of the features is not beneficial for smaller sizes of data set (100K lines), it rapidly improves when

Table 5. Results of the executions of FDA, ParFDA, and MRFDA experiments using 7-grams as features

	FDA	ParFDA	MRFDA
100K lines			
BLEU	19.82	18.58	19.21*
TER	61.32	62.49	62.37
METEOR	27.08	25.85	26.74*
CHRF3	47.60	46.13	46.96
200K lines			
BLEU	19.58	19.13	19.65*
TER	63.16	63.50	63.42
METEOR	27.12	26.28	27.14*
CHRF3	48.17	46.84	48.02
500K lines			
BLEU	18.88	18.59	18.97*
TER	64.32	64.27	65.17
METEOR	26.56	26.32	26.67**
CHRF3	47.66	47.25	47.84

increasing the size (200K lines). However, at some point, using both 3-grams and 7-grams sees similar behaviour. When selecting 500K lines, the scores obtained by BLEU and CHRF3 are better using 3-grams, whereas TER scores are better with 7-grams (and METEOR scores remain the same).

4. Retrieving good quality sentences: Bilingual FDA

In this section, we propose an extension of MRFDA in order to achieve higher performance than FDA. Sentences containing translations that are not appropriate for the context of the test set are not suitable for use as training data as they may hurt the performance (for example, depending on the context, the word “light” can mean “not heavy” or “not dark”). For this reason, we are interested in finding phrase pairs that are appropriate for the test set. These phrase pairs may be used as features of FDA instead of just n -grams in the source language. If the phrase pairs are appropriate for the test set, this will provide several improvements: (i) avoid selecting bad quality sentence pairs; (ii) avoid selecting terms that are translated differently in another domain; and (iii) retrieve more occurrences of n -grams that have multiple translations.

In the following section, we first propose a method that retrieves phrase pairs that are estimated to be useful for translating a test set (Section 4.2). Then, in Section 4.3, we use this set of phrase pairs as features of FDA.

4.1 Phrase pairs extraction using the test set

In order to extract good phrase pairs that are appropriate for a test set, we extend the *translation by pattern matching* technique proposed by Lopez (2008). Lopez proposes a more efficient grammar extraction procedure for the scenario where the test set S_{test} is known at extraction time. In

Algorithm 1 Pattern Matching Phrase Extraction algorithm (Lopez, 2008).

for all n -gram $ngr \in S_{test}$ **do**

- (1) Sample a set of parallel sentences, D_{ngr} from the training data S , where ngr is present in the source side of the sentences.
- (2) Use the word alignment information to extract only those phrase pairs where ngr is present in the source side (he calls this method *source-driven phrase extraction*).

end for

Algorithm 2 Ranked Pattern Matching Phrase Extraction algorithm.

for all $s_{test} \in S_{test}$ **do****for all** n -gram $ngr \in s_{test}$ **do**

- (1) Sample a set of parallel sentences, D_{ngr} , where ngr is present in the source side of the sentences.
- (2) Analyse D_{ngr} and rank the sentences according to how suitable they are for translating the sentence s_{test} where ngr was extracted from.
- (3) Select the top- M sentences.
- (4) Extract phrase pairs from the selected M sentences using *source-driven phrase extraction* technique.

end for**end for**

this scenario, Lopez describes how to extract only the specific phrase pairs (given a word-aligned parallel corpus) that are useful for that particular S_{test} instead of building a phrase table with all the possible phrase pairs that can be obtained from the parallel sentences. The algorithm proposed by Lopez loops over all n -grams in the test set and finds appropriate phrase pairs for them (see Algorithm 1).

The method for extracting phrase pairs has been further developed (Callison-Burch, Bannard, and Schroeder 2005; Germann 2014, 2015) so the phrase pairs are indexed using suffixarray (Manber and Myers 1993) for fast retrieval. Once the phrase pairs for all the n -grams have been extracted, the resulting set of phrase pairs are the ones appropriate for S_{test} . Building a phrase table with this set of phrase pairs means that all the entries in the phrase table are candidates for translating S_{test} (because the rest have been ignored).

The proposal of this section is to improve the *translation by pattern matching* technique by extracting the phrase pairs not from all the sentences in the sample D_{ngr} (extracted from the complete training set S) but from the best sentences. In the original work, the phrase pairs are extracted without considering the quality of the sentences or the context of ngr . Since a phrase in the source language may have multiple translations in the target language, not all the sentences are appropriate from which the phrase are to be extracted. In addition there might be sentences in D_{ngr} with low translation quality. For this reason, we propose to perform an analysis on s_{test} and D_{ngr} in order to decide which are the most appropriate parallel sentences.

Our proposal consists of separating the sampling stage into different steps. In Figure 4, we show the complete pipeline.

The proposed metric is based on how much the words in D_{ngr} are related to ngr . We want to compare the observed word distribution within D_{ngr} – which is conditioned on the n -gram ngr being present – against the general unconditioned word distribution for the full parallel text. In

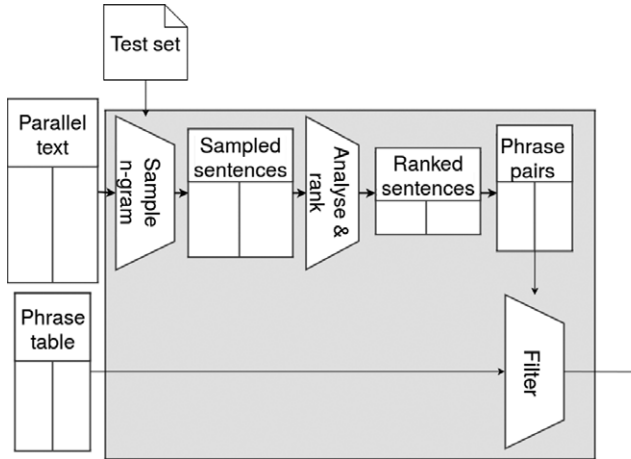


Figure 4. Using the information in the test set to filter phrase pairs.

doing so, we are assuming that if the relative frequency of words in the conditioned distribution of D_{ngr} differs from their relative frequency in the unconditioned distribution, this is caused by the requirement of ngr to be present in D_{ngr} . For every word in D_{ngr} , the difference in relative frequencies for that word between the two distributions is then used to define a notion of dependency between the word and ngr . In order to make this comparison we use a statistic based on Pearson’s chi-squared independence test. The dependency a word w in D_{ngr} has on ngr is calculated as in Equation (10):

$$dep(w, D_{ngr}) = \frac{(O(w, D_{ngr}) - E(w, D_{ngr}))^2}{E(w, D_{ngr})} \tag{10}$$

where $O(w, D_{ngr})$ is $C_{D_{ngr}}(w)$, the observed count of w in the sample D_{ngr} . Note that the count is computed only in one side of the parallel set, that is, if w is a word in the target language, then the count is computed in the target-side only. $E(w, D_{ngr}) = \frac{C_S(w) * |D_{ngr}|}{|S|}$ is the expected count of w in the source side of D_{ngr} if the distribution was the same as in the complete training data S . A higher value of $dep(w, D_{ngr})$ indicates a stronger dependence of the word w on the n -gram ngr .

Once the words (both on the source and target sides) of D_{ngr} have been weighted, we can use them to build quality and similarity metrics.

Ranking Based on Quality. We consider that words on the target side of D_{ngr} that are very dependent on the n -gram in the source side ngr are better candidates to be (part of) the translation of ngr . Sentences containing words with a low dependency on ngr indicate that they may not be useful for translating ngr .

For weighting the sentences, we can perform a normalized sum of the dependencies of the words on the target side of the sentence, as in Equation (11):

$$qual(ngr, \langle s, t \rangle) = \frac{\sum_{w \in t} dep(w, ngr)}{|t|} \tag{11}$$

where $\langle s, t \rangle$ is a pair of source-side and target-side sentence.

Table 6. Comparison of the estimated quality of translation by an SMT system with a phrase table built using all the parallel sentences (*AllData*) and performing *source-driven phrase extraction* extracting random samples of 300 sentences (*NoRanking300*). The results in bold indicate the best score. The asterisk means the result is statistically significant at level $p = 0.01$ when compared with *AllData* experiment

	AllData	NoRanking300	Qual top-10
BLEU	18.21	18.81*	18.74
TER	66.88	63.02*	63.34
METEOR	26.01	26.43*	26.70*
CHRF3	47.15	47.29	47.73

4.2 Phrase extraction analysis

In order to evaluate if the method proposed in Section 4.1 retrieves phrase pairs that are appropriate for a test set, we use them on a phrase table pruning task. We want to investigate if a reduced phrase table, containing only the entries with the phrase pairs retrieved by our method, performs as well as using the complete phrase table.

There are multiple techniques for pruning entries from the phrase table. Previous proposals contemplates removing phrase pairs if they are redundant (Zens, Stanton, and Xu 2012) or using Fisher’s Exact Test to evaluate if they occur by chance (Johnson *et al.* 2007). This is done by analyzing the contingency table of the occurrences of source and target phrases of the phrase table and computing for that the hypergeometric distribution $\frac{\binom{C(\bar{s})}{C(\bar{s},\bar{t})} \binom{N-C(\bar{s})}{N-C(\bar{s},\bar{t})}}{\binom{|PT|}{C(\bar{t})}}$, where $|PT|$ is the size of the phrase table, and $C(\bar{s})$, $C(\bar{t})$, and $C(\bar{s}, \bar{t})$ are the count occurrences of source phrase \bar{s} , target phrase \bar{t} and pair (\bar{s}, \bar{t}) , respectively, in the phrase table PT . These techniques do not make use of the information in the test set.

The goals of the experiments in this section are to explore whether ranking sentences is useful for obtaining fewer but more appropriate phrase pairs than the default *translation by pattern matching* technique. Therefore, the proposed experiments are as follows:

- *NoRanking300*: A random sample of 300 phrase pairs (in their original work, Lopez 2008 concludes that with a random sample of 300 sentences the accuracy plateau is reached).
- *Qual-10*: The top-10 phrase pairs,^b from a sample of 300 pairs, ranked using the quality score measure of Equation (11).

The phrase pairs obtained by each experiments will be used to filter entries from a phrase table built using Moses.

In Table 6, we can see a comparison of *Qual top-10* with the baseline *NoRanking300*, the original system proposed in Lopez (2008). Compared to the baseline system that selects all 300 phrase pairs (column *NoRanking300* in Table 6) with our proposal, we can observe that ranking the sentences has a positive effect on the results.

4.3 BilingualFDA

The phrase pairs obtained in Section 4.2 are useful to build a filtered phrase table to translate a particular test set as shown in Table 6. In this section, we want to demonstrate an extra utility for these phrase pairs to extend FDA and build a BilingualFDA.

^bThe number of phrases was decided by evaluating the filtered phrase table on a development set.

Table 7. Summary of the performance of FDA, MRFDA, and BilingualFDA methods

	FDA	MRFDA	BilingualFDA
100K lines			
BLEU	19.42	19.59	20.26*
TER	62.26	61.92*	61.94*
METEOR	26.76	27.04*	27.96*
CHRF3	47.06	47.39	49.07
200K lines			
BLEU	19.63	19.53	19.84*
TER	63.27	63.87	63.02*
METEOR	27.08	27.02	27.49*
CHRF3	48.01	47.87	48.43
500K lines			
BLEU	18.83	19.15*	19.62*
TER	64.44	65.44	63.21*
METEOR	26.50	26.67*	27.38*
CHRF3	47.68	47.91	48.78

Default FDA uses n -grams in the source side extracted from the test set. For example, one of the features that FDA extracts in the test sentence “die Premierminister Indiens und Japans trafen sich in Tokio.” is the n -gram “in Tokio.” Then, the algorithm will select sentences containing this n -gram in the source side. In BilingualFDA though, the features used are phrase pairs such as (“in Tokio”, “in Tokyo”). This means that it will select sentences containing “in Tokio” in the source side and “in Tokyo” on the target side. The algorithm will avoid selecting incorrectly translated sentences from the training data.

4.3.1 Experiments on BilingualFDA

We again run data selection experiments extracting different sizes of training data (100, 200, and 500K lines) using BilingualFDA. In order to build the BilingualFDA, we extend MRFDA to use phrase pairs as features. The set of phrase pairs used in the experiments are those obtained in the experiment *Qual top-10* in Section 4.2.

In Table 7, we present a summary comparing the algorithms proposed in this work. The MRFDA experiment shows the results obtained in Section 3 using n -grams of size 3 (the same used for default FDA experiments).

The last column shows the results obtained using BilingualFDA. This is the only algorithm that considers the quality of the sentences, and as a result, it performs the best and achieves statistically significant improvements at level $p = 0.01$ for all the scores we have computed (BLEU, TER, and METEOR).

5. Application to NMT

In the following, we explore whether the improvements presented in this paper are also observed with NMT models. The work of van der Wees, Bisazza, and Monz (2017) demonstrated

Table 8. Summary of the performance of FDA, MRFDA, and BilingualFDA methods in NMT

	AllData	FDA	MRFDA	BilingualFDA
100K lines				
BLEU	24.74	19.51	17.43	21.06*
TER	55.25	62.43	64.05	60.08*
METEOR	27.98	24.50	22.86	26.07*
CHRF3	48.95	42.98	39.57	45.27
200K lines				
BLEU	24.74	23.04	22.41	24.29*
TER	55.25	57.88	58.38	56.55*
METEOR	27.98	27.22	26.77	28.57*
CHRF3	48.95	47.27	46.37	49.50
500K lines				
BLEU	24.74	25.17	25.17	25.84*
TER	55.25	56.01	55.99	54.86*
METEOR	27.98	28.86	28.58	29.42*
CHRF3	48.95	49.83	49.33	50.85

that models can be improved by using in-domain data selected using CED. We replicate the experiments presented in Table 7 (we use the same data) but with neural models trained using OpenNMT-py (Klein *et al.* 2017) with the default settings: 2-layer LSTM with 500 hidden units, vocabulary size of 50,000 words for each language, executed for 13 epochs. The results are presented in Table 8.

The work of Poncelas, Maillette de Buy Wenniger, and Way (2018) explores the performance of NMT using different sizes of data selected by FDA. In their work, they show that the performance increases the more data are selected up to 2M sentences. However, the increases in performance when selecting more than 500K sentences are small (below 1% improvement in terms of BLEU). For this reason, in the NMT experiments, we include the comparison between sizes of 100, 200, and 500K sentence pairs, which are same amounts explored for SMT in this paper.

In the table, we observe that, in contrast to SMT, the scores increase the more data are used for training (the scores in subtable *500K lines* are better than those in subtable *200K lines*, which at the same time are better than *100K lines*).

When comparing column-wise the performance of the models, we see that MRFDA do not outperform FDA. The method that achieves the best results is *BilingualFDA* as it obtains statistical significant improvements over FDA at $p = 0.01$ for the models built with different sizes of data. We propose as future work a more fine-grained experiment for exploring the performance using different configurations of each method. For completeness, we also include the table with the result of the NMT baseline model trained in all data (first column of Table 8).

6. Conclusions and future work

In this work, we have explored FDA, a transductive data selection technique, to build SMT models for German to English. Models trained with data retrieved by this method are shown to be superior

to those built with the full training set, as well as models trained with data from other transductive data selection techniques.

First, we have analyzed the factors that have an influence on execution time of FDA. Using high orders of n -grams as features (which forces the method to deal with more features) and retrieving larger data sets are the main factors that cause time duration of FDA to increase. In order to solve that problem and obtain more constant execution times, we presented MRFDA, a parallelized version of FDA that benefits from using multiple CPUs. This new method executes faster than FDA yet obtains comparable performance. Furthermore, the quality achieved by MRFDA is better than with ParFDA, which is the state-of-the-art parallel version of FDA.

The last method considers improving FDA by expanding the features. Default FDA uses only n -grams in the source side as features. We proposed to use pairs of n -grams (in the source and target language) to create BilingualFDA. By preprocessing the test set, it is possible to find those n -grams pairs that are good translation of each other. Using these as features, BilingualFDA can avoid selecting those sentences that potentially hurt performance.

As far as parametrization and application of FDA are concerned, there are some remaining opportunities that we leave for future work: (i) optimal number of sentences (i.e. find a method to retrieve how many sentences should be selected to achieve the best results); (ii) optimal FDA parameters (i.e. explore new ways of estimating more effective default values); and (iii) FDA for tuning (i.e. explore the performance of the models when FDA is used to select sentences for tuning the model).

Alternative executions of FDA that can be further investigated include the use of an approximated translation of the test set (Poncelas, Way, and Sarasola 2018; Poncelas, de Buy Wenniger, and Way 2018) so that the target language is also considered.

Finally, we are interested in further exploring the algorithms explained in this work using NMT, using different configurations (Poncelas *et al.* 2018) or artificial datasets (Poncelas, de Buy Wenniger, and Way 2019a; Poncelas and Way 2019; Soto *et al.* 2020). Even if NMT systems work better with big amounts of data, data selection algorithms are useful to perform the so called “fine-tuning” (Luong and Manning 2015; Freitag and Al-Onaizan 2016), where pre-built system are improved with a small portion of in-domain data. We plan to investigate the performance of this technique using FDA (Poncelas *et al.* 2018; Poncelas, de Buy Wenniger, and Way 2019b) and the methods presented here to improve NMT models trained with large sets of parallel sentences.

Financial support. This research is supported by the ADAPT Centre for Digital Content Technology, funded under the SFI Research Centres Programme (Grant 13/RC/2106). This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 713567.

References

- Ambati V., Vogel S. and Carbonell J.G. (2011). Multi-strategy approaches to active learning for statistical machine translation. In *Proceedings of the 13th Machine Translation Summit*, Xiamen, China. Carnegie Mellon University, pp. 122–129.
- Axelrod A., He X. and Gao J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK. Association for Computational Linguistics, pp. 355–362.
- Banerjee S. and Lavie A. (2005). Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, Michigan. Association for Computational Linguistics, pp. 65–72.
- Biçici E., Liu Q. and Way A. (2015). Parfda for fast deployment of accurate statistical machine translation systems, benchmarks, and statistics. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal. Association for Computational Linguistics, pp. 74–78.
- Biçici E. and Yuret D. (2011). Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland. Association for Computational Linguistics, pp. 272–283.

- Biçici E. and Yuret D.** (2015). Optimizing instance selection for statistical machine translation with feature decay algorithms. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23(2), 339–350.
- Bojar O., Chatterjee R., Federmann C., Haddow B., Huck M., Hokamp C., Koehn P., Logacheva V., Monz C., Negri M., Post M., Scarton C., Specia L. and Turchi M.** (2015). Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal. Association for Computational Linguistics, pp. 1–46.
- Callison-Burch C., Bannard C. and Schroeder J.** (2005). Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor, USA. The Association for Machine Translation in the Americas, pp. 255–262.
- Clark J.H., Dyer C., Lavie A. and Smith N.A.** (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, Portland, Oregon. Association for Computational Linguistics, pp. 176–181.
- Dean J. and Ghemawat S.** (2008). Mapreduce: Simplified data processing on large clusters. *Communications of the ACM* 51(1), 107–113.
- Doddington G.** (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, San Diego, CA, pp. 138–145.
- Eck M., Vogel S. and Waibel A.** (2005a). Low cost portability for statistical machine translation based on n-gram coverage. In *Proceedings of MT Summit X*, Phuket, Thailand. Citeseer, pp. 227–234.
- Eck M., Vogel S. and Waibel A.** (2005b). Low cost portability for statistical machine translation based on n-gram frequency and TF-IDF. In *2005 International Workshop on Spoken Language Translation, IWSLT*, Pittsburgh, PA, USA, pp. 61–67.
- Eetemadi S., Lewis W., Toutanova K. and Radha H.** (2015). Survey of data selection methods in statistical machine translation. *Machine Translation* 29(3–4), 189–223.
- Freitag M. and Al-Onaizan Y.** (2016). Fast domain adaptation for neural machine translation. arXiv preprint [arXiv:1612.06897](https://arxiv.org/abs/1612.06897).
- Gascó G., Rocha M.-A., Sanchis-Trilles G., Andrés-Ferrer J. and Casacuberta F.** (2012). Does more data always yield better translations? In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France. Association for Computational Linguistics, pp. 152–161.
- Germann U.** (2014). Dynamic phrase tables for machine translation in an interactive post-editing scenario. In *Proceedings of the Workshop on Interactive and Adaptive Machine Translation*, pp. 20–31.
- Germann U.** (2015). Sampling phrase tables for the mooses statistical machine translation system. *The Prague Bulletin of Mathematical Linguistics* 104(1), 39–50.
- Haffari G., Roy M. and Sarkar A.** (2009). Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado. Association for Computational Linguistics, pp. 415–423.
- Heafield K.** (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland. Association for Computational Linguistics, pp. 187–197.
- 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 *Proceedings of the 10th Annual Conference of the European Association for Machine Translation*, Budapest, Hungary. European Association for Machine Translation, pp. 133–142.
- Hoang C. and Simaan K.** (2014). Latent domain translation models in mix-of-domains haystack. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland. Dublin City University and Association for Computational Linguistics, pp. 1928–1939.
- Johnson H., Martin J., Foster G. and Kuhn R.** (2007). Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic. Association for Computational Linguistics, pp. 967–975.
- Khadiji S. and Ney H.** (2005). Automatic filtering of bilingual corpora for statistical machine translation. In *International Conference on Application of Natural Language to Information Systems*, Alicante, Spain, pp. 263–274.
- Kirchhoff K. and Bilmes J.** (2014). Submodularity for data selection in machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar. Association for Computational Linguistics, pp. 131–141.
- Klein G., Kim Y., Deng Y., Senellart J. and Rush A.M.** (2017). Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations*, Vancouver, Canada. Association for Computational Linguistics, pp. 67–72.
- Kneser R. and Ney H.** (1995). Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Detroit, MI. IEEE, pp. 181–184.
- Koehn P.** (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, pp. 388–395.

- 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 SMT. In *Proceedings of 45th Annual Meeting of the ACL on Interactive Poster & Demonstration Sessions*, Prague, Czech Republic. Association for Computational Linguistics, pp. 177–180.
- Lopez A.D. (2008). *Machine Translation by Pattern Matching*. PhD Thesis, University of Maryland, College Park, MD, USA.
- Luong M.-T. and Manning C.D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, Da Nang, Vietnam, pp. 76–79.
- Manber U. and Myers G. (1993). Suffix arrays: A new method for on-line string searches. *SIAM Journal on Computing* 22(5), 935–948.
- Mandal A., Vergyri D., Wang W., Zheng J., Stolcke A., Tur G., Hakkani-Tur D. and Ayan N.F. (2008). Efficient data selection for machine translation. In *Spoken Language Technology Workshop, 2008*, Goa, India. IEEE, pp. 261–264.
- Moore, R.C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, Uppsala, Sweden. Association for Computational Linguistics, pp. 220–224.
- Och F. (2003). Minimum error rate training in statistical machine translation. In *ACL-2003: 41st Annual Meeting of the Association for Computational Linguistics, Proceedings*, Sapporo, Japan. Association for Computational Linguistics, pp. 160–167.
- Och F. and Ney H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1), 19–51.
- Ozdowska S. and Way A. (2009). Optimal bilingual data for French-English PB-SMT. In *Proceedings of the 13th Annual Meeting of the European Association for Machine Translation*, Barcelona, Spain. European Association for Machine Translation, pp. 96–103.
- Papineni K., Roukos S., Ward T. and Zhu W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics, pp. 311–318.
- Parcheta Z., Sanchis-Trilles G. and Casacuberta F. (2018). Data selection for NMT using infrequent n-gram recovery. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, Alicante, Spain. European Association for Machine Translation, pp. 219–227.
- Poncelas A. (2019). *Improving Transductive Data Selection Algorithms for Machine Translation*. PhD Thesis, Dublin City University.
- Poncelas A., de Buy Wenniger G.M. and Way A. (2018). Data selection with feature decay algorithms using an approximated target side. In *15th International Workshop on Spoken Language Translation (IWSLT 2018)*, Bruges, Belgium, pp. 173–180.
- Poncelas A., de Buy Wenniger G.M. and Way A. (2019a). Adaptation of machine translation models with back-translated data using transductive data selection methods. In *20th International Conference on Computational Linguistics and Intelligent Text Processing*, La Rochelle, France.
- Poncelas A., de Buy Wenniger G.M. and Way A. (2019b). Transductive data selection algorithms for fine-tuning neural machine translation. In *Proceedings of The 8th Workshop on Patent and Scientific Literature Translation*, Dublin, Ireland. European Association for Machine Translation, pp. 13–23.
- Poncelas A., Maillette de Buy Wenniger G. and Way A. (2017). Applying n-gram alignment entropy to improve feature decay algorithms. *The Prague Bulletin of Mathematical Linguistics* 108(1), 245–256.
- Poncelas A., Maillette de Buy Wenniger G. and Way A. (2018). Feature decay algorithms for neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, Alicante, Spain. European Association for Machine Translation, pp. 239–248.
- Poncelas A. and Way A. (2019). Selecting artificially-generated sentences for fine-tuning neural machine translation. In *Proceedings of the 12th International Conference on Natural Language Generation*, Tokyo, Japan. Association for Computational Linguistics.
- Poncelas A., Way A. and Sarasola K. (2018). The ADAPT system description for the IWSLT 2018 Basque to English translation task. In *International Workshop on Spoken Language Translation*, Bruges, Belgium, pp. 72–82.
- Poncelas A., Way A. and Toral A. (2016). Extending feature decay algorithms using alignment entropy. In *International Workshop on Future and Emerging Trends in Language Technology*, Seville, Spain. Springer, pp. 170–182.
- Popovic M. (2015). chrF: Character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal. Association for Computational Linguistics, pp. 392–395.
- Salton G. and Yang C.-S. (1973). On the specification of term values in automatic indexing. *Journal of Documentation* 29(4), 351–372.
- Silva C.C., Liu C.-H., Poncelas A. and Way A. (2018). Extracting in-domain training corpora for neural machine translation using data selection methods. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, Brussels, Belgium. Association for Computational Linguistics, pp. 224–231.
- Snover M., Dorr B., Schwartz R., Micciulla L. and Makhoul J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas, pp. 223–231.

- Soto X., Shterionov D., Poncelas A. and Way A.** (2020). Selecting backtranslated data from multiple sources for improved neural machine translation. In *Proceedings of The 58th Annual Conference of the Association for Computational Linguistics, ACL*, Seattle, USA. Association for Computational Linguistics (accepted).
- Taghipour K., Afhami N., Khadivi S. and Shiry S.** (2010). A discriminative approach to filter out noisy sentence pairs from bilingual corpora. In *Proceedings of 5th International Symposium on Telecommunications (IST 2010)*, Tehran, Iran. IEEE, pp. 537–541.
- van der Wees M., Bisazza A. and Monz C.** (2017). Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics, pp. 1400–1410.
- Vapnik V.N.** (1998). *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley-Interscience.
- Wang L., Wong D.F., Chao L.S., Lu Y. and Xing J.** (2014). A systematic comparison of data selection criteria for smt domain adaptation. *The Scientific World Journal* **2014**, 1–10.
- Zens R., Stanton D. and Xu P.** (2012). A systematic comparison of phrase table pruning techniques. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea. Association for Computational Linguistics, pp. 972–983.