Zipporah: a Fast and Scalable Data Cleaning System for Noisy Web-Crawled Parallel Corpora

Hainan Xu **Philipp Koehn**
Department of Compute Science,
Center for Language and Speech Processing,
Johns Hopkins University, U.S.A., 21218
hxu31@jhu.edu phi@jhu.edu

Abstract

We introduce Zipporah, a fast and scalable data cleaning system. We propose a novel type of bag-of-words translation feature, and train logistic regression models to classify good data and synthetic noisy data in the proposed feature space. The trained model is used to score parallel sentences in the data pool for selection. As shown in experiments, Zipporah selects a high-quality parallel corpus from a large, mixed quality data pool. In particular, for one noisy dataset, Zipporah achieves a 2.1 BLEU score improvement with using 1/5 of the data over using the entire corpus.

1 Introduction

Statistical machine translation (SMT) systems require the use of parallel corpora for training the internal model parameters. Data quality is vital for the performance of the SMT system (Simard, 2014). To acquire a massive parallel corpus, many researchers have been using the Internet as a resource, but the quality of data acquired from the Internet usually has no guarantee, and data cleaning/data selection is needed before the data is used in actual systems. Usually data cleaning refers to getting rid of a small amount of very noisy data from a large data pool, and data selection refers to selecting a small subset of clean (or in-domain) data from the data pool; both have the objective of improving translation performances. For practical purposes, it is highly desirable to perform data selection in a very fast and scalable manner. In this paper we introduce Zipporah[1], a fast and scalable system which can select an arbitrary size of good data from a large noisy data pool to be used in SMT model training.

[1] https://github.com/hainan-xv/zipporah

2 Prior Work

Many researchers have studied the data cleaning/selection problem. For data selection, there have been a lot of work on selecting a subset of data based on domain-matching. Duh et al. (2013) used a neural network based language model trained on a small in-domain corpus to select from a larger data pool. Moore and Lewis (2010) computed cross-entropy between in-domain and out-of-domain language models to select data for training language models. XenC (Rousseau, 2013), an open-source tool, also selects data based on cross-entropy scores on language models. Axelrod et al. (2015) utilized part-of-speech tags and used a class-based n-gram language model for selecting in-domain data. There are a few works that utilize other metrics. Lü et al. (2007) redistributed different weights for sentence pairs/predefined sub-models. Shah and Specia (2014) described experiments on quality estimation which, given a source sentence, select the best translation among several options. The qeclean system (Denkowski et al., 2012; Dyer et al., 2010; Heafield, 2011) uses word alignments and language models to select sentence pairs that are likely to be good translations of one another.

For data cleaning, a lot of researchers worked on getting rid of noising data. Taghipour et al. (2011) proposed an outlier detection algorithm which leads to an improved translation quality when trimming a small portion of data. Cui et al. (2013) used a graph-based random walk algorithm to do bilingual data cleaning. BiTextor (Esplá-Gomis and Forcada, 2009) utilizes sentence alignment scores and source URL information to filter out bad URL pairs and selects good sentence pairs.

In this paper we propose a novel way to evaluate the quality of a sentence pair which runs efficiently. We do not make a clear distinction

between data selection and data cleaning in this work, because under different settings, our method can perform either based on the computed quality scores of sentence pairs.

3 Method

The method in this paper works as follows: we first map all sentence pairs into the proposed feature space, and then train a simple logistic regression model to separate known good data and (synthetic) bad data. Once the model is trained, it is used to score sentence pairs in the noisy data pool. Sentence pairs with better scores are added to the selected subset until the desired size constraint is met.

3.1 Features

Since good adequacy and fluency are the major two elements that constitute a good parallel sentence pair, we propose separate features to address both of them. For adequacy, we propose bag-of-words translation scores, and for fluency we use n-gram language model scores. For notational simplicity, in this section we assume the sentence pair is French-English in describing the features, and we will use subscripts f and e to indicate the languages. In designing the features, we prioritize efficiency as well as performance since we could be dealing with corpora of huge sizes.

3.1.1 Adequacy scores

We view each sentence as a bag of words, and design a "distance" between the sentence pairs based on a bag-of-words translation model. To do this, we first generate dictionaries from an aligned corpus, and represent them as sets of triplets. Formally,

$$D_{\text{f2e}} = \{(w_{f_i}, w_{e_i}, p(w_{e_i}|w_{f_i})), i = 1, ..., m\}.$$

Given a sentence pair (s_f, s_e) in the noisy data pool, we represent the two sentence as two sparse word-frequency vectors v_f and v_e. For example for any French word w_f, we have $v_f[w_f] = \frac{c(w_f, s_f)}{l(s_f)}$, where $c(w_f, s_f)$ is the number of occurrences of w_f in s_f and $l(s_f)$ is the length of s_f. We do the same for v_e. Notice that by construction, both vectors add up to 1 and represent a proper probability distribution on their respective vocabularies. Then we "translate" v_f into v'_e, based on the probabilistic f2e dictionary, where

$$v'_e[w_e] = \sum_{w_f} v_f[w_f] p(w_e|w_f)$$

For a French word w that does not appear in the dictionary, we keep it as it is in the translated vector, i.e. assume there is an entry of $(w, w, 1.0)$ in the dictionary. Since the dictionary is probabilistic, the elements in v'_e also add up to 1, and v'_e represents another probability distribution on the English vocabulary. We compute the (smoothed) cross-entropy between v_e and v'_e,

$$\text{xent}(v_e, v'_e) = \sum_{w_e} v_e[w_e] \log \frac{1}{v'_e[w_e] + c} \quad (1)$$

where c is a smoothing constant to prevent the denominator from being zero, and set $c = 0.0001$ for all experiments in this paper (more about this in Section 4).

We perform similar procedures for English-to-French, and compute $\text{xent}(v_f, v'_f)$. We define the adequacy score as the sum of the two:

$$\text{adequacy}(s_f, s_e) = \text{xent}(v_e, v'_e) + \text{xent}(v_f, v'_f)$$

3.1.2 Fluency scores

We train two n-gram language models with a clean French and English corpus, and then for each sentence pair (s_f, s_e), we score each sentence with the corresponding model, $\mathcal{F}_{\text{ngram}}(s_f)$ and $\mathcal{F}_{\text{ngram}}(s_e)$, each computed as the ratio between the sentence negative log-likelihood and the sentence length. We define the fluency score as the sum of the two:

$$\text{fluency}(s_f, s_e) = \mathcal{F}_{\text{ngram}}(s_f) + \mathcal{F}_{\text{ngram}}(s_e)$$

3.2 Synthetic noisy data generation

We generate synthetic noisy data from good data, and make sure the generated noisy data include sentence pairs with a) good fluency and bad adequacy, b) good adequacy and bad fluency and c) bad both.

Respectively, we generate 3 types of "noisy" sentence pairs from a good corpus: a) shuffle the sentences in the target language file (each sentence in the source language would be aligned to a random sentence in the target language); b) shuffle the words within each sentence (each sentence will be bad but the pairs are good translations in the "bag-of-words" sense); c) shuffle both the sentences and

words. We emphasize that, while the synthetic data might not represent "real" noisy data, it has the following advantages: 1) each type of noisy data is equally represented so the classifier has to do well on all of them; 2) the data generated this way would be among the hardest to classify, especially type a and type b, so if a classifier separates such hard data with good performance, we expect it to also be able to do well in real world situations.

3.3 Logistic regression feature mapping

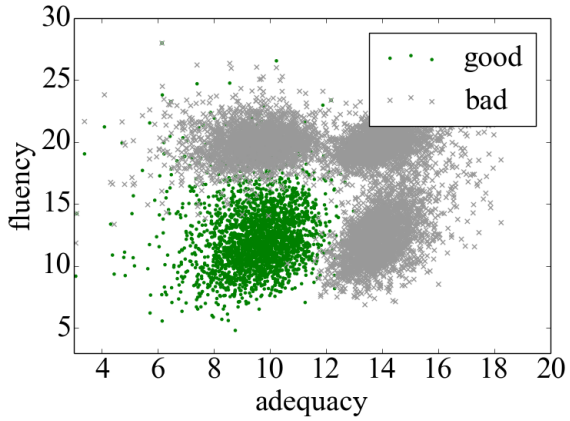

Figure 1: newstest09 fr-en data in the feature space

We plot the newstest09 data (original and auto-generated noisy ones as described in Section 3.2) into the proposed feature space in Figure 1. We observe that the clusters are quite separable, though the decision function would not be linear. We map the features into higher order forms of (x^n, y^n) in order for logistic regression to train a non-linear decision boundary.[2] We use $n = 8$ in this work since it gives the best classification performance on the newstest09 fr-en corpus.

4 Hyper-parameter Tuning

We conduct experiments to determine the value of the constant c in the smoothed cross-entropy computation in equation 1. We choose the newstest09 German-English corpus, and shuffle the sentences in the English file and combine the original (clean) corpus with the shuffled (noisy) corpus into a larger corpus, where half of them are good sentence pairs. We set different values of c and use the adequacy scores to pick the better half, and compute the retrieval accuracy. Table 1 shows that the best value for c is 0.0001, and we use that in all experiments.

c	accuracy
0.001	0.975
0.0001	**0.984**
0.00001	0.983
0.000001	0.981

Table 1: Tuning cross-entropy constant c

5 Evaluation

We evaluate Zipporah on 3 language pairs, French-English, German-English and Spanish-English. The noisy web-crawled data comes from an early version of `http://statmt.org/paracrawl`. The number of words are (in millions) 340, 487 and 70 respectively.

To generate the dictionaries for computing the adequacy scores, we use fast_align (Dyer et al., 2013) to align the Europarl (Koehn, 2005) corpus and generate probabilistic dictionaries from the alignments. We set the n-gram order to be 5 and use SRILM (Stolcke et al., 2011) to train language models on the Europarl corpus and generate the n-gram scores.

For each language pair, we use scikit-learn (Pedregosa et al., 2011) to train a logistic regression model to classify between the original and the synthetic noisy corpus of newstest09, and the trained model is used to score all sentence pairs in the data pool. We keep selecting the best ones until the desired number of words is reached.

To evaluate the quality, we train a Moses (Koehn et al., 2007) SMT system on selected data, and evaluate each trained SMT system on 3 test corpora: newstest2011 which contains 3003 sentence pairs, and a random subset of the TED-talks corpus and the movie-subtitle corpus from OPUS (Tiedemann, 2012), each of which contains 3000 sentence pairs.

Tables 2, 3 and 4 show the BLEU performance of the selected subsets of the Zipporah system compared to the baseline, which selects sentence pairs at random; for comparison, we also give the BLEU performance of systems trained on Europarl. The Zipporah system gives consistently better performance across multiple datasets and multiple languages than the baseline.[3]

[2] We avoid using multiple mappings of one feature because we want the scoring function to be monotonic both w.r.t x and y, which could break if we allow multiple higher-order mappings of the same feature and they end up with weights with different signs.

[3] We also point out that the performance of the selected

BLEU	newstest11		ted-talk		subtitle	
num-words	rand	zipp	rand	zipp	rand	zipp
10 million	21.5	24.4	24.0	27.4	12.3	14.9
20 million	22.8	25.1	25.0	27.9	12.8	15.5
50 million	24.3	26.0	27.4	28.8	14.5	15.8
100 million	25.2	26.6	28.3	**30.3**	15.0	**17.3**
200 million	26.1	**26.7**	29.9	30.0	16.4	**17.3**
340 mil (all)	26.2		30.0		16.7	
Europarl	24.4		27.0		14.2	

Table 2: BLEU Performance, French-English

BLEU	newstest11		ted-talk		subtitle	
num-words	rand	zipp	rand	zipp	rand	zipp
10 million	13.6	17.6	17.0	22.5	11.4	15.8
20 million	14.8	18.4	18.9	23.7	12.7	16.9
50 million	16.3	19.2	20.8	24.8	13.9	17.8
100 million	16.9	**19.5**	21.3	**25.0**	14.0	**18.3**
200 million	18.0	19.2	22.9	24.2	15.3	17.9
487 mil (all)	18.7		23.5		16.2	
Europarl	17.5		21.5		14.5	

Table 3: BLEU Performance, German-English

BLEU	newstest11		ted-talk		subtitle	
num-words	rand	zipp	rand	zipp	rand	zipp
10 million	24.2	25.5	25.9	28.3	17.9	19.8
20 million	25.3	26.2	28.2	29.7	19.3	21.2
50 million	26.6	26.5	29.9	**30.4**	21.3	21.4
70 mil (all)	**27.1**		30.3		**21.8**	
Europarl	25.4		28.4		19.8	

Table 4: BLEU Performance, Spanish-English

In particular, for the Germen-English corpus, when selecting less than 2% of the data (10 million words), on the TED-talk dataset, Zipporah achieves a 5.5 BLEU score improvement over the baseline; by selecting less than 4% of the data (20 million words) the system gives better performance than using all data. Peak performance is achieved when selecting 100 million words, where an improvement of 2.1 BLEU score over all data is achieved on the movie-subtitle dataset, despite only using less than 1/5 of the data.

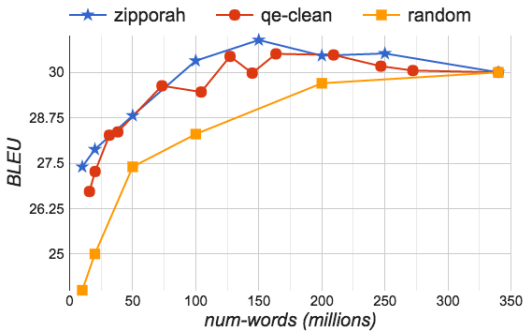

Figure 2: BLEU performance of Zipporah, qe-clean and random on TED-talks, French-English

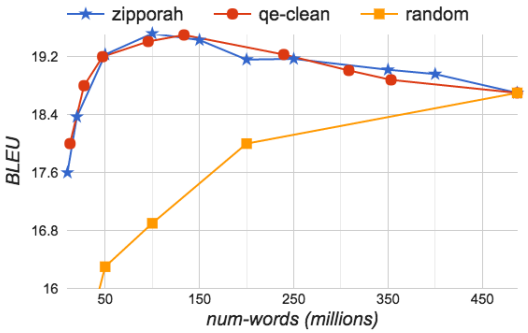

Figure 3: BLEU performance of Zipporah, qe-clean and random on newstest11, German-English

Figures 2, 3 and 4 compare the result of Zipporah with that of qe-clean (Denkowski et al., 2012; Dyer et al., 2010; Heafield, 2011) and the random baseline. We use the same data when running qe-clean, with Europarl for training and newstest09 for dev. While they both perform comparably and better than the baseline, Zipporah achieves a better peak in all the datasets, and the peak is usually achieved when selecting a smaller number of words compared to qe-clean, Another advantage of Zipporah is it allows the user to select an arbi-

subsets of the Zipporah system can surpass that of Europarl, although the Europarl corpus acts like an "oracle" in the system, upon which the dictionaries and language models for feature computations are trained.

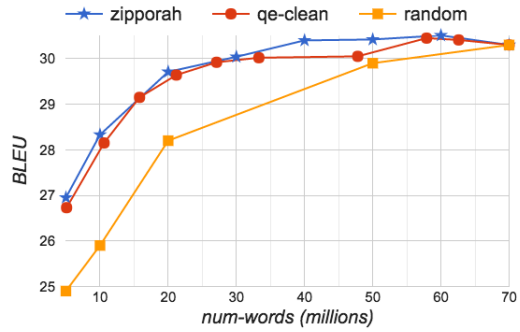

Figure 4: BLEU performance of Zipporah, qe-clean and random on TED-talks, Spanish-English

trary size from the pool.[4] We also want to emphasize that unlike qe-clean, which requires running word-alignments for all sentence pairs in the noisy corpus, Zipporah's feature computation is simple, fast and can easily be scaled for huge datasets.

6 Conclusion and Future Work

In this paper we introduced Zipporah, a fast data selection system for noisy parallel corpora. SMT results demonstrate that Zipporah can select a high-quality subset of the data and significantly improve SMT performance.

Zipporah currently selects sentences based on the "individual quality" only, and we plan in future work to also consider other factors, e.g. encourage selection of a subset that has a better n-gram coverage.

Acknowledgments

This project was funded by Google Faculty Research Award. The authors would like to thank Shuoyang Ding, Tongfei Chen, Matthew Wiesner, Winston Wu, Huda Khayrallah and Adi Renduchintala for their help during this project. The authors would also like to thank Penny Peng for her moral support.

References

Amittai Axelrod, Yogarshi Vyas, Marianna Martindale, Marine Carpuat, and Johns Hopkins. 2015. Class-based n-gram language difference models for data selection. In *IWSLT (International Workshop on Spoken Language Translation)*.

Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li, and Ming Zhou. 2013. Bilingual data cleaning for smt using graph-based random walk. In *ACL (2)*, pages 340–345.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The cmu-avenue french-english translation system. In *Proceedings of the NAACL 2012 Workshop on Statistical Machine Translation*.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *ACL (2)*, pages 678–683.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. Association for Computational Linguistics.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Miquel Esplá-Gomis and M Forcada. 2009. Bitextor, a free/open-source software to harvest translation memories from multilingual websites. *Proceedings of MT Summit XII, Ottawa, Canada. Association for Machine Translation in the Americas*.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Yajuan Lü, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *EMNLP-CoNLL*, volume 34, pages 3–350.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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

Kashif Shah and Lucia Specia. 2014. Quality estimation for translation selection. In *Proceedings of the 17th Annual Conference of the European Association for Machine Translation, Dubrovnik, Croatia*.

Michel Simard. 2014. Clean data for training statistical mt: The case of mt contamination. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas (AMTA)*.

[4] In the plots the data points of Zipporah and qe-clean are not aligned because we always select multiples of million words, but it is hard to do so with qe-clean.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. Srilm at sixteen: Update and outlook. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, volume 5.

Kaveh Taghipour, Shahram Khadivi, and Jia Xu. 2011. Parallel corpus refinement as an outlier detection algorithm. *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 414–421.

Jrg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).