Paragraph Vector for Data Selection in Statistical Machine Translation

Mirela-Stefania Duma and Wolfgang Menzel
University of Hamburg
Natural Language Systems Division
{mduma, menzel}@informatik.uni-hamburg.de

Abstract

In this paper, we investigate data selection methods used in domain adaptation for Statistical Machine Translation targeting an in-domain made up of non-standard data, such as transcriptions of spoken data. In data selection, the sentences from the general domain are scored according to their similarity to the in-domain. This research explores Paragraph Vectors as means of scoring sentences from the general domain. The experimental evaluation results show that our method improves the translation quality over the baselines, as well as over a state-of-the-art data selection method.

1 Introduction

Data selection is a widely used method for performing domain adaptation for Machine Translation (MT). Given a large pool of general domain data and a smaller-sized in-domain data, the task is to filter the sentences from the general domain with respect to their similarity to the in-domain. After scoring the general domain sentences using a similarity metric, a ratio of the general domain is kept and used for SMT. The underlying assumption is that the general domain is big enough to contain sentences similar to the in-domain. The challenges in data selection consist of choosing a metric or a method that evaluates how similar is a sentence from the general domain to the in-domain and after scoring all sentences, determining what is the ratio of general domain sentences to be kept.

As general domain data we chose the Commoncrawl corpus[1] as it is a relatively large corpus and contains crawled data from a variety of domains as well as texts having different discourse types (including spoken discourse). The in-domain consisted of the TED Talks corpora used in the IWSLT 2016 MT Evaluation Campaign[2]. The difficulties in translating TED stems from the small size of the corpus and from the unconventionality of the corpus which is a concatenation of transcribed talks having different topics. The domain adaptation problem is not only a problem of adapting to a domain, but also to spoken discourse style.

In Le and Mikolov (2014) sentences are represented as continuous vectors with empirical results that show that Paragraph Vectors outperform the traditional bag-of-words approach of representing text. It was succesffuly applied in opinion mining and information retrieval tasks (Le and Mikolov, 2014).

In this paper, we aim to determine whether using Paragraph Vectors in the scoring phase is helpful in capturing the degree of similarity of general domain sentences to TED talks. The idea was first introduced in Duma and Menzel (2016) for the task of domain adaptation to the IT domain as part of the First Conference on Machine Translation (WMT 2016). The encouraging results using Paragraph Vectors constitute the basis of our work. We aim to introduce a new scoring formula that considers sentence length and to verify whether using Paragraph Vector is also useful in the setting of translating TED talks.

We trained SMT systems on the English-German language pair and used the BLEU (Papineni et al., 2002), NIST (Doddington, 2002) and METEOR (Denkowski and Lavie, 2014) metrics in assessing the performance of the systems.

We first shortly summarize related work in data selection for SMT in Section 2, then describe Paragraph Vector in Section 3. The next section presents the experimental settings for training the

[1] http://commoncrawl.org/

[2] https://sites.google.com/site/iwsltevaluation2016/mt-track

SMT systems along with the algorithm we used in performing data selection. Lastly, Section 5 contains an overview of the systems evaluation.

2 Related work

Three approaches are commonly used in data selection: information retrieval inspired (Hildebrand et al., 2005; Lü et al., 2007; Tamchyna et al., 2012), perplexity-based (Mandal et al., 2008; Axelrod et. al, 2011; Mansour et al., 2011) and edit distance similarity inspired (Wang et al., 2013).

The state-of-the-art data selection method we chose to use for comparison with our method is perplexity-based and presented in Axelrod et. al (2011). Four language models are trained for the in-domain and the general domain source and target sides of the corpora. Given a sentence pair from the general domain, the method scores it by summing up the cross-entropy difference scores from each side of the corpus. Axelrod et. al (2015) applied this method for general domains including the Commoncrawl corpus and for the in-domain TED Talks. We name this metric PPL in the rest of the paper.

In this paper, we propose a new scoring formula for determining the similarity of a general domain sentence to the in-domain using Paragraph Vectors (Le and Mikolov, 2014) for representing the sentences as continuous vectors. The direction of using Paragraph Vectors in data selection for SMT was introduced in Duma and Menzel (2016) where the semantic similarity of sentences was successfully employed in the task of domain adaptation of MT to the IT domain as part of WMT 2016. We extend that work by introducing a new scoring formula that combines the similarity scores produced by Paragraph Vectors and we further improve the final score of a general domain sentence by using a sentence length penalty.

3 Paragraph vector

The traditional representation of text consists in the bag-of-words model which has the disadvantage of not considering the semantics of words. In order to overcome this weakness, Le and Mikolov (2014) introduce Paragraph Vectors. Similar to word vectors (Mikolov et al., 2013), Paragraph Vectors give a continuous distributed vector representation of the input. Word vectors capture the semantics of words by looking at their representations in the vector space: similar words have vectors that are closer to each other compared to non-similar words. For example, the words ”strong” and ”powerful” have their word vectors close to each other indicating their semantic similarity. Moreover, algebraic operations can be applied on the word vectors where, for example, vector(”King”) - vector(”Man”) + vector(”Woman”) gives as result a vector that is close to the vector representation of the word ”Queen” (Mikolov et al., 2013).

Going one step further than the word vectors, aiming at representing a text of variable length (phrases, sentences, documents), Paragraph Vector uses the word vectors in computing the final vector representation of the text. Since we use bilingual corpora where the basic unit is a sentence, we chose to represent sentences as vectors. The paragraph vector is concatenated with several word vectors from the sentence and used in predicting the following word given the context. The contexts have a fixed length and are sampled from a sliding window over the paragraph. The paragraph vector acts like a memory that remembers the topic of the sentence or what is missing from the current context. The word vectors and the paragraph vectors are trained using the stochastic gradient descent and back-propagation (Rumelhart et al., 1986). While paragraph vectors are unique among sentences, the word vectors are shared (Le and Mikolov, 2014).

We use single sentences as paragraphs. The reason why we adopted Paragraph Vector is because similarly to word vectors, they reflect semantic relatedness. Moreover, we have chosen Paragraph Vectors for representing sentences as vectors because the approach does not require parsing or available labeled data. The implementation of Paragraph Vectors we used is Doc2vec from the *gensim* toolkit[3].

4 Experimental Framework

For tuning the MT systems we made use of the IWSLT16.TED.dev2010 dataset, also provided by IWSLT. For evaluating the systems the IWSLT16.TED.tst2014 dataset was used.

All systems have been developed with the widely used Moses phrase-based MT toolkit (Koehn et al., 2007) and the Experiment Management System (Koehn, 2010) that facilitates the

[3] `https://radimrehurek.com/gensim/models/doc2vec.html`

preparation of scripts for experiments.

4.1 Data preprocessing

The data was tokenized, cleaned and lowercased using the scripts from EMS. Furthermore, the general domain data was filtered by removing the sentence pairs that do not pertain to the English-German language pair according to the jlangdetect library[4].

Sentences that contain non-alpha characters were removed from both corpora and punctuation was normalized. Table 1 presents some data statistics for both domains after preprocessing:

Corpora	Sentences	Tokens	
		English	German
Commoncrawl	2.34M	50.33M	46.11M
TED	196K	3.49M	3.07M

Table 1: Corpora statistics after preprocessing

4.2 Experimental settings

Word alignment was performed using GIZA++ (Och and Ney, 2003) with the default *grow-diag-final-and* alignment symmetrization method. The target side of the Commoncrawl and TED corpora was utilised in estimating 5-gram language models (LM) using the SRILM toolkit (Stolcke, 2002) with Kneser-Ney discounting (Kneser and Ney, 1995). For most of the experiments we used LM interpolation where the in-domain LM and the general domain LM were interpolated using weights tuned to minimize the perplexity on the tuning set. The same data was used for tuning the systems with MERT (Och, 2003).

4.3 Baseline systems

Two baselines were trained using the concatenation of the in-domain data and the general domain data: BS_{simple} used an LM estimated from the concatenation of the data, while the stronger baseline BS_{strong} used LM interpolation.

4.4 Data selection using Paragraph Vector

In this section the algorithm for data selection is described (Figure 1). We name our doc2vec method $SEFp$ (Sentence Embedding Filtering with penalty). The filtering procedure is similar to the one presented in (Duma and Menzel, 2016). It receives as input the bilingual in-domain corpus $\mathcal{I}n$, the bilingual general domain $\mathcal{G}en$, $\mathcal{N}$ as the number of most similar sentences that should be retrieved given a threshold δ that is used in filtering the corpus and $\mathcal{P}$, the percentage of sentences to be selected from the general domain. To train the paragraph vectors we concatenated $\mathcal{I}n$ and $\mathcal{G}en$ resulting in data set $\mathcal{C}$. The steps needed for training the doc2vec model required tagging every sentence from the source side of the concatenated corpus $\mathcal{C}_{source}$ with its corresponding line number in the corpus and building a vocabulary from the tagged $\mathcal{C}$. Therefore, a sentence that came from $\mathcal{I}n$ was tagged with a number from $[1, size_{In}]$ and a sentence that came from $\mathcal{G}en$ was tagged with a number from $[size_{In} + 1, size_{In} + size_{Gen}]$. The doc2vec model $\mathcal{M}$ was trained on the tagged C_{source}.

Given a sentence pair $(s_i, t_i) \in \mathcal{G}en$, the top $\mathcal{N}$ most similar vectors to s_i are computed and retrieved in the form of a pair $(index, score)$ where $index$ gives the tag (i.e. the line number) of the selected similar sentence to s_i and $score$ specifies the similarity between s_i and s_{index}. The similarity is computed as the cosine between the two vectors.

The next step is computing the sentence score. The retrieved top $\mathcal{N}$ scores include similarities with sentences either from $\mathcal{I}n$ or $\mathcal{G}en$. Given a sentence $s_i \in \mathcal{G}en$, only the similarity scores between s_i and sentences from $\mathcal{I}n$ $(index_j < size_{In})$ contribute to building the final sentence score. These scores are filtered using the threshold δ set to 0.5. We plan to further experiment with other values of δ in future work. Since the degree of similarity matters, we favor similarity scores with $\mathcal{I}n$ sentences that are higher over similarity scores that are lower. This preference has been implemented by means of the position index j in the ranked list of selected sentences R_i.

The final score of the general domain sentence is built by accumulating all the intermediary scores. We observed that some sentences from Commoncrawl are very long leading to a very high score. We introduced a sentence length penalty with the purpose of giving a penalty to long sentences by dividing the final score to $size_{s_i}$, the number of words for s_i.

In comparison to the work in (Duma and Menzel, 2016), here we introduce a new scoring formula with the aim of investigating new possibilities of combining similarity scores produced by

[4] https://github.com/melix/jlangdetect

Algorithm 1 Doc2vec Filtering with penalty

```
procedure FILTER($\mathcal{I}n, \mathcal{G}en, \mathcal{N}, \delta, \mathcal{P}$)
    $\mathcal{C} \leftarrow \mathcal{I}n + \mathcal{G}en$
    for each sentence $s_i \in \mathcal{C}_{source}$ do
        tag $s_i$ with the line number $i$
    build vocabulary from tagged $\mathcal{C}_{source}$
    train doc2vec model $\mathcal{M}$ using tagged $\mathcal{C}_{source}$
    for each sentence pair $(s_i, t_i) \in \mathcal{G}en$ do
        $\mathcal{R}_i = top(\mathcal{N}, mostSimilar(\mathcal{M}, s_i))$
        $Sim_{s_i} = \{(index, score) \in \mathcal{R}_i |\ index \in [1, size_C], score \in (0,1)\}$
        for $(index_j, score_j) \in Sim_{s_i}$ do
            $score_{i,j} = \begin{cases} score_{i,j}^2 * \frac{\mathcal{N}}{j}, & \text{if } index_j < size_{In} \text{ and } score_j > \delta \\ 0, & \text{otherwise} \end{cases}$
        $score_i = \sum_{j=1}^{\mathcal{N}} \frac{score_{i,j}}{size_{s_i}}$
    sort sentences $\in \mathcal{G}en$ by their score in descending order
    while $i \leq \mathcal{P}$ do
        add $(s_i, t_i)$ to $FilteredCorpus_{\mathcal{P}}$
```

Figure 1: Doc2vec filtering algorithm

Doc2Vec. Moreover, we consider the sentence length penalty in computing the final score of a general domain sentence.

After scoring all the general domain sentences, we sorted them in descending order and filtered the general domain sentences using the percentage $\mathcal{P}$ as the ratio of Commoncrawl sentences to be kept. We increased the ratio with 10% at every SMT model training.

The final step consisted in training several SMT systems on a concatenation of the reduced general domain corpus $FilteredCorpus_{\mathcal{P}}$ and the in-domain data $\mathcal{I}n$ and using LM interpolation of an LM estimated using $\mathcal{I}n$ and an LM estimated using the full $\mathcal{G}en$. The same interpolated LM was used in the PPL experiments and in the BS_{strong} baseline.

5 Evaluation and Conclusions

We evaluated the two baselines, the PPL metric and our proposed $SEFp$ metric using the BLEU, NIST and METEOR metrics, widely used in evaluating MT output.

Both data selection methods outperform the baselines as their maximum BLEU, NIST and METEOR scores are greater than the baseline scores. According to the BLEU scores, the best result is obtained by the $SEFp$ metric, when selecting 40% of the general domain data (BLEU = 20.23). It is to be noted that the best BLEU result obtained by the state-of-the-art metric is achieved when selecting 60% of the data (BLEU = 20.11), thus it requires more data compared to $SEFp$. The NIST scores indicate also that the best result is obtained using our method, when selecting 40% of the general data (NIST = 20.34). Evaluating the results using METEOR, both methods give the same maximum score of 41.84. However, our method uses 50% of the general domain data, while the PPL metric requires 80% of the general domain data to achieve the maximum score.

For future work we plan to further exploit Paragraph Vector by employing other scoring methods, evaluating the method proposed in (Duma and Menzel, 2016) on TED talks and also combining the currently presented approach with the Axelrod et al. (2011) approach. Moreover, an interesting idea for combining the bitexts (the in-domain data and the general domain selected sentences) is presented in Wang et al. (2016) where balanced concatenation with repetitions is used in order to have comparable sizes of bitexts.

To conclude, in this paper we introduced a new scoring method for data selection in SMT using

	BLEU		**NIST**		**METEOR**	
Percentage $\mathcal{P}$ of Commoncrawl	PPL	$SEFp$	PPL	$SEFp$	PPL	$SEFp$
10%	19.8	19.87	19.88	19.98	41.7	41.67
20%	19.91	19.51	20.02	19.62	41.7	41.59
30%	19.95	19.6	20.08	19.75	41.52	41.51
40%	19.96	**20.23**	20.09	**20.34**	41.63	41.77
50%	20	19.78	20.12	19.91	41.63	**41.84**
60%	**20.11**	19.89	**20.21**	20	41.73	41.61
70%	19.63	20.01	19.75	20.11	41.26	41.8
80%	20.03	19.98	20.16	20.15	**41.84**	41.78
90%	19.52	19.64	19.65	19.79	41.43	41.36
BS_strong	19.66		19.82		41.28	
BS_simple	19.79		19.89		41.11	

Table 2: Evaluation results with the BLEU, NIST and METEOR metrics

Paragraph Vector for determining the similarity of the sentences from the general domain to the in-domain. Our method outperformed the baselines and a state-of-the-art method with respect to commonly used MT evaluation metrics by achieving the highest scores using the least amount of filtered general domain data.

References

Amittai Axelrod, Xiaodong He and Jianfeng Gao. 2011. Domain Adaptation via Pseudo In-Domain Data Selection. *Proceedings of EMNLP 2011.*

Amittai Axelrod, Ahmed Elgohary, Marianna Martindale, Khánh Nguyen, Xing Niu, Yogarshi Vyas, Marine Carpuat. 2015. The UMD Machine Translation Systems at IWSLT 2015. *Proceedings of IWSLT 2015.*

Boxing Chen, Roland Kuhn and George Foster. 2013. Vector Space Model for Adaptation in Statistical Machine Translation *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1285-1293, Sofia, Bulgaria, August 4-9 2013.

Michael Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation.*

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Cooccurrence Statistics *Proceedings of the Second International Conference on Human Language Technology Research.*

Mirela-Stefania Duma and Wolfgang Menzel. 2016. Data selection for IT Texts using Paragraph Vector. *Proceedings of the First Conference on Machine Translation*, Volume 2: Shared Task Papers, pages 428–434.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval. *Proceedings of EAMT 2005.*

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for N-gram language modeling. *Proceedings ICASSP*, pages 181-184.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions.* June 25-27, 2007, Prague, Czech Republic.

Philipp Koehn. 2010. An experimental management system. *The Prague Bulletin of Mathematical Linguistics*, 94.

Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *Proceedings of the 31st International Conference on Machine Learning*, volume 32, Beijing, China. JMLR: W&CP.

Yajuan Lü, Jin Huang and Qun Liu. 2007. Improving Statistical Machine Translation Performance by Training Data Selection and Optimization. *Prooceedings of EMNLP-CoNLL 2007.*

A. Mandal, D. Vergyri, W. Wang, J. Zheng, A. Stolcke, G. Tur, D. Hakkani-Tür, and N. F. Ayan. 2008.

Efficient data selection for machine translation. *Proceedings IEEE Workshop on Spoken Language Technology*.

Saab Mansour, Joern Wuebker and Hermann Ney. 2011. Combining Translation and Language Model Scoring for Domain-Specific Data Filtering. *Proceedings of IWSLT*.

Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160-167, July 07-12, 2003, Sapporo, Japan.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, volume 29, number 1, pages 19-51.

Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, July 07-12, 2002, Philadelphia, Pennsylvania.

Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Andreas Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. *International Conference on Spoken Language Processing*.

Aleš Tamchyna, Galuščáková Petra, Kamran Amir, Stanojević Miloš and Bojar Ondřej. 2012. Selecting Data for English-to-Czech Machine Translation. *Proceedings of the Seventh Workshop on Statistical Machine Translation*.

Longyue Wang, Derek F. Wong, Lidia S. Chao, Junwen Xing and Yi Lu. 2013. Edit Distance: A New Data Selection Criterion for Domain Adaptation in SMT. *Proceedings of Recent Advances in Natural Language Processing*.

Pidong Wang, Preslav Nakov and Hwee Tou Ng. 2016. Source Language Adaptation Approaches for Resource-Poor Machine Translation. *Computational Linguistics* Vol. 42, No. 2: 277–306.