Parallel Sentence Extraction from Comparable Corpora with Neural Network Features

Chenhui Chu[1], **Raj Dabre**[2], **Sadao Kurohashi**[2]
[1]Japan Science and Technology Agency
[2]Graduate School of Informatics, Kyoto University
E-mail: chu@pa.jst.jp, raj@nlp.ist.i.kyoto-u.ac.jp, kuro@i.kyoto-u.ac.jp

Abstract

Parallel corpora are crucial for machine translation (MT), however they are quite scarce for most language pairs and domains. As comparable corpora are far more available, many studies have been conducted to extract parallel sentences from them for MT. In this paper, we exploit the neural network features acquired from neural MT for parallel sentence extraction. We observe significant improvements for both accuracy in sentence extraction and MT performance.

Keywords: Parallel Sentence Extraction, Comparable Corpora, Neural Network

1. Introduction

Neural machine translation (NMT) has achieved impressive results recently (Bahdanau et al., 2014). As the neural translation models can also be viewed as bilingual language models, they can be used to generate scores for candidate translations as neural network (NN) features for reranking the n-best lists produced by a statistical machine translation (SMT) system, whose quality rivals the state of the art (Sutskever et al., 2014).

Comparable corpora are a set of monolingual corpora that describe roughly the same topic in different languages. Although they are not exact translation equivalents of each other, there are a large amount of parallel sentences contained in the comparable texts. The task of parallel sentence extraction (Munteanu and Marcu, 2005) is to identify truly parallel sentences from the erroneous ones from comparable corpora. Intuitively, because the NN features give a measure of the bilingual similarity of a sentence pair, they could be helpful for this task. However, this assumption has not been verified previously.

In this paper, we incorporate the NN features into a robust parallel sentence extraction system (Chu et al., 2014), which consists of a parallel sentence candidate filter and a binary classifier for parallel sentence identification. The NN features are naturally used as additional features for the classifier. Experiments conducted on Wikipedia data show that the NN features improve the strong baseline system significantly for both accuracy in sentence extraction and SMT performance.

2. Parallel Sentence Extraction System

The overview of our parallel sentence extraction system is presented in Figure 1. We first align articles on the same topic in Wikipedia via the interlanguage links ((1) in Figure 1). Then we generate all possible sentence pairs by the Cartesian product from the aligned articles, and discard the pairs that do not fulfill the conditions of a filter to reduce the candidates keeping more reliable sentences ((2) in Figure 1). Next, we use a classifier trained on a small number of parallel sentences from a seed parallel corpus to identify the parallel sentences from the candidates ((3) in Figure 1). Finally, we train a NMT model on the extracted parallel sentences, and use it to obtain the NN features for the classifier to further improve the performance ((4) in Figure 1). The strategy of the filter and the features used for the classifier will be described in Section 2.1. and Section 2.2. in detail.

2.1. Parallel Sentence Candidate Filtering

A parallel sentence candidate filter is necessary, because it can remove most of the noise introduced by the simple Cartesian product sentence generator, and reduce computational cost for parallel sentence identification. Following (Chu et al., 2014), we use a filter with sentence length ratio and dictionary-based word overlap conditions.

2.2. Parallel Sentence Identification by Binary Classification

As the quality of the extracted sentences is determined by the accuracy of the classifier, the classifier becomes the core component of the extraction system. In this section, we first describe the training and testing process, and then introduce the features we use for the classifier.

2.2.1. Training and Testing

We use a support vector machine (SVM) classifier. Training and testing instances for the classifier are created following the method of (Munteanu and Marcu, 2005). We use a small number of parallel sentences from a seed parallel corpus as positive instances. Negative instances are generated by the Cartesian product of the positive instances excluding the original positive instances, and they are filtered by the same filtering method used in Section 2.1.. Moreover, we randomly discard some negative instances for training when necessary,[1] to guarantee that the ratio of negative to positive instances is less than five for the performance of the classifier.

2.2.2. Features

Baseline Features.

- Percentage of words on each side that have a translation on the other side (according to the bilingual dictionary), i.e., *word overlap*.

[1]Note that we keep all negative instances for testing.

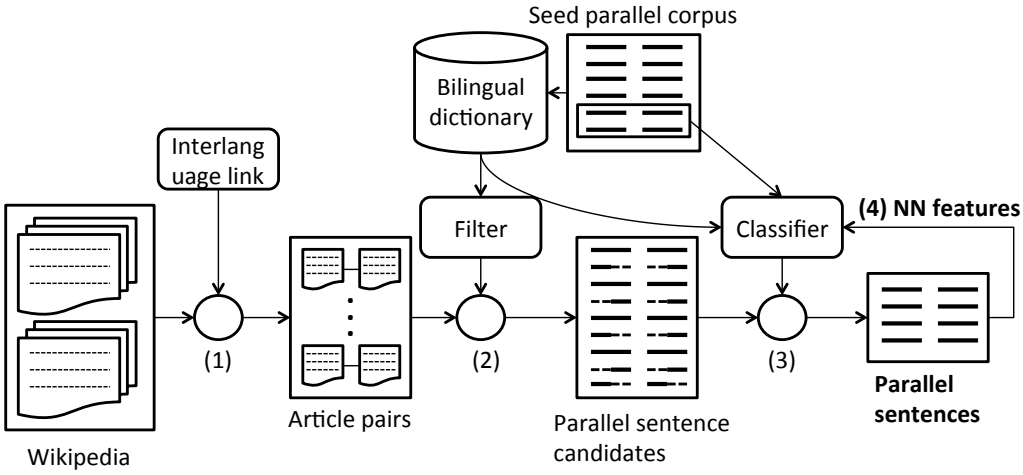

Figure 1: Parallel sentence extraction system.

- Percentage of words that are content words on each side.
- Percentage of content words on each side that have a translation on the other side (according to the bilingual dictionary).
- Sentence length, length difference, and length ratio.[2]
- Alignment features:
  - Percentage and number of words that have no connection on each side.
  - Top three largest fertilities.[3]
  - Length of the longest contiguous connected span.
  - Length of the longest unconnected substring.
- Same word features. Parallel sentences often contain same words, such as abbreviations and numbers. Same words can be helpful clues to identify parallel sentences. We use the following features:
  - Percentage and number of words that are the same on each side.

We determine a word as a content or function word using predefined part-of-speech (POS) tag sets of function words. The alignment features are extracted from the alignment results of the parallel and non-parallel sentences used as instances for the classifier. Note that alignment features may be unreliable when the quantity of non-parallel sentences is greatly larger than that of the parallel sentences.

NN Features. Using a parallel corpus, we train 4 neural translation models. For each translation direction, we train character and word based models using the corpus (2 directions and 2 types of model lead to 4 models). We use the freely available toolkit GroundHog[4] for NMT, which contains an implementation of the work by (Bahdanau et al., 2014). After training a neural translation model, it can be used to produce a score for a sentence pair, where the neural translation model can be viewed as a bilingual language model. These 4 scores are used as the NN features for the classifier.

We trained two types of models, one on the seed parallel corpus, and the other on the parallel sentences extracted by the baseline system. The models trained on the seed parallel corpus are used for producing the NN features for training and testing the classifier. We tried the use of both the two types of models to score the parallel sentence candidates for extraction.

3. Experiments

We evaluated classification accuracy, and conducted extraction and translation experiments on Chinese-Japanese data to verify the effectiveness of our proposed NN features. In all our experiments, we preprocessed the data by segmenting Chinese and Japanese sentences using a segmenter proposed by Chu et al. (2012) and JUMAN (Kurohashi et al., 1994) respectively.

3.1. Data

The seed parallel corpus we used is the Chinese-Japanese section of the Asian Scientific Paper Excerpt Corpus (ASPEC),[5] containing 680k sentences pairs ($18.2M$ Chinese and $21.8M$ Japanese tokens, respectively).

Also, we downloaded Chinese[6] (20120921) and Japanese[7] (20120916) Wikipedia database dumps. We used an open-source Python script[8] to extract and clean the text from the dumps. Since the Chinese dump is mixed of Traditional and Simplified Chinese, we converted all Traditional Chinese to Simplified Chinese using a conversion table published by Wikipedia[9]. We aligned the articles on the same topic in

[2] In our experiments, sentence length was calculated based on the number of words in a sentence.

[3] Fertility defines the the number of words that a word is connected to in an alignment (1993).

[4] https://github.com/lisa-groundhog/GroundHog

[5] http://lotus.kuee.kyoto-u.ac.jp/ASPEC

[6] http://dumps.wikimedia.org/zhwiki

[7] http://dumps.wikimedia.org/jawiki

[8] http://code.google.com/p/recommend-2011/source/browse/Ass4/WikiExtractor.py

[9] http://svn.wikimedia.org/svnroot/mediawiki/branches/REL1_12/phase3/includes/ZhConversion.php

Chinese and Japanese via the interlanguage links, obtaining 162k article pairs (2.1M Chinese and 3.5M Japanese sentences respectively).

3.2. Classification Accuracy Evaluation

We evaluated classification accuracy using two distinct sets of 5k parallel sentences from the seed parallel corpus for training and testing respectively.

3.2.1. Settings

We followed the settings used in (Chu et al., 2014).

- Word alignment tool: GIZA++.[10]
- Dictionary: Top 5 translations with translation probability larger than 0.1 created from the seed parallel corpus.
- Classifier: LIBSVM[11] with 5-fold cross-validation and radial basis function (RBF) kernel.
- Sentence length ratio threshold: 2.
- Word overlap threshold: 0.25.
- Classifier probability threshold: 0.9.

3.2.2. Evaluation

We evaluated the performance of classification by computing precision, recall and F-measure, defined as:

$$precision = 100 \times \frac{classified\_well}{classified\_parallel}, \quad (1)$$

$$recall = 100 \times \frac{classified\_well}{true\_parallel}, \quad (2)$$

$$F - measure = 2 \times \frac{precision \times recall}{precision + recall}. \quad (3)$$

Where $classified\_well$ is the number of pairs that the classifier correctly identified as parallel, $classified\_parallel$ is the number of pairs that the classifier identified as parallel, $true\_parallel$ is the number of truly parallel pairs in the test set.

3.2.3. Results

We compared the following two methods that use different features:

- Baseline: Only using the baseline features.
- +NN: Further using the proposed 4 NN features.

Results are shown in Table 1. We can see that our proposed NN features significantly improve the F-measure, by improving the recall and keeping the precision.
To understand why our proposed method contributed to the recall but not the precision, we analyzed the classification results. We found that our proposed NN features are especially helpful for the instances that the baseline features are ambiguous to make a correct decision. An analysis of several feature values for the following truly parallel sentence pair improved of the NN features is shown in Table 2.

[10]http://code.google.com/p/giza-pp

[11]http://www.csie.ntu.edu.tw/˜cjlin/libsvm

Method	Precision	Recall	F-measure
Baseline	98.58 (4733/4801)	94.66 (4733/5000)	96.58
+NN	**98.58** (4869/4939)	**97.38** (4869/5000)	**97.98**

Table 1: Classification results.

Zh: 为此，可以考虑到目前为止的教育专业没有时间在这一方面进行教学，并教材的开发也在迟延。
Ja: そのためこれまでの教育分野ではこのような側面から教える時間もなく教材開発の遅れに繋がったと考えられる。
Ref: Therefore, it is considered that until now in the field of education there is no time to teach from this aspect and the development of teaching materials is also being delayed.
We can see that the word overlap feature values of the sentence pair are between the average values of the positive and negative instances in the training data. Together with the other features, "Baseline" judges this sentence pair as non-parallel. The proposed NN feature values are greatly lower than the average values of both the positive instances and those of the negative instances, which are very deterministic to judge this sentence pair as parallel. Most of the increased $classified\_well$ sentence pairs (4733→4869) belong to this type, which significantly improved the recall.
On the other hand, low NN feature values also could be harmful. This happens for non-parallel pairs that are short sentences. As the proposed NN features are bilingual language model scores, they are lower for short sentence pairs. These low NN features could lead non-parallel sentence pairs to be judged as parallel, which has badly affect the precision and make it unchanged.

3.3. Extraction and Translation Experiments

We extracted parallel sentences from Wikipedia and evaluated Chinese-to-Japanese MT performance using the extracted sentences as training data.

3.3.1. Settings

- Tuning and testing: We used two distinct sets of 198 parallel sentences with 1 reference in (Chu et al., 2014).[12] These sentences were randomly selected from the sentence pairs extracted from the same Chinese-Japanese Wikipedia data using different methods proposed in (Chu et al., 2014),[13] and the erroneous parallel sentences were manually discarded. Note that for training, we kept all the sentences extracted by different methods except for the sentences duplicated in the tuning and testing sets.
- Decoder: Moses (Koehn et al., 2007) with default options, except for the distortion limit (6→20).
- Language model: 5-gram LM trained on the Japanese Wikipedia (12.5M sentences) using SRILM toolkit.[14]

[12]http://lotus.kuee.kyoto-u.ac.jp/˜chu/resource/wiki_zh_ja.tgz

[13]For more details of the different methods, we recommend the interested readers to refer to the original paper.

[14]http://www.speech.sri.com/projects/srilm

Method	Feature		Value	Avg value (positive)	Avg value (negative)
Baseline	Word overlap	Zh	35%	55%	18%
		Ja	38%	48%	18%
+NN	char NN score	Zh-Ja	97.76	143.66	290.42
		Ja-Zh	89.01	111.60	391.50
	word NN score	Zh-Ja	59.03	111.65	542.28
		Ja-Zh	53.06	104.48	640.81

Table 2: Analysis of feature values for an improved truly parallel sentence pair. Value denotes the feature value of the sentence pair. Avg value (positive)/(negative) denote the average feature value of the positive/negative instances used for training the classifier. Note that the higher values of the word overlap, and the lower NN values indicate the higher parallelism of a sentence pair.

Method	# sentences	BLEU-4
Baseline	126,811	36.31
+NN-ASPEC	110,648	**37.18**
+NN-WIKI	136,013	36.83

Table 3: Parallel sentence extraction and Chinese-to-Japanese translation results.

The other settings are the same as the ones used in the classification experiments.

3.3.2. Results

Parallel sentence extraction and translation results using different methods are shown in Table 3. "Baseline" and "+NN" denote the different methods described in Section 3.2.3.. We compared two types of NN features for extraction, where "+NN-ASPEC" denotes the NN features obtained with the seed parallel corpus, and "+NN-WIKI" denotes the NN features obtained with the parallel sentences extracted by the baseline system.

Although "+NN" improved the recall in the classification experiments, "+NN-ASPEC" actually decreased the number of extracted sentences. We believe that the reason for this is the domain difference between ASPEC and Wikipedia. Actually, we observed that many feature scores for the candidate sentences of "+NN-ASPEC" are infinity,[15] because of too many out-of-vocabulary words. The second line of Table 4 shows the percentage of infinite scores of "+NN-ASPEC" for the character and word based NN features, respectively. In Table 4, we also can see that there are also some infinite scores of "+NN-WIKI". The reason for this is the small size of training (126k) data for "+NN-WIKI". However, because "+NN-WIKI" was obtained with the sentences from Wikipedia, it does not have the domain problem. This led to the lower percentage compared to that of "+NN-ASPEC", and thus the increase of the number of extracted sentences by "+NN-WIKI".

Regarding to the MT performance, we can see that both "+NN-ASPEC" and "+NN-WIKI" outperforms "Baseline", which shows the effectiveness of my proposed NN features for parallel sentence extraction. In addition, "+NN-ASPEC" performs better than "+NN-WIKI". We believe the reason for this is the quality of the NN models.

[15] We replaced the infinity scores to $1,000$ in our experiments.

Method	char NN score	word NN score
+NN-ASPEC	8.44% (351k)	20.69% (861k)
+NN-WIKI	6.73% (280k)	14.58% (607k)

Table 4: Percentage of infinite scores (4.16M in total).

"+NN-ASPEC" was obtained with the NN models trained on the ASPEC corpus, where the parallel sentences are highly accurate. In contrast, "+NN-WIKI" was obtained with the sentences extracted by the baseline system, which are noisy. In addition, the sizes of the training data for obtaining "+NN-ASPEC" and "+NN-WIKI" are also greatly different (680k versus 126k). As NMT is sensitive to the quality and quantity of the training data (Bahdanau et al., 2014), "+NN-ASPEC" outperformed "+NN-WIKI".

4. Related Work

Several studies have exploited the NN features for SMT. (Sutskever et al., 2014) used the NN features to rerank the N-best list of a SMT system, which achieved a BLEU score that is close to the previous state of the art. (Cho et al., 2014) scored the phrase pairs of a SMT system with a neural translation model, and used the scores as additional NN features for decoding. (Dabre et al., 2015) used the NN features for a pivot-based SMT system for dictionary construction. In contrast, we score the sentence pairs of with a neural translation model, and use the scores as NN features for parallel sentence extraction from comparable corpora.

5. Conclusion

In this paper, we incorporated the NN features for parallel sentence extraction from comparable corpora for the first time. Experimental results verified the effectiveness of NN features for this task.

As future work, we plan to address the domain problem of "+NN-ASPEC" by a NN based sentence selection method. Namely, we train NN models on the sentences extracted by the baseline system, and then use these models to select sentences from ASPEC (or other corpora) that are similar to the sentences in Wikipedia. Hopefully, it could further improve the performance of our system.

6. Bibliographical References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Chu, C., Nakazawa, T., Kawahara, D., and Kurohashi, S. (2012). Exploiting shared Chinese characters in Chinese word segmentation optimization for Chinese-Japanese machine translation. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT 2012)*, pages 35–42, Trento, Italy, May.

Chu, C., Nakazawa, T., and Kurohashi, S. (2014). Constructing a Chinese-Japanese parallel corpus from wikipedia. In *Proceedings of the Ninth Conference on International Language Resources and Evaluation (LREC 2014)*, pages 642–647, Reykjavik, Iceland, May.

Dabre, R., Chu, C., Cromieres, F., Nakazawa, T., and Kurohashi, S. (2015). Large-scale dictionary construction via pivot-based statistical machine translation with significance pruning and neural network features. In *Proceedings of the 29th Pacific Asia Conference on Language, Information, and Computation*, Shanghai, China, October.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Kurohashi, S., Nakamura, T., Matsumoto, Y., and Nagao, M. (1994). Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language*, pages 22–28.

Munteanu, D. S. and Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504, December.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.