Bagging-based System Combination for Domain Adaptation

Linfeng Song, Haitao Mi, Yajuan Lü, Qun Liu
Key Laboratory of Intelligent Information Processing
Institute of Computing Technology
Chinese Academy of Science
P.O. Box 2704, Beijing 100190, China
{songlinfeng, htmi, lvyajuan, liuqun}@ict.ac.cn

Abstract

Domain adaptation plays an important role in multi-domain SMT. Conventional approaches usually resort to statistical classifiers, but they require annotated monolingual data in different domains, which may not be available in some cases. We instead propose a simple but effective bagging-based approach without using any annotated data. Large-scale experiments show that our new method improves translation quality significantly over a hierarchical phrase-based baseline by 0.82 BLEU points and it's even higher than some conventional classifier-based methods.

1 Introduction

In recent years domain adaptation problem in SMT becomes more important (Banerjee et al., 2010). Since an SMT system trained on a corpus with heterogeneous topics may fail to achieve a good performance on domain-specific translation, while an SMT system trained on a domain-specific corpus may achieve a deteriorative performance for out-of-domain translation (Haque et al., 2009). Besides more and more evaluation tasks begin to focus on multi-domain translation. For example the NTCIR[1] patent translation task is a multi-domain translation task since its target is to translate sentences from multiple domains such as chemistry, electron, machinery, medicine, material and so on while the NIST[2] evaluation task began to notice the translation of multi-domain web corpus since the year of 2006.

Conventional approaches usually resort to statistical classifiers and there are a plenty of notable jobs on it (Xu et al., 2007; Bertoldi and Federico, 2009; Banerjee et al., 2010). They all achieve significant improvement over baseline when a large amount of annotated monolingual data in multi-domains is available for training the classifier. Yet they shrivel when the annotated monolingual data is deficient. Others proposed unsupervised ways (Hasan and Ney, 2005; Yamamoto and Sumita, 2007) and reported improvement over baseline. Yet their results show that they do not outperform the conventional approaches obviously.

In this paper we propose a novel method for solving domain adaptation problem in SMT. This method is based on a classical ensemble learning technique, Bagging (Breiman, 1996). In more detail, firstly we use bootstrap technique to generate several development sets from the original development set, and then we tune a system for each set, finally for every sentence in the test set we combine the n-best outputs of each individual system and do re-ranking by the voting results of these individual systems. It's obvious that our method does not suffer the drawback of conventional classifier-based methods, besides experiments shows that our method achieves a better performance than some conventional classifier-based methods.

[1] http://research.nii.ac.jp/ntcir/index-en.html

[2] http://www.itl.nist.gov/iad/mig/tests/mt/

In the following parts, firstly we explain the concept of Bagging and its successful applications in NLP areas (section 2), secondly we introduce our Bagging-based domain adaptation method in details (section 3), then we evaluate our method on NTCIR9 Chinese-English Patent corpus and compare our method with some other conventional classifying based methods (section 4), finally we introduce several related works proposed in recent years (section 5, 6).

2 Preliminary

Bagging which is short for bootstrap aggregating is a general ensemble learning technique (Breiman, 1996). Bagging attempts to find a set of classifiers which are consistent with the training data and different from each other. The distribution of aggregated samples approaches the one of samples in the training set.

Given a standard training set $\boldsymbol{D}$ which contains $\boldsymbol{n}$ samples, bagging generates $\boldsymbol{m}$ new training sets $\{\boldsymbol{D_1}, \boldsymbol{D_2}, \ldots, \boldsymbol{D_m}\}$, each with $\boldsymbol{n}$ samples too, by sampling training examples from $\boldsymbol{D}$ uniformly and with replacement. By sampling with replacement, it is likely that for each new training set $\boldsymbol{D_i}$, some training samples of $\boldsymbol{D}$ will be repeatedly chosen while some will not be chosen at all. If $\boldsymbol{n}$, the size of training set $\boldsymbol{D}$, is large enough then each new training set $\boldsymbol{D_i}$ is expected to have 63.2% (Breiman, 1996) of the unique training samples of $\boldsymbol{D}$ and the rest of $\boldsymbol{D_i}$ is just duplicates. For each of the $\boldsymbol{m}$ training sets, one system is trained using it. Finally for each test sample, bagging outputs the voting result from the $\boldsymbol{m}$ systems as its final result.

Bagging has several advantages: firstly it makes the classification more stable, as any single classifier, no matter how strong, cannot perform very well when the distribution of test set is quite different from that of training set. Bagging uses the voting result of $\boldsymbol{m}$ classifiers each with a unique distribution of the same model, so generally it is stable in statistics. Secondly bagging can avoid the over-fitting problem which a plenty of classifiers suffer. Finally bagging can be seen as an unsupervised method which doesn't need the labeled corpus used to train the recognizer in domain recognizing methods.

Bagging has been used successfully in many NLP applications such as Syntactic Parsing (Henderson and Brill, 2000), Semantic Parsing (Nielsen and Pradhan, 2004), Coreference Resolution (Vemulapalli et al., 2009; Vemulapalli et al., 2010), Word Sense Disambiguation (Nielsen and Pradhan, 2004) and so on.

3 Bagging-based domain adaptation

Suppose that there are M available statistics machine systems $\{\mu(\theta_1), \mu(\theta_2), \ldots, \mu(\theta_M)\}$, the task of system combination is to build a new translation system $\nu(\mu(\theta_1), \mu(\theta_2), \ldots, \mu(\theta_M))$ which denotes the combination system. It combines the translation outputs from each of its cell system $\mu(\theta_i)$ which we call here a member system of it.

As discussed in section 1, hardly any single system can achieve a good performance on multi-domain translation problem. Besides, the translation performance heavily relies on the fitness between development set and test set. In this paper, we argue that although any member system $\mu(\theta_i)$ in our bagging based method is unstable, the combination system $\nu(\mu(\theta_1), \mu(\theta_2), \ldots, \mu(\theta_M))$ is stable and can achieve a good translation performance.

3.1 Training

In the training process, each time we bootstrap a new development set $\boldsymbol{D_i}$ from the origin development set $\boldsymbol{D}$. We consider the sentences in $\boldsymbol{D}$ to be the sampling unit and every sentence has the same probability to be chosen, that is we suppose the sampling units in development set $\boldsymbol{D}$ satisfies the uniform distribution.

We use each of the newly generated development set $\boldsymbol{D_i}$ to tune a member system $\mu(\theta_i)$. The training process is nothing special.

3.2 System combination scheme

In the last step of our method, a stable translation system $\nu\big(\mu(\theta_1), \mu(\theta_2), \ldots, \mu(\theta_M)\big)$ is built by the ensemble of the member systems $\{\mu(\theta_1), \mu(\theta_2), \ldots, \mu(\theta_M)\}$. In this part, a sentence-level combination is used to select the best translation from the K-best translation candidates of each member system.

The combination process consists of the following steps:

1) For each input sentence, uses these M member systems to decode it respectively, each system generates the top K-best translation candidates with the respective feature vector

2) Combine the M*K translation candidates and remove all the duplicates to obtain the N unique translation candidates. It should be noticed that two translation candidates are identical only if their translation string and the corresponding feature vector values are identical at the same time

3) For each of the N translation candidates we calculate the total voting score of M systems by simply adding the voting score of each system. The score of each system for each candidate is calculated using the linear combination of the system's weight vector and the candidate's feature vector. The weight vector of each member system is gained in the training process using MERT (Och, 2003). While the feature vector of each translation candidate is gained in step 1). The formula for calculating the total voting score of candidate **c** is listed below:

$$\text{score}(c) = \sum_{t=1}^{M} \overrightarrow{\text{feat}_c} \cdot \overrightarrow{\theta_t}$$

4) Finally we re-rank these N translation candidates by the total voting scores and output the one with the highest score

In order to be concise and to prove the effectiveness of bagging, we do not add any extra features in our implements and experiments.

4 Experiments

4.1 Experimental Setup

We performed the experiments on Chinese-English translation using an in-house implementation of the hierarchical phrase based SMT model (David Chiang, 2005). The model is tuned using standard MERT (Och, 2003).

We use the corpus of NTCIR9 Patent translation task[3] Chinese-English part which contains one million sentence pairs. We obtain one thousand sentence pairs for tuning and testing respectively without overlap. We use GIZA++[4] to perform the bi-directional word alignment between source and target side of each sentence pair. The final word alignment is generated using the grow-diag-final method. And at last all sentence pairs with alignment information is used to extract rules and phrases. A 5-gram language model is trained on the target side of the bilingual data using the SRILM toolkit (Stolcke, 2002). The translation quality is evaluated in terms of non case-insensitive NIST version of BLEU metric[5]. Cube pruning (Huang and Chiang, 2007) are used to prune the search space in the decoding system. In each step of the decoding process we keep the top 200 best hypotheses.

In the bagging process we tune 30 member systems with the same training data and different development sets each of which is the bootstrap of the original development set. We evaluate the fusion results of the first 5, 10, 15, 20 and 30 member systems.

4.2 Effectiveness of Bagging

Table 1 summarizes the experimental results of our bagging based method on the test set. We show the BLEU scores of the 1-best translation and the oracle translation within the fused candidate list of the first N (N=5, 10, 15, 20, 30) member systems.

	1-best	oracle
baseline	31.08	36.74
Bagging-5	31.51	40.35
Bagging-10	31.64*	42.27
Bagging-15	31.73*	42.52
Bagging-20	31.80**	42.74
Bagging-30	31.90**	42.96

Table 1 experimental results (BLEU-4) of bagging on the test set

The above experimental results show that the bagging performance improves stably while the number of member systems increases and the improvement become significant with 0.56 BLEU point gains when we fuse the first 10 member systems. Besides the bagging method achieves improvement of 0.82 BLEU point over baseline when we fuse the total 30 member systems.

[3] http://ntcir.nii.ac.jp/PatentMT/

[4] http://www-i6.informatik.rwth-aachen.de/Colleagues/och/software/GIZA++.html

[5] http://www.itl.nist.gov/iad/mig/tests/mt/2009/

Meanwhile the BLEU point of the oracle translation reaches 42.96 when we fuse the total 30 member systems, and this is 6.22 higher over baseline. We believe that this notable improvement of oracle BLEU score owes to the tremendous diversity among member systems. Later in this paper we will prove that this diversity comes from the bagging method rather than some random factors.

4.3 Prove the effectiveness of Bagging

To prove that the above improvement does come from bagging rather than some random factors, we made the following experiments: firstly we tune 30 baseline subsystems with different random initial weights, and then we evaluate the BLEU scores just like section4.2. The results are shown in Table 2 below:

	1-best	oracle
Baseline	31.08	36.74
Random-5	31.11	38.35
Random-10	31.13	38.67
Random-15	31.17	38.82
Random-20	31.23	39.04
Random-30	31.20	39.25

Table 2 experimental results (BLEU-4) of random factors on the test set

From the above results, we can see that random factors have little effect on improvement translation quality. While the slight increase of oracle BLEU score, which is just 2.51 points, shows that random factors lead to little diversity. So it can be concluded that the diversity of experimental results in previous section does come from our bagging based method rather than other random factors.

4.4 Comparison with supervised domain recognizing methods

We also investigate the effectiveness of domain recognizing based approaches for domain adaptation. In this section we will investigate one supervised method and in the next section we will investigate another unsupervised way to make a comparison with ours.

The supervised domain recognizing methods we investigated here needs a statistical classifier as the domain recognizer. As there is no annotation in the NTCIR corpus to tell us which domain each sentence belongs to, we can just train this classifier by using our in-house patent corpus that has a similar style with the NTCIR corpus. Our in-house corpus contains data from five domains which are chemistry, electron, machinery, medicine, material. The detailed steps are list below:

1) Uses the in-house patent corpus to train a five-class classifier;
2) Uses this classifier to classify the development set and test set into five classes
3) Then for each class of the development set, we use it to tune a system
4) In decoding step, for every sentence in the test set, we translate it using the relative system; finally, we evaluate the translation results of the whole test set.
5) If the number of remnant classes is less than two the process is over. Otherwise we combine the two classes in the development set with the least amount of data to form a new class. Besides, we combine the relevant classes of test set and then go to step 3)

The results are listed below in Table3

classes	1	2	3	4	5
BLEU	31.08	31.36	31.63	31.49	30.90

Table 3 results of the supervised domain recognizing based method

From the above result we can see that: when the class number is five, the translation performance drops below the baseline due to the data sparsity problem; as the class number decreases, the performance ascends and reaches the highest record of 31.63 BLEU score when the class number is three; when the class number is two, the performance begins to drop as the weak discrimination of the two-class recognizer; finally when there is only one class, this method degenerates to the baseline.

From the above results we see a trivial improvement of BLEU score which makes it less competitive with our method.

4.5 Comparison with unsupervised domain recognizing methods

In this section, we investigate the effectiveness of an unsupervised domain recognizing method that is similar to Yamamoto et al. (2007). To avoid data sparsity we just cluster the target part of bilingual corpus to form domain specific language model for

each subsystem, while the subsystems share one general rule table extracted from the whole training data.

We test the results with the cluster number from 2 to 5, and the results are listed below:

clusters	2	3	4	5
BLEU	31.09	31.24	31.05	30.61

Table 4 results of the unsupervised domain recognizing based method

From the above results we can see a similar situation: when there are too many clusters the translation performance drops due to data sparsity; and as the cluster number decreases, the performance ascends at first and reaches the highest record of 31.24 BLEU score when the cluster number is three; and finally drops as the discrimination of class recognizer becomes weak.

5 Related Work

Langlais (2002) first mention Domain Adaptation problem in SMT area by mention the problem of how to use a SMT to translate a corpus far different from the one it has been trained on. Then he makes notable achievement by integrating specific lexicon tables.

Eck et al. (2004) proposed a language model adaptation technique in SMT using information retrieval techniques. Firstly, each test document is translated with general language model; and then the translation is used to select the most similar documents; then the adapted language model is built using these documents; finally the test document is re-translated using the adapted language model.

Hasan and Ney (2005) proposed a method for building class-based language models. He applies regular expressions based method to cluster the sentences into specific classes. And then he interpolates them with the main language models to elude the data sparseness. And finally this method achieves improvements in terms of perplexity reduction and error rates.

Koehn and Schroeder (2007) carried out a scheme of integrating in-domain and out-of-domain language models using log-linear features of an SMT model, and used multiple decoding paths for combining multiple domain translation tables within the framework of the Moses decoder[6].

Xu et al. (2007) proposed a method which uses the information retrieval approaches to classify the input sentences. This method is based on domains along with domain-dependent language models and feature weights which are gained in the training process of SMT models. This method resulted in a significant improvement in domain-dependent translation.

6 Conclusion and Future Work

We have proposed a bagging-based system combination scheme to address the issue of multi-domain translation problem. Our method combines several member systems to form a new enhanced system which is stable and well-performing in domain adaptation. Experimental results show that our method is effective in improving the translation accuracy on multi-domain SMT.

Acknowledgments

The authors were supported by National Natural Science Foundation of China Contract 60736014 and 60873167. We are grateful to the anonymous reviewers for their valuable comments. And we would like to thank Yun Huang, Fendong Meng, Jun Xie, Jinsong Su and Hao Xiong for their valuable suggestions on this paper.

References

Leo Breiman. 1996. *Bagging predictors.* Machine Learning, 24(2):123-136, 1996.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. *BLEU: a method for automatic evaluation of machine translation.* In Proceedings of ACL 2002, pages 311-318.

Stolcke, A. 2002. *SRILM-an extensible language modeling toolkit.* In Proceedings of ICSLP 2002, pages 901-904.

Franz J. Och. 2003. *Minimum error rate training in statistical machine translation.* In Proceedings of ACL 2003, pages 160-167.

David Chiang. 2005. *A hierarchical phrase-based model for statistical machine translation.* In Proceedings of ACL 2005, pages 263-270.

[6] http://www.statmt.org/moses/

Liang Huang and David Chiang. 2007. *Forest rescoring: Faster decoding with integrated language models.* In Proceedings of ACL 2007, pages 144-151.

John C. Henderson and Eric Brill. 2000. *Bagging and Boosting a Treebank Parser.* In Proceedings of NAACL 2000, pages 34-41.

R. D. Nielsen and S. Pradhan. 2004. *Mixing weak learners in semantic parsing.* In Proceedings of EMNLP 2004, pages 80-87.

Smita Vemulapalli, Xiaoqiang Luo, John F. Pitrelli and Imed Zitouni. 2009. *Classifier Combination Techniques Applied to Coreference Resolution.* In Proceedings of NAACL HLT Student Research Workshop and Doctoral Consortium 2009, pages 1-6.

Smita Vemulapalli, Xiaoqiang Luo, John F. Pitrelli and Imed Zitouni. 2010. *Using Bagging and Boosting Techniques for Improving Coreference Resolution.* Informatica 34: 111-118, 2010

Nielsen, Rodney and Sameer Pradhan. 2004. *Mixing weak learners in semantic parsing.* In Proceedings of EMNLP 2004, pages 80-87.

Langlais, P. 2002. *Improving a general-purpose statistical translation engine by terminological lexicons.* In Proceedings of Coling: Second international workshop on computational terminology, pages 1-7.

Eck, M., S. Vogel and A. Waibel 2004. *Language model adaptation for statistical machine translation based on information retrieval.* In Proceedings of LREC 2004, pages 327-330.

Hasan, S. and H. Ney. 2005. *Clustered language models based on regular expressions for SMT.* In Proceedings of EAMT 2005, pages 133-142.

Koehn, P. and J. Schroeder. 2007. *Experiments in domain adaptation for statistical machine translation.* In Proceedings of the Second Workshop on Statistical Machine Translation, pages 224-227.

Xu J., Y. Deng, Y. Gao, and H. Ney. 2007. *Domain dependent statistical machine translation.* In Proceedings of the MT Summit XI, pages 515-520.

Haque, Rejwanul, Sudip Kumar Naskar, Josef van Genabith and Andy Way. 2009. *Experiments on Domain Adaptation for English-Hindi SMT.* In Proceedings of PACLIC 23, pages 670-677.

Pratyush Banerjee, Jinhua Du, Baoli Li, Sudip Kr. Naskar, Andy Way and Josef van Genabith. 2010. *Combining multi-domain statistical machine translation models using automatic classifiers.* In Proceedings of AMTA 2010.

Hirofumi Yamamoto and Eiichiro Sumita. 2007. *Bilingual cluster based models for statistical machine translation.* In Proceedings of EMNLP-CoNLL 2007, pages 514-523.