A Common Case of Jekyll and Hyde: The Synergistic Effect of Using Divided Source Training Data for Feature Augmentation

Yan Song
City University of Hong Kong
83, Tat Chee Ave., Kowloon
Hong Kong, China
clksong@gmail.com

Fei Xia
University of Washington
PO Box 354340
Seattle, WA 98195, USA
fxia@uw.edu

Abstract

Feature augmentation is a well-known method for domain adaptation and has been shown to be effective when tested on several NLP tasks (Daume III, 2007). However, a limitation of the method is that it requires labeled data from the target domain and very often such data is unavailable. In this paper, we propose to use training data selection to divide the source domain training data into two parts, pseudo target data (the selected part) and source data (the unselected part), and then apply feature augmentation on the two parts of the training data. This approach has two advantages: first, feature augmentation can be applied even when there is no labeled data from the target domain; second, the approach can take advantage of all the training data including the part that is not selected by training data selection. We evaluate the approach on Chinese word segmentation and part-of-speech tagging and show that it outperforms the baseline where no feature augmentation is applied.

1 Introduction

The goal of domain adaptation is to alleviate the degradation of NLP systems when training and test data are from different domains. There have been many approaches to domain adaptation, and two of well-known ones are feature augmentation and training data selection. Feature augmentation makes three copies of each feature in the original feature set (one for the source domain, one for the target domain, and one for the general domain) so that features appeared in the source and the target domains can be differentiated in case they behave differently in the two domains; the method has been shown to be effective for several NLP tasks (Daume III, 2007). However, a limitation of the method is that it requires labeled data in the target domain, a condition that is hard to meet when creating labeled data in the target domain is expensive and time-consuming.

Training data selection addresses the differences between the source and target domains by choosing a subset of the training data in the source domain that is similar to the data in the target domain. When the amount of source training data is large, this method often provides better performance than using the entire training data (Moore and Lewis, 2010; Axelrod et al., 2011; Plank and van Noord, 2011; Song et al., 2012). However, when the amount of the training data is small, the selected subset is unlikely to outperform the entire training data because the trained model cannot benefit from unselected labeled data.

To address the limitations of both methods, we propose to divide the whole source training data into two subsets via training data selection. We then treat the selected subset as coming from a *pseudo target domain* (i.e., a pseuodo domain that is similar to the target domain) and keep the unselected data in the source domain. Now we have labeled data from both domains, we can apply feature augmentation in the usual way; that is, we distinguish features from the source domain and the ones from the pseudo target domain. Notice that the 'unselected subset' is also used by the trainer, unlike the standard training data selection method where the unselected part is totally discarded by the trainer. In addition, we propose a coverage-based measure for training data selection. We evaluate our approach on two NLP tasks, Chinese word segmentation (CWS) and part-of-speech (POS) tagging, and show that it outperforms the systems which use the entire training data without training data selection or feature augmentation.

The reminder of this paper is organized as fol-

International Joint Conference on Natural Language Processing, pages 623–631,
Nagoya, Japan, 14-18 October 2013.

lows. Section 2 presents previous work on training data selection and feature augmentation. Section 3 describes our approach in details and introduces a coverage-based measure for training data selection. Section 4 reports experimental results on two NLP tasks with discussion on the results.

2 Related Work

Two main aspects of our work are dividing training data and applying feature augmentation. In this section, we discuss related work in these aspects.

2.1 Training Data Selection

Training data selection is a common approach to domain adaptation. Moore and Lewis (2010) proposed to rank training sentences according to the difference of the cross entropy values of a given sentence, and showed that training data selection improved the performance of statistical machine translation systems. Axelrod et al. (2011) used cross entropy in three ways: the first one directly measured cross entropy for the source side of the text; the second one was similar to (Moore and Lewis, 2010) and ranked the data using cross entropy difference; the third one took into account the bilingual data on both the source and the target side of translations. Both studies showed that the selected subset of training data worked better than the entire training corpus for machine translation. In addition to these studies, there has been other work (e.g., (Eck et al., 2005; Munteanu and Marcu, 2005; Hildebrand et al., 2005; Lu et al., 2007)) that shows training data selection is an effective way to improve MT.

Plank and van Noord (2011) experimented with several training data selection methods to improve the performance of dependency parsing and POS tagging. These methods fell into two categories: probabilistically-motivated and geometrically-motivated. Their experiments demonstrated that the proposed training data selection methods outperformed random selection.

In our previous study (Song et al., 2012), we proposed several entropy-based measures for training data selection, including averaged entropy gain (AEG), cross entropy, difference of entropy, and description length gain (DLG)-based measures. Among them, AEG worked well on CWS and POS tagging and outperformed other measures including difference of cross entropy. In this study, we are using the same data sets as in that study and we will compare our new coverage-based measure with AEG.

2.2 Feature Augmentation

Feature augmentation (Daume III, 2007) is a well-known domain adaptation method in the supervised setting, when labeled data exist for both source and target domains. The idea is to distinguish instances from the source and target domains by making three copies of each original feature: one copy for the source domain, one copy for the target domain and a third copy for the general domain that contains both the source and target domains. Daume evaluated the method on several sequence labeling tasks (e.g., named entity recognition, POS tagging and shallow parsing) and showed that this method outperformed several baselines and previous approaches. The method is easy to implement and does not require modifications to the trainer.

3 Our Approach

In order to perform feature augmentation on the whole training data, the very first step is to split the training data into two subsets. Training data selection is an effective way to choose a subset from the whole source domain data that is similar to the target domain. The question is what measures should be used for calculating similarity between a source sentence and the target domain. In this section, we discuss some existing entropy-based measures and propose a novel coverage-based measure. Then we explain how we apply feature augmentation to the two subsets.

3.1 Entropy-based Measures

Among the existing similarity measures used by training data selection, many of them focus on the similarity of probability distributions from the training and test data and use entropy-based formulas (Moore and Lewis, 2010; Axelrod et al., 2011; Song et al., 2012). Cross entropy is the most prevailing metric to evaluate the probability distribution similarity between a training sentence and the test data. Eq. 1 shows the formula for cross entropy for a language (marked as CEL, as in the context of evaluating a language model), where n is the length of sentence s, p is an ngram language model, and x_i represents the i-th word in the sen-

tence given the previous words.

$$CEL(s,p) = -\frac{1}{n}\sum_{i=1}^{n}\log p(x_i) \quad (1)$$

The difference of cross entropy (DCE) for a sentence s is formulated as

$$DCE(s,p,q) = |CEL(s,p) - CEL(s,q)| \quad (2)$$

where p and q are two language models, built from the source domain and the target domain respectively. For training data selection, sentences are sorted by DCE scores and the ones with low scores are considered to be similar to the target domain (Moore and Lewis, 2010; Axelrod et al., 2011).

Another well-performed measure is AEG (Song et al., 2012). Let C be a corpus and s be a sentence from the source domain; we define entropy gain (EG) of s according to C as in Eq 3, where q is a probability distribution estimated from C and $q1$ is one estimated from $C+s$, a new corpus formed by adding s to C. Intuitively, if s is similar to C, $q1$ will be very similar to q and $EG(s,c)$ will be small.

$$EG(s,C) =\mid H(C+s,q1) - H(C,q) \mid \quad (3)$$

$H(X,p)$ follows the standard definition of entropy in information theory, where X is a discrete random variable with m possible outcomes $\{x_1,...,x_m\}$ and p is a probability distribution of X. Given a corpus C, one can collect a set of ngrams (in words or characters) from C and X is then derived from the set.

$$H(X,p) = -\sum_{i=1}^{m} p(x_i)\log p(x_i) \quad (4)$$

Average entropy gain (AEG) is EG normalized by sentence length, shown in Eq 5.

$$AEG(s,C) = \frac{EG(s,C)}{length(s)} \quad (5)$$

3.2 Coverage-based Data Selection

We propose a coverage-based measure, which differs from the entropy-based measures in two aspects. First, this measure uses ngram coverage, not probability similarity, as the criterion for selecting training data. The rationale is that we would like the selected data to have a good coverage of the test data, because in many NLP tasks, especially in CWS and POS tagging, out-of-vocabulary (OOV) is a main problem affecting system performance and the problem is more severe when the training and test data come from different domains. Second, existing training data selection methods (such as the ones listed in Section 3.1) select the current sentence without considering the effect of adding that to the previously selected sentences. Our method tackles this problem by considering the overall effect of the selected subset. As checking all the subsets is computationally expensive, we use a greedy search to find the best training sentence based on the current selected subset.

The coverage-based data selection is presented in Algorithm 1. Here, L, T, and p refer to the original training data, test data and the proportion (in percentage) of training data to be selected. L_s and L_u are the output, which refer to the selected and unselected subsets of the training data respectively. By conducting such selection method, training data is divided into two parts.

Algorithm 1 Coverage-based data selection.

Input: L, T, p
Output: L_s, L_u

```
L_s = φ, L_u = L
while Sizeof(L_s) < Sizeof(L) * p do
    for each sentence s_i in L_u do
        compute cov(L_s ∪ {s_i}, T)
    end for
    id = argmax_i cov(L_s ∪ {s_i}, T)
    L_s = L_s ∪ {s_id}, L_u = L_u − {s_id}
end while
return L_s, L_u
```

In Algorithm 1, coverage function $cov(C,T)$ represents the coverage of ngrams in a test data T given a corpus C, as shown in Eq. 6. Here, ng is an ngram[1] and $NgramSet(T)$ refers to the set of ngram types in T, and the denominator $\mid NgramSet(T) \mid$ is the size of the set.[2]

$$cov(C,T) = \frac{\sum_{ng\in NgramSet(T)} count(ng,C)}{\mid NgramSet(T) \mid} \quad (6)$$

To handle the problem of data sparsity, we use the following back-off counting method to find

[1]Where the units for composing an ngram are different with respect to different tasks, i.e., they are characters in word segmentation and words in POS tagging.

[2]We also investigated using ngram tokens for coverage computation; we will include the comparison of ngram types and ngram tokens in the final version of this paper.

partial covered low order ngrams inside the high order ngram. The idea of such ngram counting is similar to back-off methods in language modeling. Given an ngram $t_{i-n+1}...t_{i-1}t_i$ in T, we calculate the $count(\ )$ function as in Eq. 7. α is used to determine the value of the "partial credit" given to a substring of the ngram appearing in C. The value of α is set to 0.5 empirically.[3]

$$count(t_{i-n+1}...t_{i-1}t_i, C) = \quad (7)$$
$$\begin{cases} 1, & if\ \ t_{i-n+1}...t_{i-1}t_i\ \ appears\ in\ C \\ \alpha \cdot count(t_{i-n+2}...t_{i-1}t_i), & otherwise \end{cases}$$

In Eq. 7, t_i is a token in the ngram, i.e., a character in the CWS task and a word in the POS tagging task. If a high order ngram is not found in C, the $count(\ )$ function is called recursively until a shorter ngram inside the original ngram is found. The value of the $count(\ )$ function is zero only if the token t_i itself is an OOV. For the experiments in this paper, we use trigram to count the ngram coverage.

3.3 Feature Augmentation

As we mentioned before, a limitation of feature augmentation (Daume III, 2007) is that it requires labeled data from the target domain, and very often such data is not available. To overcome this limitation, we use training data section on the source domain data, treat the selected part of data as from a *pseudo target domain*, and leave the unselected part in the source domain. Then a feature augmentation is performed on such two "new" domains; that is, it makes three copies of each original feature: f_s for the source domain, f_t for the target domain, and f_g for the general domain. Following Daume (Daume III, 2007), the general domain is simply the union of the source and the target domains. In this case, the target domain refers to our pseudo target domain; the features associated to the pseudo target domain and the test data are augmented as in Eq. 8, and the features associated to the unselected source domain data are shown in Eq. 9.

$$f \rightarrow < f_g, 0, f_t > \quad (8)$$

$$f \rightarrow < f_g, f_s, 0 > \quad (9)$$

Another potential issue with feature augmentation is that making several copies of all the features could worsen the problem of data sparsity. It is worth exploring whether duplicating only certain features would produce better performance than duplicating all the features. To test out the idea, we ran another set of experiments where only unlexicalized features (e.g., word type, POS tags of previous words) are duplicated. The experimental results in Section 4 confirmed our intuition and showed that augmenting only unlexicalized features works better.

4 Experiments

In this study, we ran several sets of experiments. We compared our training data selection with other methods, and then evaluated our revised feature augmentation method on the CWS and POS tagging tasks.

4.1 Data

The Chinese Penn Treebank (CTB) version 7.0[4] (Xia et al., 2000) is used in our experiments. It contains about 1.2 million words from five genres: Broadcast Conversation (BC), Broadcast News (BN), Magazine (MZ), Newswire (NW), and Weblog (WB). The details of the five genres of CTB 7.0 are shown in Table 1.

We divide the data in each genre into ten folds based on character counts, and use the first eight folds for training, the next fold for development, and the last fold for testing. In order to make the size of the training data for each genre to be the same, we set the training size to be the size of the training folds in the BC genre (the smallest genre in the CTB 7.0). We do the same for the development data. For testing, we use the whole test fold for each genre. The sizes of the data sets used in the experiments are shown in Table 2.[5]

Without loss of generality, we use BC and NW as the test genres; for each test genre, we use the union of training folds from other four genres as the training data.

[3] We tried different value of α in ranging from 0 to 1, where $\alpha = 0$ means there is no back-off. The results indicate that when $\alpha = 0$, selection performance is much worse than the case $\alpha > 0$, while when $\alpha > 0$, selection performance varies so little by using different values of α.

[4] Linguistic Data Consortium No. LDC2010T07

[5] Although we are not using the development fold for the experiments in this study, we still split the data into training, development, and test folds to facilitate comparison with other studies that use the same data split.

Genre	# of chars	# of words	# of files	Sources
Broadcast Conversation (BC)	275,289	184,161	86	China Central TV, CNN, MSNBC, Phoenix TV, etc.
Broadcast News (BN)	482,667	287,442	1,146	China Broadcasting System, China Central TV, China National Radio, Voice of America, etc.
Magazine (MZ)	402,979	256,305	137	Sinaroma
Newswire (NW)	442,993	260,164	790	Xinhua News, Guangming Daily, People's Daily, etc.
Weblog (WB)	342,116	208,257	214	Newsgroups, Weblogs
Total	1,946,044	1,196,329	2,373	

Table 1: Statistics of the CTB 7.0.

	BC	BN	MZ	NW	WB
Training	211,795	211,826	211,834	211,853	211,796
Development	30,678	30,760	30,708	30,726	30,746
Test	32,816	48,317	37,531	44,543	33,623

Table 2: Statistics of training, development, and test portions of each genre in CTB 7.0. The numbers are character counts.

4.2 Training Data Selection

To demonstrate our coverage-based training data selection method, we first compare its performance on POS tagging with other two methods, AEG (Song et al., 2012) and random selection.[6] The selected proportion of training data range from 10% to 90%, based on character counts. Here, we use Stanford POS Tagger (Toutanova et al., 2003). The results on BC and NW are shown in Table 3 and 4, with comparison to random selection methods.[7]

Our coverage-based training data selection method outperforms random selection on both BC and NW. It also outperforms AEG when a low percentage of data is selected, while its performance is comparable or slightly lower than AEG when a higher percentage of data is selected. To understand this behavior, we compare some statistics of the data sets, as in Table 5.

Since OOV rate is important for CWS and POS tagging, we want to compare our coverage-based method and AEG for this factor, and the results are presented in Table 6.

The table shows that when a small percentage (e.g., 10%, 20%) of source-domain data is selected, the OOV rate of the test data is much lower when Cov is used. In contrast, when a large percentage (e.g., 80% and 90%) of training data is selected, the OOV rates are similar between Cov and AEG. This could be the reason why Cov outperforms AEG when a small percentage of training data is selected, but not so when more training data is selected.

For the rest of the experiments, we will use Cov for training data selection and test whether our revised feature augmentation approach provides some improvement for CWS and POS tagging.

4.3 Chinese Word Segmentation

To evaluate feature augmentation on CWS, we use a conditional random fields (CRF) word segmenter as described in (Song and Xia, 2012). A nice property of the segmenter is that it incorporates unsupervised learning to identify possible new words in the test data in order to enhance the segmenter's performance on OOVs. To be more specific, the segmenter uses description length gain (DLG) (Kit and Wilks, 1999) for lexical acquisition as that was performed in (Kit, 2000; Kit, 2005). Then the decision of the unsupervised word segmentation is represented as features T_0^i, which indicates the tag of the current character C_0 when it belongs to a word whose length i ranges from 1 to 5 charac-

[6] Song et al. (2012) showed that AEG works better than cross entropy, as well as difference of cross entropy, on CWS and POS tagging. Therefore we only compare our method with AEG in this paper.

[7] For each percentage, the result of random selection are the average of three runs of random selection.

Percentage	Cov	AEG	RDM
10%	**90.08**	89.61	88.60
20%	**91.13**	91.01	89.74
30%	**91.40**	**91.40**	90.59
40%	**91.70**	91.67	91.25
50%	91.89	**91.94**	91.37
60%	92.24	**92.31**	91.84
70%	92.40	**92.53**	91.84
80%	**92.43**	92.41	92.11
90%	**92.48**	92.45	92.22
100%	92.30	92.30	92.30

Table 3: Performance of Stanford POS tagger when tested on BC and trained on the other four genres. The largest number in each row is in bold. Cov, AEG and RDM refer to our coverage-based method, Average entropy gain and random selection.

Percentage	Cov	AEG	RDM
10%	**89.97**	87.73	87.53
20%	**91.15**	89.64	89.23
30%	**91.73**	90.74	90.31
40%	**91.91**	91.41	91.32
50%	**92.21**	91.86	91.38
60%	**92.18**	92.03	91.63
70%	**92.32**	92.19	91.90
80%	92.41	**92.45**	92.28
90%	**92.51**	92.48	92.33
100%	92.56	92.56	92.56

Table 4: Performance of Stanford POS tagger when tested on NW and trained on the other four genres. The largest number in each row is in bold. Cov, AEG and RDM refer to our coverage-based method, Average entropy gain and random selection.

Test genre	BC	NW
Tokens in training	536,356	533,594
Tokens in test	22,088	25,916
OOV tokens	1,034	1,986
OOV rate	4.68%	7.66%

Table 5: Statistics (in words) of the entire training and test data for BC and NW.

	BC		NW	
%	Cov	AEG	Cov	AEG
10%	9.04%	11.53%	14.27%	19.61%
20%	5.22%	8.21%	10.19%	14.59%
80%	4.76%	5.19%	7.66%	8.18%
90%	4.68%	4.85%	7.66%	7.94%

Table 6: The OOV rate (in words) when a different percentage (10%, 20%, 80% and 90%) of training data is selected by coverage-based method (Cov) and AEG against test data on BC and NW.

ters. These features are added to the standard feature set for supervised learning. The new feature set is in Table 7, where the subscript -1, 0, and +1 refer to the previous, current and next character, respectively.

Description	Features
Char Unigrams	C_{-1}, C_0, C_{+1}
Char Bigrams	$C_{-1}C_0, C_0C_{+1}, C_{-1}C_{+1}$
DLG Features	$T_0^1, T_0^2, T_0^3, T_0^4, T_0^5$

Table 7: Feature template of our CRF segmenter.

For feature augmentation, we compare two settings: one duplicates all the features and the other duplicates only the unlexicalized features. The results when tested on BC are in Table 8. It shows that augmenting unlexicalized features provides better performance than augmenting all features. For the rest of experiments, feature augmentation will duplicate only the unlexicalized features.

Table 9 shows the performance of using feature augmentation on CWS when tested on NW. Table 8 and 9 both show that our approach on divided training data improves system performance significantly (e.g., over 0.6% when tested on BC) without using any external resources. For Tables 9 and 11, we use a ten-partition two-tailed paired Student t-test for significance test.

4.4 POS Tagging

To evaluate feature augmentation on POS tagging, we used an in-house CRF tagger.[8] Table 10 shows the feature set used by the tagger, where subscript -1, 0, and +1 refer to the previous, current and

[8]The reason that we use our in-house CRF POS tagger, instead of the Stanford POS tagger, is that we have not found an easy way to extend Stanford POS tagger to support feature augmentation.

% of data selected	Unlex. Feat. Aug.			All Feat. Aug.		
	F	P	R	F	P	R
Baseline	94.10	93.87	94.34	94.10	93.87	94.34
10%	**94.70**	94.30	95.09	93.71	93.43	94.00
20%	**94.72**	94.35	95.09	94.06	93.98	94.14
30%	**94.62**	94.23	95.01	**94.19**	94.07	94.31
40%	**94.51**	94.07	94.96	**94.11**	93.96	94.26
50%	**94.51**	94.08	94.94	93.96	93.80	94.12
60%	94.10	93.77	94.43	93.96	93.86	94.06
70%	**94.16**	93.86	94.46	94.06	93.99	94.12
80%	94.08	93.80	94.37	93.88	93.81	93.95
90%	94.08	93.84	94.32	93.90	93.60	94.20

Table 8: Performance of feature augmentation on CWS, with unlexicalized and all features augmented. The pseudo target data is selected by coverage-based method. The segmenter is tested on *BC*, and trained on the other four genres in CTB 7.0. F-score (F), Precision (P) and Recall (R) are presented. F-scores higher than the baseline are in bold.

%	F	P	R
Baseline	93.70	93.90	93.50
10%	**93.82**	93.97	93.66
20%	**93.90***	94.07	93.73
30%	**93.90***	94.05	93.76
40%	**93.92****	94.06	93.78
50%	**93.89***	94.07	93.71
60%	**93.91****	94.09	93.72
70%	**93.91****	94.07	93.76
80%	**93.89***	94.06	93.72
90%	**93.84**	94.03	93.64

Table 9: Performance of feature augmentation on CWS, with unlexicalized features augmented. The pseudo target data is selected by coverage-based method. The segmenter is tested on *NW*, and trained on the other four genres in CTB 7.0. F-score (F), Precision (P) and Recall (R) are presented. F-scores higher than the baseline are in bold. Symbols * and ** indicate significance at p=0.05 and p=0.01 against the baseline, respectively.

next word, respectively. This feature set is similar to the one used in the Stanford POS tagger, but our tagger does not include some hard coded treatment and rules (e.g., bidirectional transition rules) used by the Stanford tagger. As a result, the performance of our tagger is slightly lower than the Stanford tagger. For instance, when tested on *BC* and trained on the other four genres, the tagging accuracy of our tagger is 91.95%, compared to 92.30% by the Stanford tagger (see the last row in Table 11 and Table 3).

Description	Features
Word Unigrams	W_{-1}, W_0, W_{+1}
Word Bigrams	$W_{-1}W_0, W_0W_{+1}, W_{-1}W_{+1}$
Word Prefix	P_0
Word Suffix	S_0
Word Prefix Type	TP_0
Word Suffix Type	TS_0

Table 10: Feature template of our CRF POS tagger.

Table 11 shows the results of POS tagging with feature augmentation. The test genre is BC or NW, and the training data come from the other four genres. The first row lists the percentage of training data chosen by our coverage-based training data selection. The baseline shows the performance of our CRF tagger when the whole training set is used without training data selection and feature augmentation. In the table, the higher-than-baseline tagging accuracy in each test are marked in boldface. Similar to CWS, training data selection followed by feature augmentation improves the performance of the POS tagger.

4.5 Discussion

In all, there are several observations from Tables 8 and 9 for CWS, and Table 11 for POS tagging. First, there is a small, but statistically significant, improvement when we treat selected and unselected data as two domains and apply fea-

Percentage	BC	NW
10%	**92.21**	92.21
20%	**92.31***	**92.41**
30%	**92.40****	**92.52***
40%	**92.39****	**92.48***
50%	**92.44****	**92.44**
60%	**92.43****	**92.42**
70%	**92.45****	**92.38**
80%	**92.40****	92.33
90%	**92.31***	92.31
baseline	91.95	92.36

Table 11: Performance of our POS tagger with feature augmentation when tested on BC and NW. Numbers presented in the table are tagging accuracy, and the ones higher than the baseline are in bold. Symbols * and ** indicate significance at p=0.05 and p=0.01 against the baseline, respectively.

ture augmentation (e.g., 91.95% vs. 92.45% on BC in Table 11). Second, duplicating only a subset of features outperforms duplicating all the features, as the large number of features for the latter strategy could aggravate the data sparsity problem. Augmenting some features (e.g., lexicalized) could actually hurt the performance. Third, with regard to the percentage of training data selected for the pseudo target domain, system performance improves when the percentage of selected data increases from 10% up to a certain point (70% for testing on BC and 30% for testing on NW on POS tagging), and afterwards it starts to degrade because newly added pseudo target domain data is no longer quite similar to the target domain. The optimal size of the selected subset may depend on how similar the training data is to the test data. Fourth, when comparing CWS and POS tagging, we can find the same trend in feature augmentation across different tasks. That is, when feature augmentation on CWS has higher improvement, usually it also brings higher improvement on POS tagging when comparing across different test data (e.g., the improvement on BC is higher than NW for CWS, and the same is true for POS tagging).

5 Conclusion

This study has made two contributions to domain adaptation. First, we proposed an approach that combines training data selection and feature augmentation. It tackles the limitations of both feature augmentation and training data selection methods as it does not require labeled data from the target domain while it takes advantage of the entire training data. Consequently, it significantly improves system performance over the baseline. We also demonstrate that augmenting some features works better than augmenting all the features because the latter setting triples the number of features which could lead to severe data sparsity problem. Our experimental attempts confirmed the fact that augmenting less-sparse features (unlexicalized one, e.g., prefix and suffix, character type) led to better performance than all features. Second, we proposed a new measure for training data selection, which selects training sentences to maximize the coverage of ngrams on the test data. It showed a better performance than other measures especially when a small subset of training data is selected. The approaches has been evaluated on two NLP tasks, namely, Chinese word segmentation and part-of-speech tagging. Both tasks confirmed the effectiveness of our approaches and yield better performance than the baseline settings.

For future work, we would like to apply automatic feature selection to determine what kind of features should be duplicated to boost the benefits of feature augmentation. We would also like to evaluate our approach on other NLP tasks, and test its performance with other machine learning algorithms.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of EMNLP2011*, pages 355–362.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 256–263.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Low cost portability for statistical machine translation based on n-gram coverage. In *Proceedings of MT Summit X*, pages 227–234.

Almut Silja Hildebrand, Matthias Eck, and Stephan Vogel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of EAMT 2005*, pages 133–142.

Chunyu Kit and Yorick Wilks. 1999. Unsupervised learning of word boundary with description length gain. In *Proceedings of CoNLL-99*, pages 1–6.

Chunyu Kit. 2000. *Unsupervised Lexical Learning as Inductive Inference*. Ph.D. thesis, University of Sheffield.

Chunyu Kit. 2005. Unsupervised lexical learning as inductive inference via compression. In J. W. Minett and W. S.Y. Wang, editors, *Language Acquisition, Change and Emergence*, pages 251–296.

Yajuan Lu, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of EMNLP-CoNLL2007*, pages 343–350, Prague, Czech Republic, June. Association for Computational Linguistics.

R.C. Moore and W. Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of ACL2010, Short Papers*, pages 220–224.

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

B. Plank and G. van Noord. 2011. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 1566–1576.

Yan Song and Fei Xia. 2012. Using a goodness measurement for domain adaptation: A case study on chinese word segmentation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3853–3860, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1580.

Yan Song, Prescott Klassen, Fei Xia, and Chunyu Kit. 2012. Entropy-based training data selection for domain adaptation. In *Proceedings of COLING 2012*, pages 1191–1200, Mumbai, India, December. The COLING 2012 Organizing Committee.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL)*, pages 173–180.

Fei Xia, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Fudong Chiou, Shizhe Huang, Tony Kroch, and Mitch Marcus. 2000. Developing Guidelines and Ensuring Consistency for Chinese Text Annotation. In *Proceedings of the Second Language Resources and Evaluation Conference (LREC)*.