Using a Goodness Measurement for Domain Adaptation: A Case Study on Chinese Word Segmentation

Yan Song*† and Fei Xia*
*University of Washington
Seattle, WA 98195, USA
†City University of Hong Kong
83, Tat Chee Ave., Kowloon, Hong Kong
E-mail: {yansong, fxia}@uw.edu

Abstract

Domain adaptation is an important topic for natural language processing. There has been extensive research on the topic and various methods have been explored, including training data selection, model combination, semi-supervised learning. In this study, we propose to use a goodness measure, namely, description length gain (DLG), for domain adaptation for Chinese word segmentation. We demonstrate that DLG can help domain adaptation in two ways: as additional features for supervised segmenters to improve system performance, and also as a similarity measure for selecting training data to better match a test set. We evaluated our systems on the Chinese Penn Treebank version 7.0, which has 1.2 million words from five different genres, and the Chinese Word Segmentation Bakeoff-3 data.

Keywords: word segmentation, domain adaptation, description length gain

1. Introduction

Domain adaptation is an important topic for natural language processing (NLP) because, without it, the performance of NLP systems often degrades significantly when training and test data come from different domains. There has been extensive research on this topic, and the methods include training data selection (e.g., (Moore and Lewis, 2010; Plank and van Noord, 2011)), model combination (e.g., (McClosky et al., 2010)), feature copying (Daume 2007), semi-supervised learning (e.g., (McClosky et al., 2006)), and many more.

In this study, we focus on domain adaptation for Chinese word segmentation (CWS). There have been many studies on CWS in the past few decades (e.g., (Xue, 2003; Low et al., 2005; Zhao et al., 2006; Song et al., 2009; Sun et al., 2011)), and it is well-known that the main challenge to CWS is to identify the out-of-vocabulary (OOV) words (Huang et al., 2007). We propose to use an existing goodness measure, namely description length gain (DLG) (Kit and Wilks, 1999), for domain adaptation for CWS. Intuitively, the DLG score of a character sequence indicates the reduction of description length of a corpus when the sequence is treated as a unit and all the occurrences of the sequence in the corpus are replaced by the index of this unit. Previously, DLG has been used to identify OOV words (Kit and Liu, 2005) and as global features for supervised learning (Zhao and Kit, 2008). We use DLG in two ways. First, like in (Zhao and Kit, 2008), we add DLG-based features to two supervised systems (one is a CRF segmenter and the other is a joint model for CWS and POS tagging), and show that the performance of the systems improve significantly, especially when the training and test genres are very different. Second, we define a DLG-based similarity measure for selecting a subset of training data for a given test corpus, and show that this automatic selection method outperforms random selection in several scenarios that we have tested.

The paper is organized as follows. In Section 2, we provide the definition of DLG. In Section 3, we describe our baseline segmenters and how we extend them by incorporating DLG-based features. In Section 4, we define a DLG-based similarity measure for training data selection. Section 5 describes the corpora used in our evaluation, and Section 6 reports experimental results of the systems described in Section 3 and 4. The final section draws the conclusion.

2. Description Length Gain

Description length gain (DLG), based on the theory of minimum description length (Rissanen, 1989), is a goodness measure proposed by Kit and Wilks (1999) as an unsupervised learning approach to lexical acquisition. More formally, for the CWS task, let X=x_1 x_2 ... x_n be a corpus, which is a string of characters. Kit (1998) defines the description length of *X*, *DL(X),* as the Shannon-Fano code length for the corpus, which is shown in Eq (1). Here, n is the size of the corpus, *V* is the set of distinct tokens (i.e., the vocabulary) in *X*, and *c(x)* is the count of a token x in *X*.

$$DL(X) = n\ H(X) = -\sum_{x \in V} c(x) \log \frac{c(x)}{n} \quad (1)$$

Kit and Wilks (1999) uses Eq (2) to calculate the description length gain (DLG) from identify a substring s in X as a segment or chunk. Here $X[r \rightarrow s]$ denotes a new corpus after the operation of replacing all s with a new character r, which can be seen as an index for the newly identified chunk s; the operator $\oplus$ denotes a concatenation operation of two strings with a delimiter inserted in between.

$$DLG(s \in X) = DL(X) - DL(X[r \rightarrow s] \oplus s) \quad (2)$$

Note that it is trivial to recover the original corpus X from the new corpus, $X[r \rightarrow s] \oplus s$.

3. DLG for improving CWS

Intuitively, the DLG of a string s indicates the reduction of description length of a corpus X when the characters in s are treated as a unit and all the occurrences of s in X are replaced by the index of the unit. Therefore, the more frequent s is in X and the longer s is, the higher DLG(s) is. However, strings with high DLG scores include not only words, but also common word collocations (e.g., verb + aspect marker, noun compounds). Consequently, DLG scores alone, e.g., using the sum of DLG scores as the objective function for unsupervised segmentation as in (Kit and Wilks, 1999), are not sufficient for achieving high CWS performance. Instead of relying on DLG scores only, we choose to add DLG-based features to supervised segmenters. The DLG-based features are explained in Section 3.2.

In order to evaluate the effect of adding DLG-based features, we build two baseline systems: (1) SEG is a supervised, CRF based word segmenter, and (2) SEG+POS is a joint model for CWS and POS tagging. We then add DLG-based features to both systems, resulting in two new systems: SEG + DLG and SEG+POS+DLG. The four systems are described below.

3.1 SEG and SEG+POS: two baseline systems

In our first baseline system, SEG, we follow the general practice of treating CWS as a character tagging task (Xue, 2003), and build a Conditional Random Fields (CRF) (Lafferty et al., 2001) tagger. We adopt a six-tag set used in (Zhao and Kit, 2008). The six tags present a single-character word (S), the first three positions (B1, B2, B3), the middle position (M), and the last position (E) of a multiple-character word, respectively. For instance, if "*c1 c2 c3 c4 c5*" is a word, the corresponding tags will be "c1/B1 c2/B2 c3/B3 c4/M c5/E". The features used by SEG are shown in Table 1.

When the training data includes POS tag labels, previous studies (Kruengkrai et al., 2009; Zhang et al., 2010; Sun, 2011) have shown that a joint model for CWS and POS tagging improves performance of both tasks. Our second baseline system, SEG+POS, is a joint model. Building it on top of SEG is straightforward; we only need to replace the six-tag set in the SEG system with a set consisting of x-y tags, where x is one of the six tags for word segmentation and y is a POS tag. For instance, given a sentence "c1c2/NN c3/VV c4c5/Adv", the corresponding tag sequence will be "c1/B1-NN c2/E-NN c3/S-VV c4/B1-Adv c5/E-Adv". We use the same feature set as in SEG (see Table 1).

Type	Feature	Function
Unigram	C_{-1}, C_0, C_1	The previous, current, and next character
Bigram	$C_{-1}C_0$, C_0C_1	The bigrams that include the current character
Jump	$C_{-1}C_1$	The previous and next characters

Table 1: Features used in SEG and SEG+POS.

3.2 SEG+DLG and SEG+POS+DLG: adding DLG-based features

Zhao and Kit (2008) has demonstrated that integrating unsupervised segmentation criteria into supervised learning for CWS is an effective way to improve the performance of OOV words' recognition, and thus improve the overall performance of CWS. They tested four segmentation criteria: frequency of substring after reduction, DLG, Accessor Variety (Feng et al., 2004), and Boundary Entropy. In this study, we focus on DLG, as it can be easily extended for training data selection.

Let *X1* and *X2* be the training corpus and the test corpus, respectively. Let *X* be the concatenation of *X1* and *X2*. The SEG+DLG system is created in three steps:

1. We calculate DLG(s) for every character string s in X according to Eq (2). For the sake of efficiency, we only look at strings with no more than five characters.

2. We form the feature vector for each character in the training and test data. In addition to the features in the SEG system, we add five DLG-based features, $feat_j$ (j=1, 2, …, 5), for the current character *c*, which present the most likely tag for *c* according to DLG scores if c belongs to a word of length j in the current sentence.

3. We train and test the CRF model with the feature vectors formed in Step 2.

Note that the word boundary information in the training or test data is NOT needed for calculating DLG in Step 1 or the DLG-based features in Step 2. The first and the third step are straightforward. Let us explain Step 2 with an example.

Suppose a sentence has five characters "*c1 c2 c3 c4 c5*", and we want to form the feature vector for c3. Table 2 show the *LogDLG* scores for all the character sequences in the sentence that contain c3, where *LogDLG(s)* is simply the largest integer no greater than log(DLG(s)), as shown in Eq (3).[1] We use *LogDLG(s),* instead of *DLG(s),* to eliminate the effect of minor differences in DLG scores

[1] When a character sequence appears only once in the corpus or the length of the sequence is one, its DLG score is negative. In order to ensure the operation of the log function, we normalize the raw DLG scores with a big number Z. The value of Z is not important because the final score is used only for comparison.

and reduce the number of features. The last column of Table 2 shows the tag of c3 if the string in the first column is treated as a word.

$$LogDLG(s \in X) = \mathit{floor}\ (\log DLG(s)) \qquad (3)$$

String s	LogDLG(s)	c3's tag in s
c3	1	S
c2 c3	2	E
c3 c4	1	B1
c1 c2 c3	4	E
c2 c3 c4	1	B2
c3 c4 c5	2	B1
c1 c2 c3 c4	4	B3
c2 c3 c4 c5	3	B2
c1 c2 c3 c4 c5	3	B3

Table 2: DLG scores for strings that contain c3. The scores are calculated from the whole corpus, not just from the current sentence.

Based on the scores in Table 2, the DLG-based features for c3 are calculated, as shown in Table 3. The first column is the length of a word that c3 could belong to, which ranges from one to five. For each length j, we look at all the strings in Table 2 with that length and choose the string with the highest *LogDLG(s)* score. The string is listed in the second column, its score is in the third column, and c3's tag in the string is in the fourth column.

Finally, the last column shows the DLG-based feature of the form F_j= tag-score, where j, tag, and score come from the 1st, 3rd and 4th columns of the table. The feature is a binary feature, and it stores the most likely tag for the current character based on comparison of *LogDLG* scores for the strings with the same length and the highest *LogDLG* score. We use several features, one for each length, because, as mentioned before, DLG scores tend to be high for long, frequent strings and thus it is good to have separate features for different string lengths. The training step will learn how useful those features are. Zhao et al. (2008) described a similar way for filtering the features by comparing their scores.

Leng	String with the highest score (s)	Score of s	c3's tag in s	Feature added
1	c3	1	S	F1=S-1
2	c2 c3	2	E	F2=E-2
3	c1 c2 c3	4	E	F3=E-4
4	c1 c2 c3 c4	4	B3	F4=B3-4
5	c1 c2 c3 c4 c5	3	B3	F5=B3-3

Table 3: DLG-based features for character c3 in sentence "*c1 c2 c3 c4 c5*", based on the *LogDLG* scores in Table 2.

Adding the same kind of DLG-based features to SEG+POS yields our fourth segmenter, SEG + POS + DLG, a system that benefits from both the DLG features and the joint model.

4. DLG for Training Data Selection

A common approach to domain adaptation is training data selection; that is, choosing a subset of the training data that is more suitable for a given test set. This strategy not only reduces demand on computational resources, but could also potentially improve system performance, as demonstrated by several previous studies (Lu et al., 2007; Plank and van Noord, 2011; Moore and Lewis, 2010; Axelrod et al., 2011) for tasks such as parsing, language modeling, and machine translation.

In this section, we propose to use a DLG-based similarity score to select training data for CWS. We evaluate our automatic selection system on the Chinese Penn Treebank v7.0, which has data from five genres. Because some genres have only a few dozen files, we select sentences, not files, from the training data.

4.1 Similarity Measurement

A key issue for training data selection is what kind of measurement we can use to estimate the similarity between a text segment (e.g., a sentence) in the training data and the test data. Because the text segment in our study is a sentence, probability distributions such as word or topic distributions would not work well because a sentence is too short to collect reliable distributions.

Intuitively, we want to use a measure that checks the overlap between substrings in a training sentence and the test corpus. In addition to the percentage of overlap, we also want the measure to take into consideration the length and the frequency of a substring. DLG is such a measure because the more frequent and the longer a substring is, the higher its DLG score is. Based on this, we define a similarity measure, *Sim(Sent, X),* between a training sentence *Sent* and a test corpus *X* as the average of DLG scores of substrings in Sent, as shown in Eq (4). The higher the score is, the more similar the sentence is to the test corpus based on overlapping substrings and their frequencies.

$$Sim(sent, X) = \frac{1}{n} \sum_{s \in Substr(Sent)} DLG(s) \qquad (4)$$

Here, *Substr(Sent)* is the set of substrings in *Sent*, and *n* is the size of the set. As always, for the sake of efficiency, we only consider substrings with at most five characters. Note that *DLG(s)* is calculated with respect to the whole test corpus *X*, and the calculation of *Sim(Sent, X)* does not use word boundary information in the training or the test data.

4.2 Selecting training sentences

Given the similarity measure, selecting training sentences is straightforward. First, we enumerate all the substrings (with length 5 or lower) in the training data; second, we calculate DLG scores for these substrings with respect to the test corpus; third, we calculate *Sim(Sent, X)* for each training sentence Sent using Eq (4); fourth, we sort the training sentences in descending order of *Sim(Sent, X)* and select a certain percentage (e.g., 5% or 10%) of the training corpus from the top. For evaluation, the sentences selected in this procedure will be compared with randomly selected sentences with respect to CWS performance.

5. Evaluation Corpora

For evaluation, we use several datasets: the Chinese Penn Treebank and the Bakeoff-3 data. Both have been used in many previous studies.

5.1 The Chinese Penn Treebank

The Chinese Penn Treebank (CTB) (Xia et al., 2000) has been developed since late 1990s. The first release of the corpus, CTB1, consisted of newswire articles only, but in later versions, text from more genres and sources were added. The latest release is version 7.0, which includes 1.2 million words in five genres, as shown in Table 4.

Genre	# of words [2]	# of files
Newswire (nw)	260k	811
Magazine (mz)	258k	130
Broadcast news (bn)	287k	1,207
Broadcast Conversation (bc)	184k	86
Weblog (web)	210k	214
Total	1,199k	2,448

Table 4: Statistics of the CTB 7.0

Data Set	File IDs	# of sents	# of words
Training	1-270 400-931 1001-1151	18,089	493,939
Development	301-325	350	6,821
Test	271-300	348	8,008

Table 5: Statistics of the CTB 5.0

In order to compare our systems with previous systems, we use CTB 5.0 as our second data set. We follow the data split used in several previous studies, as in Table 5.

5.2 The Bakeoff-3 data

The Bakeoff-3 data set was first collected for the third international Chinese language processing evaluation (Levow, 2006), and has been used as a benchmark since then. It consists of four tracks, two of which (CITYU, CKIP) use traditional characters, the others (MSRA, CTB) use simplified characters. The statistics of those corpora are in Table 6.

Corpus	Encoding	Training	Test
CITYU	BIG5	1.6M	220K
CKIP	BIG5	5.5M	91K
MSRA	GB	1.3M	100K
CTB	GB	509K	155K

Table 6: The sizes of training and test data (in numbers of words) in the Bakeoff-3 data.

5.3 The Chinese Gigaword corpus

For the two segmenters using DLG scores, the scores can be calculated from the union of the training and test data. In that case, the segmenters are built for a given test set. To test how well the segmenters work without using a particular test set, we use the Chinese Gigawords (GW)[3] as an additional resource; that is, DLG scores can be calculated from the union of the training data and GW, and therefore are not specific to a test set. In order to be able to compare our systems with (Wang et al., 2011), we use the same subset of the corpus as in their experiments and the size is about 200 million words. As always, DLG scores are calculated from raw data.

6. Experimental results

In this section, we report the performance of the systems built in Section 3 and 4. For all the experiments on CTB7.0, for the purpose of comparison, we break the data in each genre into three portions: 20% as testing data, the first 150K words of the 80% as training data, and the next 25K words of the 80% as the development data.[4]

6.1 The SEG system on the CTB 7.0

In order to understand how domain variations affect system performance, we trained and tested our baseline system, SEG, on the five genres in the CTB 7.0. A matrix of CWS results are shown in Table 7, in which the row and column show the genres of the training and test data, respectively. For instance, cell (bc, mz) means SEG is trained on the training portion of "bc", and tested on the test portion of "mz". The two numbers in a cell are the overall F-score for all words and the F-score for OOV words, respectively. In each column, the highest score (according to overall F-score) is in boldface; the second highest one is marked with an underline, and the lowest one is marked with a wavy underline.

There are several observations from Table 7. First, not surprisingly, the highest accuracy is achieved when the training and test data are from the same genres. Second, some genres (e.g., mz and web) are harder than others (e.g., bc, bn, and nw), as shown in the last row. Third, from the matrix we can define the closeness of genres according to CWS performance; that is, given test data in genre G1 and training data in genre G2 and in G3, we can

[2] The numbers of words in this table are based on our own calculation from the CTB7 final release. They are slightly different from the numbers on the LDC release page at http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2010T07

[3] Released by the LDC with catalog number LDC2005T14.

[4] Because the sizes of "bc" and "web" genres are smaller, the training and development data for them are slightly smaller than 150K and 25K words, respectively.

say that G2 is closer to G1 than G3 to G1 if the overall F-score in cell (G2, G1) is higher than the one in cell (G3, G1). For example, the first column indicates that "web" is the closest to "bc", and "nw" is the most "distant" from "bc", when compared to other non-bc genres.

	bc	bn	mz	nw	web
bc	**93.92/71.08**	87.99/51.13	81.75/45.11	85.74/46.64	85.54/46.91
bn	87.94/56.78	**94.10/69.65**	82.76/48.58	87.62/54.60	85.14/46.35
mz	86.81/57.76	87.74/52.81	**91.58/66.68**	88.21/51.01	85.67/48.34
nw	83.05/52.02	89.69/61.62	85.84/55.04	**94.57/69.38**	83.77/46.07
web	92.13/69.47	89.45/57.12	84.19/50.63	86.79/50.95	**91.50/62.94**
All	96.09/73.02	96.10/72.75	93.17/66.60	96.18/71.67	93.47/61.99

Table 7: Performance of the SEG system, trained and tested on various genres. For all the experiments (except the "All" row), the training data size is 150k words, and the test data is 20% of the data in the test genre. The training data for the "All" row is the union of the training portions of all five genres.

6.2 SEG vs. SEG+DLG on the CTB 7.0

To test the effect of DLG-based features, we ran two sets of experiments, using bc and mz as the test genres. For each test genre, we use three training data sets: one from the test genre, one from the genre that is the closest to the test genre, and the third from the farthest genre, based on the overall F-scores in Table 7. Each training data set has 150K words. The results of SEG+DLG are shown in Table 8. To facilitate comparison, we copied the SEG performance from Table 7. Table 8 shows that in all six pairs of experiments, SEG+DLG outperforms SEG in both overall F-score and OOV F-score. Furthermore, the farther away the training and the test genres are, the greater the improvement of SEG+DLG over SEG is.

Training genre	SEG	SEG+DLG
Test genre is "bc"		
bc	93.92/71.08	94.96/77.02
web	92.13/69.47	93.38/76.23
nw	83.05/52.02	86.67/63.10
Test genre is "mz"		
mz	91.58/66.68	92.58/71.63
mw	85.84/55.04	88.27/62.91
bc	81.75/45.11	86.16/57.50

Table 8: Performance of SEG v.s. SEG+DLG. The test data is 20% of data in the "bc" or "mz" genre. The training data is 150K words from three genres. The two numbers in each cell are overall F-score and OOV F-score.

6.3 SEG and SEG+DLG on the Bakeoff-3

Table 9 shows the results of the two segmenters on Bakeoff-3 data. The scores in the first row are from the best system for each track. For the first two tracks, we collect DLG scores from three different sources, as indicated by the content in the parentheses. We did not do it for the last two tracks because the data in those tracks are traditional characters, whereas the Gigaword corpus uses simplified characters. For each dataset, the highest overall F-score is in bold, and the highest R_{oov} (recall of the OOV words) is in bold and with underline.

The table shows that for all the tracks our SEG+DLG system achieves the highest Roov, and its overall F-score is higher than the best systems participated in Bakeoff-3, and very close to the results in (Zhao and Kit, 2008). Furthermore, SEG+DLG performs well even when the DLG scores are estimated from "Train + GW", without using the test data.

	MSRA	CTB	CityU	AS
Best systems in Bakeoff3 (2006)	96.30/61.20	93.30/70.70	97.20/78.70	95.80/70.20
(Zhao and Kit, 2008)	96.60/66.20	94.31/76.08	**97.47**/80.05	**95.86/**69.35
Our systems				
SEG	95.98/66.43	93.04/70.96	96.82/77.97	95.38/65.99
SEG+DLG (Train+Test)	96.43/**69.40**	94.28/**76.36**	97.32/**80.29**	95.84/**69.37**
SEG+DLG (Train+GW)	96.61/69.43	94.27/75.74	----	
SEG+DLG (Train+Test+GW)	**96.68**/69.28	**94.35**/76.31	----	

Table 9: Performance on Bakeoff-3 data. The corpora used to calculate DLG scores are specified in the parentheses: "Train" is the training data, "Test" is the test data, and "GW" stands for the Chinese Gigaword Corpus.

6.4 Four segmenters on the CTB 5.0

Now we compare the performance of all four segmenters with the state-of-the-art segmenters that use both word segmentation and POS tagging annotation in the training data. In this case, two measures are used: the overall F-score on word segmentation (F_{seg}), and a joint overall F-score (F_{joint}) where a match is a word in the system output that agrees with the gold standard in both the word boundary and the POS tag of the word. For this experiment, we use CTB 5.0, not CTB 7.0, because CTB 5.0 is the dataset used by the previous studies in Table 10.

The table shows the performance of our two systems with DLG-based features is in par with the top systems in the field. Compared to those top systems, our systems are arguably less complex and therefore easier to implement and extend.

System	F_{seg}	F_{joint}
Kruengkrai et al., 2009	97.87	93.67
Zhang and Clark, 2010	97.78	93.67
Sun, 2011	**98.17**	94.02
Wang, 2011*	98.11	94.18
Our systems		
SEG	97.28	--
SEG+DLG (train+test)	97.91	--
SEG+DLG (train+test+GW)*	98.13	--
SEG+POS	97.84	94.02
SEG+POS+DLG (train + test)	97.98	94.01
SEG+POS+DLG (train+test+GW)*	98.12	**94.20**

Table 10: System performance on the CTB5. The asterisk means that the system uses the Gigaword corpus as an additional resource.

6.5 Training data selection on CTB 7.0

For training data selection, given a training genre G1 and the test data from genre G2, we want to select a subset of sentences from the training portion of G1 for that test data. We compare the data selected by our system with random selection in three scenarios:

(1) Training and test data are from a same genre. The results for using “bc” are in Table 11.

(2) Training and test data are from different genres, for which we choose “nw” as G1 and “bc” as G2 because “nw” is the most distance genre from “bc” according to Table 7. The results for training on “nw” and testing on “bc” are in Table 12.

(3) Training data is made up from multiple genres that do not include the test genre. In one experiment, we chose “mz” as the test genre, because it is the most difficult genre according to Table 7. The training data is the union of the training portion of all other four genres. The results are in Table 13.

In Tables 11-13, the scores for random selection are the average scores of five random selections; if a subset of training data produces better results than the whole training data, the corresponding scores are in bold.

% of the training data	Selected (F-all/F-oov)	Random (F-all/F-oov)
1%	82.66/46.26	79.66/41.48
5%	88.15/57.75	87.41/55.98
10%	90.05/61.35	89.00/60.01
20%	91.53/65.39	90.74/65.29
40%	93.10/69.63	92.53/69.47
60%	94.02/72.33	93.80/72.23
80%	94.85/76.08	94.65/75.97
100%	94.96/77.02	94.96/77.02

Table 11: Segmentation results when training and test data are both from the “bc” genre. The numbers in the second and third columns are the f-scores for all the words and OOV words.

Several observations are worth noting. First, our selection method outperforms the random selection for all three scenarios, and the gap is larger when a smaller percentage of training data is selected. Second, the gap between “Selected” and “Random” is larger in Tables 12 and 13 than in Table 11, a desirable property since training data selection is more meaningful when the training and test data are from different genres.

% of the training data	Selected (F-all/F-oov)	Random (F-all/F-oov)
1%	78.11/47.61	71.34/40.60
5%	83.37/57.56	78.42/49.43
10%	84.48/58.78	81.37/54.23
20%	85.34/60.39	83.67/57.40
40%	86.26/61.22	84.72/59.09
60%	86.44/62.31	85.60/59.80
80%	**86.73/63.28**	86.24/61.51
100%	86.67/63.10	86.67/63.10

Table 12: Segmentation results when the training data is from “nw” and the test data is “bc”.

% of the training data	Selected (F-all/F-oov)	Random (F-all/F-oov)
1%	81.60/52.24	75.47/45.38
5%	85.02/57.10	83.15/54.56
10%	87.35/59.70	85.88/57.33
20%	88.65/63.60	87.02/62.31
40%	90.12/65.32	88.86/64.97
60%	91.02/69.70	90.55/68.54
80%	**91.95/72.02**	91.02/70.46
100%	91.34/71.18	91.34/71.18

Table 13: Segmentation results when the test data is from “mz” and the training data is from the other four genres.

7. Conclusion

We demonstrate that DLG can help domain adaptation for CWS in several ways: first, it can be incorporated into supervised segmenters as features and the resulting systems achieve the state of the art on several benchmark datasets. Second, it can be used for training data selection, to find a subset of training data that better match a test set.

For future work, we plan to explore other goodness measures (such as accessor variety) and determine whether they can be combined with DLG for CWS domain adaptation. We will also look at other tasks such as POS tagging and parsing, and investigate whether there are goodness measures like DLG which could help domain adaptation for those tasks.

8. Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This research is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D11PC20153. The

U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

9. References

Amittai Axelrod, Xiaodong He, Jianfeng Gao. 2011. Domain Adapatation via Pseudo In-Domain Data Selection. In *Proceedings of EMNLP-2011*, pages 355-362.

David Chiang and Daniel M. Bikel. 2002. Recovering latent information in treebanks. In *Proceedings of COLING-2002*.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of ACL-2007*, Prague, Czech Republic.

Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004. Accessor variety criteria for Chinese word extraction. *Computational Linguistics*, 30(1):75–93.

Jenny Rose Finkel and Christopher D. Manning. 2009.Hierarchical bayesian domain adaptation. In *Proceedings of HLT-NAACL-2009*, pages 602–610, Boulder, Colorado, June.

Aidan Finn and Nicholas Kushmerick, 2006. Learning to classify documents according to genre: Special Topic Section on Computational Analysis of Style. *Journal of the American Society for Information Science and Technology*, v.57 n.11, pages 1506-1518.

Chang-Ning Huang and Hai Zhao. 2007, Chinese Word Segmentation: A Decade Review. *Journal of Chinese Information Processing*. 2007(,5): pages. 8-19.

Chunyu Kit and Yorick Wilks. 1999. Unsupervised learning of word boundary with description length gain. In M. Osborne and E. T. K. Sang, editors, In *Proceedings of CoNLL-1999*, pages 1–6.

Canasai Kruengkrai, Kiyotaka Uchimoto, Junichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and POS tagging. In *Proceedings of ACL-IJCNLP 2009*, pages 513–521.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML'01*, pages 282–289.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In Proceedings of *SIGHAN-5*, pages 108–117, Sydney, Australia, July 22-23.

Jin Kiat Low, Hwee Tou Ng, and Wenyuan Guo. 2005. A maximum entropy approach to Chinese word segmentation. In *Proceedings of SIGHAN-4*, pages 161–164.

Yajuan Lu, Jin Huang and Qun Liu. 2007. Improving Statistical Machine Translation Performance by Training Data Selection and Optimization. In *Proceedings of EMNLP-2007*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of COLING-ACL-2006*, pages 337–344, Sydney, Australia, July.

Robert Moore and William Lewis. 2010. Intelligent Selection of Language Model Training Data. In *Proceedings of ACL-2010*, pages 220-224.

Barbara Plank and Gertjan van Noord, 2011. Effective Measures of Domain Similarity for Parsing. In *Proceedings of ACL-2011*, pages 1566-1576, Portland, Oregon, USA.

J Rissanen. 1989. *Stochastic Complexity in Statistical Inquiry*. World Scientific, N.J.

Yan Song, Chunyu Kit, Ruifeng Xu, and Hai Zhao. 2009. How Unsupervised Learning Affects Character Tagging based Chinese Word Segmentation: A Quantitative Investigation, in *Proceedings of International Conference on Machine Learning and Cybernetics (ICMLC) 2009*.

E. Stamatatos, N. Fakotakis, and G. Kokkinakis, 2000. Text genre detection using common word frequencies. In *Proceedings of the 18th conference on Computational linguistics (COLING)*, Saarbrücken, Germany.

Weiwei Sun. 2011. A Stacked Sub-Word Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging. In *Proceedings of ACL-2011*, pages 1385–1394.

Weiwei Sun and Jia Xu. 2011. Enhancing Chinese Word Segmentation Using Unlabeled Data. In *Proceedings of EMNLP-2011*, pages 970-979. Edinburgh, Scotland, UK.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL*, pages 252-259.

Yiou Wang, Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, Kentaro Torisawa. 2011. Improving Chinese Word Segmentation and POS Tagging with Semi-supervised Methods Using Large Auto-Analyzed Data. In *Proceedings of IJCNLP-2011*, pages 309-317, Chiang Mai, Thailand.

Fei Xia, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Fu-Dong Chiou, Shizhe Huang, Tony Kroch, and Mitch Marcus. 2000. Developing Guidelines and Ensuring Consistency for Chinese Text Annotation. In *Proceedings of LREC-2000*.

Nianwen Xue. 2003. Chinese Word Segmentation as Character Tagging. *International Journal of Computational Linguistics and Chinese Language Processing*, 8(1):29-48.

Yue Zhang and Stephen Clark. 2010. A fast decoder for joint word segmentation and POS-tagging using a single discriminative model. In *Proceedings of EMNLP-2010*, pages 843–852, Cambridge, MA.

Hai Zhao, Chang-Ning Huang and Mu Li. 2006. An improved Chinese word segmentation system with conditional random field. In Proceeding of *SIGHAN-5*, pages 162–165, July 22-23.

Hai Zhao and Chunyu Kit. 2008. Exploiting Unlabeled Text with Different Unsupervised Segmentation Criteria for Chinese Word Segmentation. In *Proceedings of CICLing-2008*, pages 17-23, Haifa, Israel.