Automatic Determination of a Domain Adaptation Method for Word Sense Disambiguation Using Decision Tree Learning

Kanako Komiya
Tokyo University of Agriculture and Technology
2-24-16 Naka-cho, Koganei
Tokyo, 184-8588 Japan
kkomiya@cc.tuat.ac.jp

Manabu Okumura
Tokyo Institute of Technology
4259 Nagatsuta Modori-ku
Yokohama 226-8503 Japan
oku@pi.titech.ac.jp

Abstract

Domain adaptation (DA), which involves adapting a classifier developed from source to target data, has been studied intensively in recent years. However, when DA for word sense disambiguation (WSD) was carried out, the optimal DA method varied according to the properties of the source and target data. This paper describes how the optimal method for DA was determined depending on these properties using decision tree learning, given a triple of the target word type of WSD, the source data, and the target data, and discusses what properties affected the determination of the best method when Japanese WSD was performed.

1 Introduction

Classifiers in standard supervised machine learning have been trained for data in domain A using manually annotated data in domain A, e.g., to train classifiers for newswires using newswires. However, classifiers for data in domain B have sometimes been necessary when there have been no or few manually annotated data, and there have only been manually annotated data in domain A, which have been related to domain B. Domain adaptation (DA) involves adapting the classifier that have been trained from data in domain A (source domain) to data in domain B (target domain). This has been studied intensively in recent years.

However, the optimal method of DA varied according to the properties of the data in the source domain (the source data) and the data in the target domain (the target data) when DA for word sense disambiguation (WSD) was carried out. (We will show it in Section 4.)

We define a case as a triple of the target word type of WSD, the source data, and the target data. This paper describes how the optimal method for DA was determined depending on these properties using decision tree learning given a case and discusses what properties affected the determination of the best method when Japanese WSD was performed.

This paper is organized as follows. Section 2 reviews related works on DA and Section 3 explains how a DA method is automatically determined. Section 4 describes the data we used. How to label the data and how to train the classifiers using these are explained in Section 5. We present the results in Section 6 and discuss them in Section 7. Finally, we conclude the paper in Section 8.

2 Related Work

The DA problem can be categorized into three types depending on the information for learning, i.e., supervised, semi-supervised, and unsupervised approaches. A classifier in a supervised approach is developed from a large amount of labeled source data and a small amount of labeled target data with the aim of classifying target data better than a classifier developed only from the target data. A classifier in a semi-supervised approach is developed from large amounts of labeled source data and unlabeled target data with the aim of classifying target data better than a classifier developed only from the source data. Finally, a classifier is developed from a large amount of labeled source data with the aim of classifying target data accurately in the unsupervised approach. We focused on the supervised DA of WSD in this paper.

Many researchers have investigated DA within or outside the area of natural language processing. Chan and Ng (2006) carried out the DA of WSD by estimating class priors using an EM algorithm. Chan and Ng (2007) also conducted the DA of WSD by estimating class priors using the EM algorithm, but this was supervised DA using

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1107–1115,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

active learning.

In addition, Daumé III (2007) worked on the supervised DA. He augmented an input space and made triple length features that were general, source-specific, and target-specific. This was easy to implement, could be used with various DA methods, and could easily be extended to multi-domain adaptation problems. Daumé III et al. (2010) extended the work in (Daumé III, 2007) to semi-supervised DA. It inherited the advantages of the supervised version and outperformed it by using unlabeled target data.

Agirre and de Lacalle (2008) worked on the semi-supervised DA of WSD. They applied singular value decomposition (SVD) to a matrix of unlabeled target data and a large amount of unlabeled source data, and trained a classifier with them. Agirre and de Lacalle (2009) worked on the supervised DA using almost the same method, but they used a small amount of labeled source data instead of the large amount of unlabeled source data.

Jiang and Zhai (2007) demonstrated that performance increased as examples were weighted when DA was applied. This method could be used with various other supervised or semi-supervised DA methods. In addition, they tried to identify and remove source data that misled DA, but they concluded that it was only effective if examples were not weighted.

Zhong et al. (2009) proposed an adaptive kernel approach that mapped the marginal distribution of source and target data into a common kernel space. They also conducted sample selection to make the conditional probabilities between the two domains closer.

Raina et al. (2007) proposed self-taught learning that utilized sparse coding to construct higher level features from the unlabeled data collected from the Web. This method was based on unsupervised learning.

Tur (2009) proposed a co-adaptation algorithm where both co-training and DA techniques were used to improve the performance of the model.

The research by Blitzer et al. (2006) involved work on semi-supervised DA, where they calculated the weight of words around the pivot features (words that frequently appeared both in source and target data and behaved similarly in both) to model some words in one domain that behaved similarly in another. They applied SVD to the matrix of the weights, generated a new feature space, and used the new features with the original features.

The closest work to ours is that by McClosky et al. (2010) who focused on the problem where the best model for each document is not obvious when parsing a document collection of heterogeneous domains. They studied it as a new task of *multiple source parser adaptation*. They proposed a method of parsing a sentence that first predicts accuracies for various parsing models using a regression model, and then uses the parsing model with the highest predicted accuracy. The main difference is that their work was about parsing but ours discussed here is about Japanese WSD. They also assumed that they had labeled corpora in heterogeneous domains but we have not. We determined the best DA method using the decision tree learning given a triple of the target word type of WSD, the source data, and the target data and found what features affected the determination of the best method.

Harimoto et al. (2010) measured the distance between domains to conduct DA using a suitable corpus in parsing. In addition, van Asch and Daelemans (2010) reported that performance in DA could be predicted depending on the similarity between source and target data using automatically annotated corpus in parsing. They focused on how corpora were selected for use as source data according to the distance between domains, but here we focus on how to select a method of DA depending on properties such as the distance between domains.

3 Automatic determination of DA method

We expected the average accuracy of WSD, when DA methods that were determined automatically were used for all cases, to be higher than when the original methods were used collectively. Hence, we would be able to determine the best DA method automatically using decision tree learning. A decision tree would indicate what features affect the determination of the optimal method of DA.

3.1 DA methods for WSD

Two methods were used as the DA methods for WSD in this study.

- *Target Only*: Train a classifier with a small amount of target data that are randomly selected and manually labeled but without source data.

- *Random Sampling*: Train a classifier with source data and a small amount of target data that are randomly selected and manually labeled.

Ten word tokens of the target data were randomly selected and manually labeled in all the experiments.

Libsvm (Chang and Lin, 2001), which supports multi-class classification, was used as the classifier for WSD. A linear kernel was used according to the results obtained from preliminary experiments. Seventeen features were introduced to train the classifier.

- Morphological features
  - Bag-of-words（4 features）
  - Part-of-speech (POS)（4 features）
  - Finer subcategory of POS（4 features）
- Syntactic feature（1 feature）
  - If the POS of a target word is a noun, the verb which the target word modifies
  - If the POS of a target word is a verb, the case element of "ヲ" (wo, objective) for the verb
- Semantic features
  - Semantic classification code（4 features）

Morphological features and semantic features were extracted from the surrounding words (two words to the right and left) of the target word. POS and finer subcategory of POS can be obtained using a morphological analyzer. We used ChaSen [1] as a morphological analyzer, the Bunruigoihyo thesaurus (National Institute for Japanese Language and Linguistics, 1964) for semantic classification codes, and CaboCha [2] as a syntactic parser. Five-fold cross validation was used in the experiments.

3.2 Labels of Decision Tree

One of the following labels was given to every case depending on the most accurate method and as we shall explain later, two labels (TO and RS) or three labels (TO, RS, and SA) were used for classification. Note that the decision tree determines which DA method should be used, *Random Sampling* or *Target Only*, given the properties of the source and target data for each case.

- TO: The cases in which *Target Only* had higher accuracy than *Random Sampling*.
- RS: The cases in which *Random Sampling* had higher accuracy than *Target Only*.
- SA: The cases in which *Random Sampling* and *Target Only* had the same accuracy.

3.3 Features of Decision Tree

We think the optimal method for DA varies depending on the distribution of the source data and the target data, the distance between them, and so on. The following 40 features (consisting of 24 types) in total were used for decision tree learning.

1. Simulation accuracy of *theOther*: The accuracy of WSD when a classifier was trained with the source data and tested with ten labeled word tokens of the target data.
2. Simulation accuracy of *Target Only*: The accuracy of WSD when a classifier was trained with ten labeled word tokens of the target data and tested using a leave-one-out cross-validation method.
3. Ratio of two simulation accuracies: (1) / (2).
4. Number of source data: The number of word tokens in the whole source data.
5. Number of target data: The number of word tokens in the whole target data.
6. Number of source data / target data: (4) / (5).
7. The number of word senses that appeared in the whole source data set.
8. The number of word senses that appeared in ten word tokens of the target data.
9. The number of word senses of the WSD target words in the dictionary.
10. The number of word tokens of the most frequent sense (MFS) of the whole source data.
11. The number of word tokens of MFS in the ten labeled word tokens of the target data.

[1] http://sourceforge.net/projects/masayu-a/

[2] http://sourceforge.net/projects/cabocha/

12. Whether the MFS of the whole source data and the ten labeled word tokens of the target data were the same or not.

13. Percentage of MFS in source data: (10) / (4).

14. Percentage of MFS in ten word tokens of target data: (11) / the number of word tokens in the labeled target data (=10).

15. Percentage of MFS in ten word tokens of target data in source data: The number of source data word tokens with MFS in ten labeled word tokens of the target data / (4).

16. Percentage of MFS of source data in ten word tokens of target data: The number of word tokens with MFS in the source data in ten labeled word tokens of the target data / the ten word tokens.

17. The JS divergence between the distribution of word sense IDs of 4/5 of the whole source data and the distribution of word sense IDs of ten labeled word tokens of target data. Abbreviated as "JSD (word sense)".

18. The JS divergence between the feature distributions for WSD of the whole source and the whole target data (17 kinds, cf. Section 3.1) Abbreviated as "JSD (*)". *is a feature name in Section 3.1.

19. The summation of 17 kinds of JS divergences (18). Abbreviated as "JSD (Feature_plus)".

20. The JS divergence between the distribution of the whole source data and the whole target data feature units, when a unit is the sequence of 17 kinds of WSD features. Abbreviated as "JSD (Feature_all)".

21. The number of word senses that did not appear in ten labeled word tokens of the target data but did in the whole source data.

22. The number of common word senses between the whole source data and ten labeled word tokens of the target data.

23. Percentage of common word senses between whole source data and ten word tokens of target data in ten word tokens: the number of word tokens whose word senses appeared in both the whole source data and ten labeled word tokens of the target data in the ten word tokens / the ten word tokens.

24. Percentage of common word senses between whole source data and ten word tokens of target data in source data: the number of word tokens whose word senses appeared in both the whole source data and ten labeled word tokens of the target data in the source data/ (4).

The C4.5 of Quinlan (1993) was used as the algorithm for decision tree learning and a binary tree was generated. The experiments were conducted with five-fold cross validation. The threshold values for pruning were optimized with preliminary experiments using 1/4 of the training data set as a development data set. Here, the entropy of a node was tuned as the threshold value in 0.1 increments. The value of a smaller tree was used when more than one threshold value gave the same accuracy.

4 Data

Three data were used for the experiments: (1) the sub-corpus of white papers in the Balanced Corpus of Contemporary Japanese (BCCWJ) (Maekawa, 2008), (2) the sub-corpus of documents from a Q&A site on the WWW of BCCWJ, and (3) Real World Computing (RWC) text databases (newspaper articles) (Hashida et al., 1998). DAs were conducted in six directions according to various source and target data. Word senses were annotated in these corpora according to a Japanese dictionary, i.e., the Iwanami Kokugo Jiten (Nishio et al., 1994). It has three levels for sense IDs, and we used the fine-level sense in the experiments. Multi-sense words that appeared equal or more than 50 times in both source and target data were selected as the target words in the experiment. There were 24 word types for white papers ⇔ Q&A site, 22 for white papers ⇔ newspaper articles, and 26 for Q&A site ⇔ newspaper articles. Twenty-eight word types and 144 cases were used in the experiments in total. Table 1 lists the minimum, maximum, and average number of word tokens in each case. Table 2 shows the list of target words.

Table 3 summarizes the results from the DA experiments when the source data and the target data were swapped [3]. These results were the av-

[3] Multi-sense words that appeared less than 50 times in source or target data were included as the target words in the

Table 1: Minimum, maximum, and average number of word tokens in each case

Genre	Min.	Max.	Ave.
BCCWJ white papers	58	7,610	2074.50
BCCWJ Q&A site	82	13,976	2300.43
RWC newspaper	50	374	164.46

erage accuracies of all the target words of WSD. *Self*, which is standard supervised learning with the target data, assuming that fully annotated data were obtained and could be used for learning and *theOther*, which is standard supervised learning only with the source data, were tested as references. We found that the optimal method of DA varied depending on the corpora that were used as the source and target data.

5 Labeling of data and learning of decision tree

We tried to generate decision trees in various ways and compared their accuracies to find the most effective way of generating the best decision tree.

5.1 Labeling of data

TO, RS, or SA was given to every case depending on the most accurate method. SA was particularly given to cases in which *Random Sampling* and *Target Only* had the same accuracy.

The difference between accuracies and the definition of SA was treated in two ways.

- *Equal*: The SA label was assigned when the WSD accuracies of *Target Only* and *Random Sampling* were totally equal.
- *Chi-square*: The SA label was assigned when the WSD accuracies of *Target Only* and *Random Sampling* were not significantly different according to a chi-square test. The level of significance in the test was 0.05.

Table 4 indicates the number of cases and the total word types given TO or RS labels according to *Equal* and *Chi-square*.

5.2 Treatment of SA in decision tree learning

The third label, SA, was assigned to cases with no difference between the accuracies of *Random Sampling* and *Target Only* and was treated in two ways in the experiments.

Table 2: The list of target words

Number of senses	Target words (in Japanese)	Sense example in English
2	場合 自分	case self
3	事業 情報 地方 社会 思う 子供	project information area society suppose child
4	分かる 考える	understand think
5	含む 使う 技術	contain use technique
6	関係 時間 一般 現在 作る	connection time general present make
7	今	now
8	前	before
10	持つ	have
11	進む	advance
12	見る	see
14	入る	enter
16	言う	say
21	出す	serve
22	手 出る	hand leave

- *Ternary classification with SA*: Perform ternary classification of TO, RS, and SA in training and the testing.
- *Binary classification without SA*: Remove cases with SA labels from training data set and perform binary classification of TO and RS. All the cases are used for the testing.

Figure 1 shows the frame format for five-fold cross validation of decision tree learning when *Binary classification without SA* was used. There was a total of 144 cases, and only data with TA or RS labels were used as a training data set for decision tree learning. There were 129 cases when *Equal* was used and 69 cases when *Chi-square* was used (dark gray parts). A classifier was developed from 4/5 of the training data set (white

experiment of this figure.

Table 3: Results from DA experiments when source and target data were swapped

DA method	Accuracy	
Source data	Q&A site	white papers
Target data	white papers	Q&A site
theOther	79.65%	83.35%
Random sampling	85.40%	83.86%
Target Only	88.20%	77.74%
self	95.97%	91.65%

Table 4: Number of cases and total word types given TO or RS labels according to *Equal* and *Chi-square*

Source data	Target data	*Equal*	*Chi*
white papers	Q&A site	21	13
white papers	newspaper	18	9
Q&A site	newspaper	25	12
Q&A site	white papers	25	12
newspaper	white papers	20	11
newspaper	Q&A site	20	12
Total cases		129	69
Total word types		27	20

parts) and tested using 1/5 of the whole cases (light gray part) in one execution of five-fold cross validation. (Three-quarter of the training data (3/5 of the whole data set) and 1/4 of the training data (1/5 of the whole data set) was used for the training data and the test data of the parameter tuning respectively.)

We calculated the accuracies of WSD using the DA method for the labels (*Target Only* for TO and *Random Sampling* for RS). We used *Random Sampling* for cases with SA labels when *Ternary classification with SA* was used. When *Binary classification without SA* was used, we had no correct answers for the cases of SA in the test phase. However, either label could be given to them because they were cases in which *Random Sampling* and *Target Only* had the same accuracy. Therefore, we assigned TO or RS to them depending on the decision tree that was generated.

5.3 Classification

As Table 1 indicates, some cases had many word tokens and some had few. For example, a case had 58 word tokens when the source data were from the RWC newspaper articles, the target data were from the BCCWJ white papers, and the target word type was "言う". (This case had 58 word tokens because the word appeared 58 times). Therefore, the average accuracy of WSD could be improved if cases that had more word tokens could be predicted more precisely. We also classified cases with weighted word tokens, as well as just classified cases, in two ways.

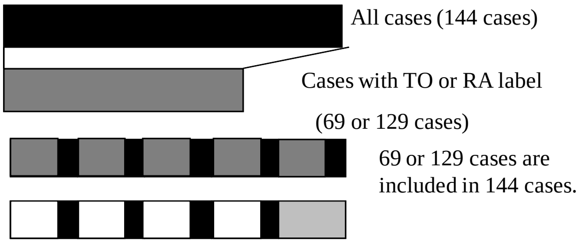

Figure 1: Five-fold cross validation of decision tree learning when *Binary classification without SA* was used

- *Case classification*: Perform decision tree learning on the assumption that every case has the same weight.
- *Classification with weighting of word tokens*: Perform decision tree learning with weighting on the assumption that every case has the weight of the number of word tokens in the case.

The weights of the cases were used to calculate entropy.

6 Results

Table 5 lists the average accuracies of WSD when the original methods were used collectively. The average accuracies were calculated from 232,116 word tokens in 144 cases. (They were micro-averaged over the word tokens.) Table 5 indicates that *Target Only* outperformed *Random Sampling* and its accuracy was 81.23%. Here, *Selected Source Only* is a DA method where train a classifier with only selected source data that are similar to the target data. We used 0.8 of cosine distance as a threshold value. We did not include this method as a DA method that is automatically determined but showed as reference.

Table 6 summarizes the average accuracies of WSD when the DA methods that were determined were used for every case. There were nine ways of determining the DA methods: eight automatic and one manual. The eight automatic approaches

Table 5: Average accuracy of WSD when methods were used collectively

DA method	Accuracy of WSD
Target Only	81.23 %
Random Sampling	80.28 %
Selected Source Only	82.27 %

Table 6: Average accuracy of WSD when methods that were determined automatically were used

Way to determine a method	Accuracy of WSD
Equal_3_case	82.36%
Equal_3_token	82.44%
Chi_3_case	83.49%
Chi_3_token	83.42%
Equal_2_case	**83.50**%
Equal_2_token	81.88%
Chi_2_case	82.55%
Chi_2_token	82.92%
manually	85.25%

were all combinations of the two choices in Sections 5.1, 5.2, and 5.3. Abbreviations of the ways in Table 6 are in the format of a_b_c where a is the choice for Section 5.1, b is that for Section 5.2 and c is that for Section 5.3. The "3" and "2" represent *Ternary classification with SA* and *Binary classification without SA*, and "case" and "token" represent *Case classification* and *Classification with weighting of word tokens*, respectively. When the manual approach was used, the DA method with the highest accuracy was chosen manually for every case. Its average accuracy was in the upper bound for our proposed method.

Table 6 shows that the automatic way with the highest average accuracy was Equal_2_case. The accuracy was 83.50%, and it significantly outperformed *Target Only* and *Selected Source Only* in Table 5 (81.23% and 82.27%) . This means that the average accuracy of WSD when DA methods that were determined automatically were used was higher than when the original methods were used collectively.

7 Discussion

7.1 Comparison of ways of determining DA methods

We compare the results for ways of determining DA methods in this section. First, usually *Chi-square* was better for labeling of data, but *Equal* was better when *Binary classification without SA* and *Case classification* were used simultaneously. Second, usually *Ternary classification with SA* was better for treating SA in decision tree learning, but *Binary classification without SA* was better when *Equal* and *Case classification* were used simultaneously. Finally, we could not find any patterns of the results for *Case classification* or *Classification with weighting of word tokens*.

The approach with the highest accuracy in the eight automatic ways was where SA labels were applied when the accuracies of WSD for the two methods were totally equal, binary classification was performed without SA, and cases were classified without weighted word tokens (Equal_2_case). Significant differences were found in the three comparisons above when they were evaluated with a Chi-square test when the other conditions were the same.

The accuracy of decision tree learning was 60.42% when the Equal_2_case was used. This value was not very high, which may be due to the optimizing threshold value of pruning with a development data set. We think the reason low classification accuracy with decision trees did not critically influence the average accuracy of WSD was that most errors were for cases where *Random Sampling* and *Target Only* had almost the same accuracy.

7.2 Discussion on learned decision tree

We present the decision tree with the highest accuracy in an appendix in five executions of five-fold cross validation in the decision tree learning of the Equal_2_case, whose average WSD accuracy was the highest in the eight automatic ways of learning, and we discuss the features and their values that contributed to the generation of the tree.

First, TO was assigned when the "ratio of two simulation accuracies $>= 0.40$" was false in the root node of the decision tree. Therefore, TO was assigned to cases whose ratio of "Simulation accuracy of *theOther* / Simulation accuracy of *Target Only*" was lower than 0.40. That is, TO was assigned to the cases when the accuracy of WSD when a classifier was trained with ten labeled word tokens of the target data and tested using a leave-one-out cross-validation method was higher than the accuracy of WSD when a classifier was trained with the source data and tested using ten labeled word tokens of the target data. In other words, this indicates that the simulation using ten manually labeled word tokens of the target data was an important clue in predicting the optimal DA method.

Next, TO was selected when JSD (bag-of-words of one word to the left of the target word) was equal to or more than 0.61 in the node with level one. This indicated that a classifier should only be trained with ten labeled word tokens of target data, without source data, when the distributions of the feature of bag-of-words of one word to the left of the WSD target word differed between the source and target data.

Moreover, TO was selected when JSD (semantic classification code of one word to the right of the target word) was equal to or more than 1.00 in the node with level two. This indicated that a classifier should only be trained with ten labeled word tokens of target data, without source data, when the distributions of the semantic classification code features of one word to the right of the WSD target word differed between the source and target data.

The decision tree in the appendix is small and simple, but the second best tree (there is a difference in only one case) consists of 13 questions. According to the tree, TO tends to be assigned when JS divergences such as JSD (word sense), JSD (Feature_plus), and JSD (Syntactic feature) are large, and RS tends to be assigned when they are small. Large JS divergence indicates that the distributions of a feature are different, and small JS divergence indicates that the distributions of a feature are close. This indicates that when the distributions of the important feature for WSD are different and the source data are not sufficiently close to the target data, the source data should not be used.

8 Conclusion

We described how the optimal method of DA could be determined depending on the properties of the source and target data using decision tree learning and found what properties affected the determination of the best method when Japanese WSD was performed. We defined a case as a triple of the target word type of WSD, the source data, and the target data, all of which were classified into two labels (TO and RS) or three labels (TO, RS, and SA). Here, the case with TO should only be trained with a small amount of target data, the case with RS should be trained with source data and a small amount of target data, and SA represents a case with no difference between the accuracies for the two methods. The average accuracy of WSD when the DA methods that were determined automatically were used was significantly higher than when the original methods were used collectively. We automatically generated a decision tree in eight ways, the most accurate of which was with SA label when the WSD accuracies of the two methods were totally equal, performed binary classification without SA, and classified cases without weighted word tokens. The top node in the tree that was generated indicated that simulation using ten manually labeled word tokens of the target data was an important clue enabling the optimal DA method to be predicted.

Acknowledgments

We would like to thank the reviewers for very constructive and detailed comments. This research is supported by Grants-in-Aid for Scientific Research, Priority Area ”Japanese Corpus”.

A Generated decision tree

Upper edge represents true and lower edge represents false.

```
          ┌── TO: 3 cases
     ┌── JSD（bag-of-words of one word to the left
     |    |     of the target word）>= 0.61
     |    |     ┌── TO: 3 cases
     |    └── JSD（semantic classification code
     |          |  of one word to the right of
     |          |  the target word) >= 1.00
     |          └── TO: 28 cases RS: 62 cases
Ratio of two simulation accuracies >= 0.40
     └── TO: 7 cases
```

References

Eneko Agirre and Oier Lopez de Lacalle. 2008. On robustness and domain adaptation using svd for word sense disambiguation. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 17–24.

Eneko Agirre and Oier Lopez de Lacalle. 2009. Supervised domain adaption for wsd. In *Proceedings of the 12th Conference of the European Chapter of the Association of Computational Linguistics*, pages 42–50.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural coppespondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128.

Yee Seng Chan and Hwee Tou Ng. 2006. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 89–96.

Yee Seng Chan and Hwee Tou Ng. 2007. Domain adaptation with active learning for word sense disambiguation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 49–56.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Hal Daumé III, Abhishek Kumar, and Avishek Saha. 2010. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing, ACL 2010*, pages 23–59.

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

Keiko Harimoto, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Kobunkaiseki no bunyatekiou ni okeru seido teika youin no bunseki oyobi bunyakan kyori no sokutei syuhou, in japanese. In *Proceedings of NLP2010*, pages 27–30.

Koichi Hashida, Hitoshi Isahara, Takenobu Tokunaga, Minako Hashimoto, Shiho Ogino, and Wakako Kashino. 1998. The rwc text databases. In *Proceedings of the First International Conference on Language Resource and Evaluation*, pages 457–461.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271.

Kikuo Maekawa. 2008. Balanced corpus of contemporary written japanese. In *Proceedings of the 6th Workshop on Asian Language Resources (ALR)*, pages 101–102.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 28–36.

National Institute for Japanese Language and Linguistics. 1964. *Bunruigoihyo*. Shuuei Shuppan, In Japanese.

Minoru Nishio, Etsutaro Iwabuchi, and Shizuo Mizutani. 1994. *Iwanami Kokugo Jiten Dai Go Han*. Iwanami Publisher, In Japanese.

John Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. 2007. Self-taught learning: Transfer learning from unlabeled data. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 759–766.

Gokhan Tur. 2009. Co-adaptation: Adaptive co-training for semi-supervised learning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009.*, pages 3721–3724.

Vincent van Asch and Walter Daelemans. 2010. Using domain similarity for performance estimation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing, ACL 2010*, pages 31–36.

Erheng Zhong, Wei Fan, Jing Peng, Kun Zhang, Jiangtao Ren, Deepak Turaga, and Olivier Verscheure. 2009. Cross domain distribution adaptation via kernel mapping. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1027–1036.