Analysis of Selective Strategies to Build a Dependency-Analyzed Corpus

Kiyonori Ohtake
National Institute of Information and Communications Technology (NICT),
ATR Spoken Language Communication Research Labs.
2-2-2 Hikaridai "Keihanna Science City" Kyoto 619-0288 Japan
kiyonori.ohtake [at] nict.go.jp

Abstract

This paper discusses sampling strategies for building a dependency-analyzed corpus and analyzes them with different kinds of corpora. We used the Kyoto Text Corpus, a dependency-analyzed corpus of newspaper articles, and prepared the IPAL corpus, a dependency-analyzed corpus of example sentences in dictionaries, as a new and different kind of corpus. The experimental results revealed that the length of the test set controlled the accuracy and that the longest-first strategy was good for an expanding corpus, but this was not the case when constructing a corpus from scratch.

1 Introduction

Dependency-structure analysis plays a very important role in natural language processing (NLP). Thus, so far, much research has been done on this subject, with many analyzers being developed such as rule-based analyzers and corpus-based analyzers that use machine-learning techniques. However, the maximum accuracy achieved by state-of-the art analyzers is almost 90% for newspaper articles; it seems very difficult to exceed this figure of 90%. To improve our analyzers, we have to write more rules for rule-based analyzers or prepare more corpora for corpus-based analyzers.

If we take a machine-learning approach, it is important to consider what features are used. However, there are several machine-learning techniques, such as support vector machines (SVMs) with a kernel function, that have strong generalization ability and are very robust for choosing the right features. If we use such machine-learning techniques, we will be free from choosing a feature set because it will be possible to use all possible features with little or no decline in performance. Actually, Sasano tried to expand the feature set for a Japanese dependency analyzer using SVMs in (Sasano, 2004), with a small improvement in accuracy.

To write rules for a rule-based analyzer, and to produce an analyzer using machine-learning techniques, it is crucial to construct a dependency-analyzed corpus. Such a corpus is very useful not only for constructing a dependency analyzer but also for other natural language processing applications. However, building this kind of resource is very expensive and labor-intensive because it is difficult to annotate a large amount of dependency-analyzed corpus in short time.

At present, one promising approach to mitigating the annotation bottleneck problem is to use selective sampling, a variant of active learning (Cohn et al., 1994; Fujii et al., 1998; Hwa, 2004). In general, selective sampling is an interactive learning method in which the machine takes the initiative in selecting unlabeled data for the human to annotate. Under this framework, the system has access to a large pool of unlabeled data, and it has to predict how much it can learn from each candidate in the pool if that candidate is labeled.

Most of the experiments that had been carried out in the previous works for selective sampling used an annotated corpus in a limited domain. The most typical corpus is WSJ of Penn Treebank. The reason why the domain was so limited is very simple; corpus annotation is very expensive. However, we want to know the effects of selective sampling for corpora in various domains because a dependency analyzer constructed from a corpus does not always analyze a text in limited domain.

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 635–642,
Sydney, July 2006.

On the other hand, there is no clear guideline nor development strategy for constructing a dependency-analyzed corpus to produce a highly accurate dependency analyzer. Thus in this paper, we discuss fundamental sampling strategies for a dependency-analyzed corpus for corpus-based dependency analyzers with several types of corpora. This paper unveils the essential characteristics of basic sampling strategies for a dependency-analyzed corpus.

2 Dependency-Analyzed Corpora

We use two dependency-analyzed corpora. One is the Kyoto Text Corpus, which consists of newspaper articles, and the other one is the IPAL corpus, which contains sentences extracted from the "example of use" section of the enties in several dictionaries for computers. The IPAL corpus was recently annotated for this study as a different kind of corpus.

2.1 Kyoto Text Corpus

In this study we use Kyoto Text Corpus version 3.0. The corpus consists of newspaper articles from Mainichi Newspapers from January 1st to January 17th, 1995 (almost 20,000 sentences) and all editorials of the year 1995 (almost 20,000 sentences). All of the articles were analyzed by morphological analyzer JUMAN and dependency analyzer KNP[1]. After that, the analyzed results were manually corrected. Kyoto Text Corpus version 4.0 is now available, holding on additional 5,000 annotated sentences in the corpus to version 3.0 for case relations, anaphoric relations, omission information and co-reference information[2].

The original POS system used in the Kyoto Text Corpus is JUMAN's POS system. We converted the POS system used in the Kyoto Text Corpus into ChaSen's POS system because we used ChaSen, a Japanese morphological analyzer, and CaboCha[3] (Kudo and Matsumoto, 2002), a dependency analyzer incorporating SVMs, as a state-of-the art corpus-based Japanese dependency structure analyzer that prefers ChaSen's POS system to that of JUMAN. In addition, we modified some bunsetu segmentations because there were several inconsistencies in bunsetu segmentation.

Table 1 shows the details of the Kyoto Text Corpus.

	Kyoto Text Corpus	
	(General)	(Editorial)
# of sentences	19,669	18,714
# of bunsetu	192,154	171,461
# of morphemes	542,334	480,005
vocabulary size	29,542	17,730
bunsetu / sentence	9.769	9.162

Table 1: Kyoto Text Corpus

2.2 IPAL corpus

IPAL (IPA, Information-technology Promotion Agency, Lexicon of the Japanese language for computers) dictionaries consist of three dictionaries, the IPAL noun dictionary, the IPAL verb dictionary and the IPAL adjective dictionary. Each of the dictionaries includes example sentences. We extracted 7,720 sentences from IPAL Noun, 5,244 sentences from IPAL Verb, and 2,366 sentences from IPAL Adjective. We analyzed them using CaboCha and manually corrected the errors. We named this dependency-analyzed corpus the IPAL corpus. Table 2 presents the details of the IPAL corpus. One characteristic of the IPAL corpus is that the average sentence length is very short; in other words, the sentences in the IPAL corpus are very simple.

# of sentences	15,330
# of bunsetu	67,170
# of morphemes	156,131
vocabulary size	11,895
bunsetu / sentence	4.382

Table 2: IPAL corpus

3 Experiments

We carried out several experiments to determine the basic characteristics of several selective strategies for a Japanese dependency-analyzed corpus. First, we briefly introduce Japanese dependency structure. Second, we carry out basic experiments with our dependency-analyzed corpora and analyze the errors. Finally, we conduct simulations to

[1] `http://www.kc.t.u-tokyo.ac.jp/nl-resource`

[2] `http://www.kc.t.u-tokyo.ac.jp/nl-resource/corpus.html`

[3] `http://chasen.org/˜taku/software/cabocha/`

ascertain the fundamental characteristics of these strategies.

3.1 Japanese dependency structure

The Japanese dependency structure is usually defined in terms of the relationship between phrasal units called *bunsetu* segments. Conventional methods of dependency analysis have assumed the following three syntactic constraints (Kurohashi and Nagao, 1994a):

1. All dependencies are directed from left to right.
2. Dependencies do not cross each other.
3. Each *bunsetu* segment, except the last one, depends on only one *bunsetu* segment.

Figure 1 shows examples of Japanese dependency structure.

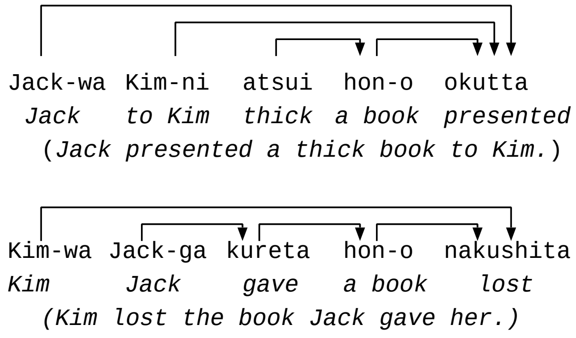
Jack-wa Kim-ni atsui hon-o okutta
Jack to Kim thick a book presented
(Jack presented a thick book to Kim.)
Kim-wa Jack-ga kureta hon-o nakushita
Kim Jack gave a book lost
(Kim lost the book Jack gave her.)

Figure 1: Examples of Japanese dependency structure

In this paper, we refer to the beginning of a dependency direction as a "modifier" and the end of that as a "head."

3.2 Analyzing errors

We performed a cross-validation test with our dependency-analyzed corpora by using the SVM-based dependency analyzer CaboCha. The feature set used for SVM in CaboCha followed the default settings of CaboCha.

First, we arbitrarily divided each corpus into two parts. General articles of the Kyoto Text Corpus were arbitrarily divided into KG0 and KG1, while editorials were also divided into ED0 and ED1. The IPAL corpus was arbitrarily divided into IPAL0 and IPAL1. Second, we carried out cross-validation tests on these divided corpora.

Table 3 shows the results of the cross-validation tests. We employed a polynomial kernel for the SVM of CaboCha, and tested with second- and third-degree polynomial kernels. The input data for each test were correct for morphological analysis and bunsetu segmentation, though in practical situations we have to expect some morphological analysis errors and bunsetu mis-segmentations.

In Table 3 "Learning" indicates the learning corpus, "Test" represents the test corpus, and "Degree" denotes the degree of the polynomial function. In addition, "Acc." indicates the accuracy of dependency-analyzed results and "S-acc." indicates the sentence accuracy that is the ratio of sentences that were analyzed without errors.

Learning	Test	Degree	Acc.(%)	S-acc.(%)
KG0	KG0	2	94.06	65.51
KG0	KG0	3	99.96	99.71
KG0	KG1	2	89.50	50.35
KG0	KG1	3	89.23	49.33
KG1	KG0	2	89.60	49.89
KG1	KG0	3	89.21	49.05
ED0	ED1	2	90.77	55.58
ED1	ED0	2	90.52	54.62
IPAL0	IPAL1	2	97.43	92.25
IPAL1	IPAL0	2	97.69	93.06
KG0	IPAL0	2	97.76	93.15
ED0	IPAL0	2	97.56	92.81

Table 3: Results of cross-validation tests

Table 3 also shows the biased evaluation (closed test; the test was the training set itself) results. In the cross-validation results of KG0 and KG1, the average accuracy of the second-degree kernel was 89.55 (154,455 / 172,485)% and the average sentence accuracy was 50.12 (9,858 / 19,669)%. In other words, there were 18,030 dependency errors in the cross validation test. We analyzed these errors.

Against the average length (9.769) of the corpus shown in Table 1, the average length of the sentences with errors in the cross-validation test is 12.53 (bunsetu / sentence). These results confirm that longer sentences tend to be analyzed incorrectly.

Next we analyzed modifier bunsetu that were mis-analyzed. Table 4 shows the top ten POS sequences that consisted of modifier mis-analyzed bunsetu.

We also analyzed the distance between modifier bunsetu and head bunsetu of the mis-analyzed dependencies. Table 5 shows top ten cases of the distance. In Table 5 "Err." indicates the distance between a modifier and a head bunsetu of mis-analyzed dependencies, "Correct" indicates

POS sequence	Frequency
noun, case marker	835
verb, comma	576
noun, topic marker	444
adverbial noun, comma	370
verb	336
number, numeral classifier, comma	318
noun, adnominal particle	304
adverb	304
verb, verbal auxiliary	281
verb, conjunctive particle, comma	265

Table 4: Modifier POS sequences of mis-analyzed dependencies and their frequencies in the cross-validation test (top 10)

the distance between a modifier and a correct (should modify) head bunsetu in each case of mis-analyzed dependencies, and "Freq." denotes their frequency.

Err.	Correct	Freq.	Err.	Correct	Freq.
1	2	3,117	2	4	478
2	1	1,362	3	2	436
3	1	919	4	1	434
1	3	863	4	2	379
2	3	482	1	4	329

Table 5: Frequencies of dependency distances at error and correct cases in the cross-validation test (top 10)

3.3 Selective sampling simulation

In this section, we discuss selective strategies through two simulations. One is expanding a dependency-analyzed corpus to construct a more accurate dependency analyzer, and the other is an initial situation just beginning to build a corpus.

3.3.1 Expanding situation

The situation is as follows. First, the corpus, Kyoto Text Corpus KG1, is given. Second, we expand the corpus using the editorials component of the Kyoto Text Corpus. Then we consider the following six strategies: (1) Longest first, (2) Maximizing vocabulary size first, (3) Maximizing unseen dependencies first, (4) Maximizing average distance of dependencies first, (5) Chronological order, and (6) Random.

We briefly introduce these six strategies as follows:

1. Longest first (Long)
   Since longer sentences tend to have complex structures and be analyzed incorrectly, we prepare the corpus in descending order of length. The length is measured by the number of bunsetu in a sentence.

2. Maximizing vocabulary size first (VSort)
   Unknown words cause unknown dependencies, thus we sort the corpus to maximize its vocabulary size.

3. Maximizing unseen dependencies first (UDep)
   This is similar to (2). However, we cannot know the true dependencies. The analyzed results by the dependency analyzer based on the current corpus are used to estimate the unseen dependencies. The accuracy of the estimated results was 90.25% and the sentence accuracy was 54.03%.

4. Maximizing average distance of dependencies first (ADist)
   It is difficult to analyze long-distance dependencies correctly. Thus, the average distance of dependencies is an approximation for the difficulty of analysis.

5. Chronological order (Chrono)
   Since there is a chronological order in newspaper articles, this strategy should feel quite natural.

6. Random (ED0)
   Chronological order seems natural, but newspaper articles also have cohesion. Thus, the vocabulary might be unbalanced when we consider the chronological order. We also try randomized order; actually, we used the corpus ED0 as the randomized corpus.

We sorted the editorial component of the Kyoto Text Corpus by each strategy mentioned above. After sorting, corpora were constructed by taking the top N sentences of each corpus sorted by each strategy. The size of each corpus was balanced with the number dependencies.

We constructed dependency analyzers based on each corpus, KG1 plus each prepared corpus, then tested them by using the following corpora: (a) K-mag, (b) IPAL0, and (c) KG0.

Corpus	# of sent.	# of bunsetu	vocabulary size	# of dependencies	# of bunsetu / sent.
Long	5,490	81,759	13,266	76,269	14.89
VSort	8,762	85,031	16,428	76,269	9.705
UDep	5,524	81,793	13,371	76,269	14.81
ADist	6,950	83,223	13,074	76,273	11.97
Chrono	9,342	85,609	13,278	76,267	9.164
ED0	9,357	85,628	13,561	76,271	9.151
K-mag	489	4,851	2,501	4,362	9.920
IPAL0	7,665	33,484	8,617	25,819	4.368
KG0	9,835	96,283	21,616	86,448	9.790
I-Long	5,523	91,972	20,068	86,449	16.65
I-VSort	8,437	94,881	28,867	86,444	11.25

Table 6: Detailed information of corpora

K-mag consists of articles from the Koizumi Cabinet's E-Mail Magazine. This magazine was first published on May 29th 1999 and is still released weekly. K-mag consists of articles of the magazine published from May 29th 1999 to July 19th 1999. In addition, since March 25th 2004 an English version of this E-Mail Magazine has been available. Thus, currently this E-mail Magazine is bilingual. The articles of this magazine were analyzed by the dependency analyzer CaboCha, and we manually corrected the errors.

K-mag includes a wide variety articles, and the average sentence length is longer than in newspapers. Basic information on K-mag is also provided in Table 6.

Learning corpus	Acc.(%)	S-acc.(%)
KG1	87.25	49.69
KG1+LONG	**87.67**	**51.53**
KG1+Vsort	87.25	50.10
KG1+UDep	87.57	51.12
KG1+ADist	**87.67**	50.72
KG1+Chrono	87.57	50.31
KG1+Rand	87.60	49.69

Table 7: Analyzed results of K-mag (which is different domain and has long average sentence length) with these learning corpora

3.3.2 Simulation for initial situation

The results revealed that the longest-first strategy seems the best way. Here, however, a question arises: "Does the longest-first strategy always provide good predictions?" We carried out an experiment to answer the question. The experimental results we presented above were simulations of an expanding corpus. On the other hand, it is also possible to consider an initial situation for building a dependency-analyzed corpus. In such a situation, which would be the best strategy to take?

Learning corpus	Acc. (%)	S-acc.(%)
KG1	97.68	93.02
KG1+LONG	**97.75**	**93.22**
KG1+Vsort	97.70	93.06
KG1+UDep	**97.75**	93.18
KG1+ADist	97.70	93.10
KG1+Chrono	97.71	93.06
KG1+Rand	97.69	93.06

Table 8: Analyzed results of IPAL0 (which is different domain and has short average sentence length) with these learning corpora

We carried out a simulation experiment in which there was no annotated corpus; instead we began to construct a new one. We used general articles from the Kyoto Text Corpus and tried the following three strategies: (a) Random (actually, KG0 was used), (b) Longest first (I-Long), and (c) maximizing vocabulary size first (I-VSort). Three corpora were prepared by these strategies. Table 6 also shows the corpora information. In this experiment, the corpora were balanced with respect to the number of dependencies. We used CaboCha with these corpora and tested them with K-mag, ED0, and IPAL0. Table 10 shows the results of the experiment.

	K-mag		ED0		IPAL0	
Corpus	Acc. (%)	S-acc. (%)	Acc. (%)	S-acc. (%)	Acc. (%)	s-acc(%)
Random (KG0)	87.87	49.69	**90.17**	53.64	**97.76**	**93.15**
I-Long	87.41	49.28	90.11	52.96	97.66	92.94
I-VSort	**87.92**	**50.31**	90.14	**53.86**	97.72	93.06

Table 10: Results of initial situation experiment

Learning corpus	Acc. (%)	S-acc. (%)
KG1	89.60	49.89
KG1+LONG	**89.99**	51.25
KG1+Vsort	89.97	51.31
KG1+UDep	89.98	**51.39**
KG1+ADist	89.98	51.01
KG1+Chrono	89.86	51.09
KG1+Rand	89.95	51.20

Table 9: Analyzed results of KG0 (which is the same domain and has almost the same average sentence length) with these learning corpora

4 Discussion

4.1 Error analysis

To analyze corpora, we employed the dependency analyzer CaboCha, an SVM-based system. In general, when one attempts to solve a classification problem with kernel functions, it is difficult to know the kernel function that best fits the problem. To date, second- and third-degree polynomial kernels have been empirically used in Japanese dependency analysis with SVMs.

In the biased evaluation (the test corpus was the learning corpus), the third-degree polynomial kernel produced very accurate results, almost 100%. On the other hand, in the open test, however, the third-degree polynomial kernel did not produce results as good as the second-degree one. We conclude from these results that the third-degree polynomial kernel suffered the over-fitting problem.

The second-degree polynomial kernel produced on accuracy of almost 94% in the biased evaluation, and this can be considered as the upper bound for the second degree polynomial kernel to analyze Japanese dependency structure. The accuracy was stable when we adjusted the soft-margin parameter of the SVM. However, there were several annotation errors in the corpus. Thus, if we correct such annotation errors, the accuracy would improve.

Table 4 indicates that case elements consisting of nouns and case markers were frequently mis-analyzed. From a grammatical point of view, a case element should depend on a verb. However, the number of relations between verbs and case elements is combinatorial explosion. Thus, we can conclude that the learning data were not sufficient for relations between verbs and case elements to analyze unseen relations.

On the other hand, in Table 4, verbs take many places in comparison to their distribution in the test set corpus. These verbs tend to form conjunctive structures and it is known that analyzing conjunctive structure is difficult (Kurohashi and Nagao, 1994b). Particularly when a verb is a head of an adverbial clause, it seems very difficult to detect a head bunsetu, which is modified by the verb.

From Table 5, we can conclude that the analyzed errors centered on short-distance relations; the analyzer especially tends to mis-analyze the correct distance of two as one. Typical cases of such mis-analysis are "N_1-*no* N_2-*no* N_3" and "[adnominal clause] N_1-*no* N_2." In some cases, it is also difficult for humans to analyze these patterns correctly.

4.2 Selective sampling simulation

The results revealed very small differences between strategies possibly due to insufficient corpus size. However, there was an overall tendency that the accuracy depended heavily whether how many long sentences with very long dependencies were included in the test set. Table 3 shows a simple example of this. In the cross-validation tests the accuracy of the general articles, the average length of which was 9.769 bunsetu / sentence, was almost 1% lower than that of the editorial articles, whose average length was 9.162 bunsetu / sentence. The reason why sentence length controlled the accuracy was that an error in the long-distance dependency may have caused other errors in order to satisfy the condition that dependencies do not cross each other in Japanese dependencies. Thus, many errors occurred in longer sentences. To improve the accuracy, it is vital to analyze very long-distance dependencies correctly.

From Tables 7, 8 and 9, the strategy of longest first appears good for the expanding situation even if the average length of the test set is very short like in IPAL0. However, in the initial situation, since there is no labeled data, the longest-first strategy is not a good method. Table 10 shows that the random strategy (KG0) and the strategy of maximizing vocabulary size first (I-VSort) were better than the longest-first strategy (I-Long). This is because the test sets comprised short sentences and we can imagine that there were dependencies included only in such short sentences. In other words, the longest-first strategy was heavily biased toward long sentences and the strategy could not cover the dependencies that were only included in short sentences.

On the other hand, the number of such dependencies that were only included in short sentences was quite small, and this number would soon be saturated when we built a dependency analyzed corpus. Thus, in the initial situation, the random strategy was better, whereas after we prepared a corpus to some extent, the longest-first strategy would be better because analyzing long sentences is difficult.

In the case of expansion, the longest-first strategy was good, though we have to consider the actual time required to annotate such long sentences because in general longer sentences tend to have more complex structures and introduce more opportunities for ambiguous parses. This means it is difficult for humans to annotate such long sentences.

5 Related works

To date, many works on selective sampling were conducted in the field related to natural language processing (Fujii et al., 1998; Hwa, 2004; Kamm and Meyer, 2002; Riccardi and Hakkani-Tür, 2005; Ngai and Yarowsky, 2000; Banko and Brill, 2001; Engelson and Dagan, 1996). The basic concepts are the same and it is important to predict the training utility value of each candidate with high accuracy. The work most closely related to this paper is Hwa's (Hwa, 2004), which proposed a sophisticated method for selective sampling for statistical parsing. However, the experiments carried out in that paper were done with just one corpus, WSJ Treebank. The study by Baldridge and Osborne (Baldridge and Osborne, 2004) is also very close to this paper. They used the Redwoods treebank environment (Oepen et al., 2002) and discussed the reduction in annotation cost by an active learning approach.

In this paper, we focused on the analysis of several fundamental sampling strategies for building a Japanese dependency-analyzed corpus. A complete estimating function of training utility value was not shown in this paper. However, we tested several strategies with different types of corpora, and these results can be used to design such a function for selective sampling.

6 Conclusion

This paper discussed several sampling strategies for Japanese dependency-analyzed corpora, testing them with the Kyoto Text Corpus and the IPAL corpus. The IPAL corpus was constructed especially for this study. In addition, although it was quite small, we prepared the K-mag corpus to test the strategies. The experimental results using these corpora revealed that the average length of a test set controlled the accuracy in case of expansion; thus the longest-first strategy outperformed other strategies. On the other hand, in the initial situation, the longest-first strategy was not suitable for any test set.

The current work points us in several future directions. First, we shall continue to build dependency-analyzed corpora. While newspaper articles may be sufficient for our purpose, other resources seem still inadequate. Second, while in this work we focused on analysis using several fundamental selective strategies for a dependency-analyzed corpus, it is necessary to provide a function to build a selective sampling framework to construct a dependency-analyzed corpus.

References

Jason Baldridge and Miles Osborne. 2004. Active learning and the total cost of annotation. In *Proceedings of EMNLP*.

Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, pages 26–33.

David A. Cohn, Les Atlas, and Richard E. Ladner.

1994. Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

Sean P. Engelson and Ido Dagan. 1996. Minimizing manual annotation cost in supervised training from corpora. In *Proceedings of the 34th Annual meeting of Association for Computational Linguistics*, pages 319–326.

Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. 1998. Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4):573–598.

Rebecca Hwa. 2004. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276.

Teresa M. Kamm and Gerard G. L. Meyer. 2002. Selective sampling of training data for speech recognition. In *Proceedings of Human Language Technology*.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69.

Sadao Kurohashi and Makoto Nagao. 1994a. KN Parser: Japanese dependency/case structure analyzer. In *Proceedings of Workshop on Sharable Natural Language Resources*, pages 48–55.

Sadao Kurohashi and Makoto Nagao. 1994b. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4):507–534.

Grace Ngai and David Yarowsky. 2000. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 117–125.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods treebank: Motivation and preliminary applicatoins. In *Proceedings of COLING 2002*, pages 1–5.

Giuseppe Riccardi and Dilek Hakkani-Tür. 2005. Active learning: Theory and applications to automatic speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(4):504–511.

Manabu Sasano. 2004. Linear-time dependency analysis for Japanese. In *Proceedings of Coling 2004*, pages 8–14.