IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 9–24
RECEIVED 01/11/10 ACCEPTED 21/12/10 FINAL 11/04/11

Large-vocabulary Lexical Choice with Rich Context Features

YUSUKE MATSUBARA[1] AND JUN'ICHI TSUJII[1,2]

[1] *The University of Tokyo, Japan*
[2] *Manchester Interdisciplinary Biocentre, UK*

ABSTRACT

This paper shows that syntactic information improves large-scale statistical lexical choice. Given a set of possible words to use, statistical lexical choice is the task to choose the most appropriate word to fill a gap in a sentence. The state-of-the-art methods of statistical lexical choice either rely only on window-based cooccurrence information or are focused on to specific word classes. We present a discriminative model of statistical lexical choice with local syntactic features and document-level features, in addition to window-based features. We evaluated our systems in the setting where we try to select the best substitution candidates for all occurrences of the content words, as well as in the smaller evaluation sets used in previous works. Experimental results on Penn Treebank and BLLIP corpus showed that the proposed method outperformed the state-of-the-art methods and that syntactic features improved the performance of prediction of lexical choice.

1 INTRODUCTION

Choosing a right word that conveys the meaning in mind is a difficult task, even for human. We encounter similar challenges in constructing systems in various application areas of natural language processing, including text information retrieval, machine translation and natural language generation. In most text collections including highly specialized ones, writers may use different expressions to denote the same concept. This issue, which is known as synonymy or lexical variation, has significant importance in improving coverage of IR systems. In machine translation, the selection of translation words is an important subtask, especially when the output of the system should fit to a specific style or controlled vocabulary [1]. Another straightforward application is the lexical chooser in natural language generation systems [2]. A lexical chooser chooses the most appropriate lexical entry from the lexicon, given a semantic representation of the text to be generated. Having an accurate lexical chooser in natural language generation is important especially when the size of the vocabulary is large.

The task of choosing a right word given a context, which we call *lexical choice*, strongly relates to paraphrasing. In both of the two areas, we try to model synonymy of the words that occur in corpora. The results of paraphrasing methods are usually evaluated by comparing each set of the output expressions with those taken from thesauri constructed by human. It means that they aim to build context-independent knowledge of synonymy. While paraphrasing takes more importance on the aspect of unsupervised mining of synonymous expressions from corpora, lexical choice focuses on how to filter true synonyms from a set of substitution candidates, given a specific context.

To obtain accurate models of synonymy, it is necessary to capture context information. Strictly speaking, it is almost impossible to have a perfectly-interchangeable set of expressions in natural language. [3] For example, people may accept that the verb *command* can be replaced with *tell* in a sentence of military-related context, such as *The general* commanded/told *the officers that ...*. The two words *command* and *tell* are obviously non-interchangeable in general context, but can be interchangeable if the context supports that substitution.

This motivates the statistical modeling of lexical choice, which aims, unlike traditional context-insensitive approaches to paraphrasing, to find which set of expression can be used interchangeably, given a specific context. A more formal description of lexical substitution will be given in Section 2.

Previous researches on lexical choice mostly focused on either the use of local context or limited coverage of vocabulary, as discussed in Section 3. In this paper, we propose to use wider context that spans over

[3] In this paper, we refer to the task defined in Section 2 as *lexical choice*, *lexical substitution* or *context-sensitive lexical parphrasing*, following different terminologies of previous researches.

syntactic phrases and a document, and evaluate with a more realistic size of vocabulary.

2 Task description

Following [3] and [4], we define the task of lexical choice as follows.

We are given a sentence, candidate substitutions, and a *lexical gap*, or a position which has to be filled with one of the candidate substitutions. Our task is to choose the most appropriate word from the candidate substitutions considering the context surrounding the lexical gap.

Let us illustrate the task description with an example of a computer-aided writing system composed of a paraphraser and lexical chooser. A user of the system inputs a sentence " *Workers dumped large burlap sacks of the imported substance*$_{[p_1]}$ *into a huge bin*." In the sentence, with a mark $[p_1]$ he tells the system that he has low confidence in his selection of the marked word and wants the system to suggest better alternatives. The system generates candidate expressions for the marked position, or *lexical gap* denoted by $[p_1]$. First, the paraphraser retrieves candidate substitutions, that is, expressions that are synonymous to the input expression *substance*, such as *substance*, *stuff* and *material*. Then, from the candidate substitutions, the lexical chooser chooses the substitution that is most probable to be placed in the lexical gap, typically by exploiting the information from available large scale corpora.

3 Previous work

In this section, we summarize existing approaches to context sensitive lexical paraphrasing.

A number of statistical measures has been proposed to measure how much a candidate word is relevant to the surrounding context. Among different statistical measures for the appropriateness of a candidate substitution to a lexical gap, t-score[3], the conditional probability given the local syntactic context [2], PMI score [4] achieved the best accuracy for the test set provided by Edmonds [3]. As the first method to tackle the problem of context sensitive lexical paraphrasing, Edmonds [3] proposed to use the sum of t-scores of the cooccurrences of a candidate word and context words. Inkpen [4] improved Edmonds' method by using Pointwise Mutual Information (PMI) criteria [5] to measure the associations between a candidate word and the context surrounding the target lexical gap the system tries to fill in. The appropriateness of a candidate word t to the context is measured by sum of the PMI scores defined as

$$\mathrm{PMI}(w_{-k}^{-1}, w_1^k, t) = \sum_{w \in (w_{-k}^{-1} \cup w_1^k)} \frac{C(w,t)N}{C(w)C(t)}, \tag{1}$$

where w_{-k}^{-1} denotes the k preceding words to the lexical gap, w_1^k denotes the k following words to the lexical gap, $C(\cdot,\cdot)$ denotes the number of the cooccurrences of two words, $C(\cdot)$ denotes the frequency of a word, N denotes the total number of word tokens. Gardiner and Dras [6] presented an approximation of Inkpen's method to accommodate the corpora with provided as N-gram instead of full text.

Bangalore and Ranbow [2] were first to model the task of lexical choice as a multi-class probabilistic classification. They proposed a method to fill a lexical gap by using local syntactic information provided by their syntactic chooser for natural language Generation. In similar task settings, Inkpen significantly improved her unsupervised PMI method with a boosting method with the features of PMI scores and word occurrences in the context windows [4]. Connor and Roth [7] proposed a bootstrapping approach composed of weak learners and a global classifier. A set of weak learners which correspond to context words in a fixed-length window and dependency relations surrounding each lexical gap. A binary classifier with global features, which learns from the aggregated prediction of the weak learners, tries to predict whether the substitution is appropriate or not, given tuple of a original word, a word to substitute with, a context sentence that contains the original word.

It has still not been clear how well lexical choice works for larger vocabulary size, for example, thousands of words or more. In the works that shares Edmonds' experimental setting [3, 4, 6], they only evaluated the performance for specific seven synonym sets composed of the words with less polysemy and similar frequency. The reason why they used such controlled evaluation set was that they wanted to focus on exploring other features than word frequency. Another approach is to treat lexical choice as a binary classification of word substitutions. The existing methods among that type either cover only a specific class of words, such as verbs, in order to exploit local syntactic structure [7], or were evaluated against a human-annotated, but small set of lexical substitutions [8].

We consider that it is promising to apply lexical choice to the term expansion in IR. While traditional term expansion methods without actual user feedback improve the recall of IR systems, it has been well known that they also tend to invite the risk of degradation of precision [9]. When we have an accurate and wide-coverage lexical choice module, we may achieve improvement of recall with less noisy expansion, by filtering out the expansion candidate which are inappropriate to the surrounding context.

Aiming to the application to term expansion explained above, we focus on constructing models of lexical choice with a larger vocabulary, which provides a more realistic estimation of the effectiveness of lexical choice for the application of IR.

4 PROPOSED METHOD

Our system is composed of two main steps: substitution candidate generator and substitution selector. Given an input document, in the first step, the candidate generator spots the word occurrences which can be substituted, and assigns possible substitution candidates to it, using simple dictionaries of substitutions. In the second step, the substitution selector assigns probabilities to the candidates based on a single maximum entropy model, which allows us to use rich features including document-based features in a single multiclass classifier setting for this problem. We describe each step in detail in the following sections.

4.1 *Substitution candidate generation*

We generate lexical gaps and substitution candidates by simply matching the given document with a *substitution dictionary*. A *substitution dictionary* provides a mapping from a pair of word and its part-of-speech (POS) to the set of its substitutions, which can replace the input word in *some* context. This assumption on dictionaries allows us to exploit an automatically extracted dictionary of substitutions or a domain-specific dictionary curated by human, although we used only a WordNet[10]-derived substitution dictionary in the experiment presented in this paper. Note that, for polysemous words, we simply use a merged set of the words taken from all of their synsets and let the system to choose correct ones considering the context.

4.2 *Substitution Selection*

Given the context C and the set of candidate substitutions S, employing the maximum entropy framework [11], we directly model the conditional probability of the candidate $w \in S$,

$$P(w|S;C) = \frac{\exp\left[\Lambda \cdot F(w,C,S)\right]}{\sum_{v \in C} \exp\left[\Lambda F(v,C,S)\right]}, \tag{2}$$

where $F = \{f_1, f_2, \cdots, f_K\}$ is a feature vector and Λ is the weight vector. We obtain the most probable substitution candidate $\hat{w}$ for given C and S as

$$\hat{w} = \operatorname*{argmax}_{w \in C} P(w|S;C) \tag{3}$$

Here, we choose a point-wise prediction approach, rather than sequential / global optimization approach. Point-wise approaches allows us to try a wide range of features without taking inhibiting computational cost. Instead, it doesn't allow us to directly model the consistency of a combination of predictions.

We will try to capture the consistency as features, as described in the following sections and listed in Table 1. We will discuss limitations of this approach in Section 6.

4.3 *Training*

In order to obtain labeled examples used to train this model, we make a naive assumption that a choice of a word in a given context is correct if and only if the same choice is found in the training corpus. We extract training examples from a POS-tagged corpus by generating lexical gaps and substitution candidates in the same manner as explained in Section 4.1, and assigning positive labels to the actually used words in the training corpus and negative labels to those which is not used, based on the assumption above.

4.4 *Feature extraction*

In order to capture different levels of context in a document, we use three different sets of features: *window-based features document-based features*. In addition that, we also use simple but effective *baseline features*. *window-based* and *baseline* features were essentially the same ones as used in [4].

A *window-based feature* is a function defined on a fixed-length window surrounding the target lexical gap. A *document-based feature* is a function defined on the bag-of-words of the document which contains the target lexical gap. By using document-based features, we can capture the word associations that occur in long-distance context, including the consistency of the wording across the document.

In syntax-based features, we use the information from the result of syntactic parse of the given sentence. The feature called *subtree* in Table 1, which is one of the syntax-based features, captures the association of a simplified syntactic position and a candidate word. This feature is meant to be beneficial, for example, when the candidate noun has strong tendency to be modified by some adjectives.

For example, from the candidate word *chairman* in the following sentence,

> Mr. Vinken is chairman$_{[p_1]}$ of Elsevier N.V., the Dutch publishing$_{[p_2]}$ group.

we can extract following features.

For the candidate *chairman* in the gap $[p_1]$:

Feature	Value
frequency	-10.9
unigram_*chairman*	true
pmi_*of,chairman*	0.102
pmiavg	0.0615
...	

For the candidate *publishing* in the gap $[p_2]$:

Feature	Value
frequency	-10.9
unigram_*publishing*	true
pmi_*Dutch,publishing*	0.102
pmiavg	0.0615
subtree_*publishing*,DT/NNP/*VBG/NN	true
...	

All the features used in the experiment are listed in Table 1.

[4] The value of frequency feature can be real-valued, when we perform normalization.

Table 1. List of features

Feature template	Type	Definition	Value type
`frequency`(w)	baseline	frequency of w	integer [4]
`unigram`_v(w)	baseline	true iff w is v	binary
`pmi`_v(w)	window	$\mathrm{PMI}(v, w)$ for p preceding / q following words	real
`pmiavg`(w,S)	window	$\dfrac{\sum_{v \in window} \mathrm{PMI}(v, w)}{window\ size}$	real
`cache`_w(w,S)	document	true iff w is seen somewhere else in S except for the original position	binary
`cache-l`_w(w,S)	document	true iff the lemma of w is seen somewhere else in S except for the original position	binary
`cache-o`_w(w,S;C)	document	true iff the lemma of one of C is seen somewhere else in S	binary
`subtree`_v, t(w,S;C)	syntax	true iff the sequence t matches the sequence of the tags of the sibling node of w in S and v	binary
`unigram+pos`_v, T(w,S)	syntax	true iff the pair of surface and POS (w,S) equals to (v,T)	binary

5 EXPERIMENTS

In this section, we compare our all-words maximum entropy model and new document-based features with the Inkpen’s supervised method described in Section 3.

5.1 *Experimental settings*

Table 2. Tasks and corpora

		Source corpus	#word tokens
Sample words task	Training	BLLIP 1988 + Penn Treebank	16893445
	Testing	BLLIP 1987	22926540
All words task	Training	Penn Treebank	10778880
	Testing	(4-fold cross validation)	-

Table 3. Part-of-speech mappings

mapping A	mapping B	target POS
NN,NNS,NNP,NNPS	NN	noun
JJ,JJR,JJS	JJ	adjective
VB,VBD, VBG,VBN,VBP,VBZ	VB	verb
others	others	*ignored*

Table 4. Edmonds' seven evaluation sets for lexical choice

Part-of-speech	substitution candidates[5]
adjective	*difficult, tough, hard*
noun	*error, mistake, oversight*
noun	*job, task, duty*
noun	*responsibility, commitment, obligation, burden*
noun	*material, stuff, substance*
verb	*give, provide, offer*
verb	*resolve, settle*

We evaluated our methods with two tasks. The first one, which we call *sample-words* task, is the mostly same experimental setting as [4]. The second one, which we call *all-words* task, is a task in which we try to substitute all of the content words. Table 2 shows statistics of each task. Note that the number of testing samples is very different since we do not limit the substitution candidates to specific set.

In training of both of the two task, we generated substitution candidates using a substitution dictionary that maps a POS-tagged word to the words connected by at least one synonymy relation in WordNet 3.0 [10], which has 117,659 words and 206,941 senses. The parts-of-speech are converted using the mapping A shown in Table 3.

We trained our models with the extracted samples from all sections of the Penn Treebank corpus 3.0 [12] and the sections W8_001 to W8_019 of BLLIP 1987-89 WSJ Corpus. We filtered only the parts-of-speech starting with NN (nouns), JJ (adjectives), VB (verbs) and RB (adverbs) as targets of substitution.

Testing differs for each of the two tasks. In the first setting, which we call *sample-words* task, we compared our method with previous methods of Edmonds' and Inkpen's, by evaluating the prediction accuracy for

[5] Originally Edmonds referred to it as *near-synonyms* [3].

the same corpus, Wall Street Journal of 1987, used by their evaluation. Specifically, we generated lexical gaps and substitution candidates using Edmonds' seven near-synonym sets shown in Table 4 for evaluation, which is shared by [3], [4] and [6].

In the second setting, which we call *all-words* task, we tried to substitute all the content words in the given text. The intention behind this setting is that we want to evaluate for a wider-coverage of substitution candidates, aiming to applying lexical substitution to term expansion in IR.

We evaluated our models with the extracted samples from the sections W7_001 to W7_127 of BLLIP 1987-89 WSJ Corpus. PMI scores are estimated on the sections from W8_001 to W8_108 and from W9_001 to W9_41 of BLLIP WSJ Corpus, with the 5 words window and normalization with a sigmoid function. Note that the corpus used to estimate PMI was a relatively small corpus consists of domain specific texts.

For preprosessing the queries to substitution dictionaries and for the `cache-l` feature explained in Table 1, we lemmatized the target words by using the rewriting rules and exception lists provided in the WordNet implementation. For the parameter of `pmi` features, we used $p = 3$ and $q = 2$.

Since Inkpen did not explicitly mention about how she obtained and used part-of-speech information to identify target gaps to fill, we could not make faithful reproduction of her experimental settings. Gardiner [6] also reported similar difficulty in reproducing the settings. As a result, the sample-words task is slightly different from Inkpen's task. This difference can be seen in the difference between the numbers of the test samples. We assume that this difference comes from the different mappings from fine-grained parts-of-speech in the corpus to WordNet's coarse-grained ones. Table 3 shows two criteria we used to map from the tag set of Penn Treebank to three coarse parts-of-speech, namely, noun, adjective and verb, which are used in WordNet.

5.2 *Evaluation method*

Since the cost of human judgements for the size of corpora given in Table 2 is prohibitive, we evaluate the results with the exact match to the original word in a gap. This evaluation also serves as a mean to compare our method with previous researches, since this was the one used by the state-of-the-art method of Inkpen's [4]. As Inkpen mentioned, the exact match gives substantial underestimation of the true accuracy, since some of the substitution candidates other than the original one can be acceptable in the given context.

5.3 *Experimental results*

Table 5.3 shows the experimental results of Sample-words task. Table 5.3 shows the experimental results of All-words task. Every difference between a pair of accuracy values listed in the tables is shown to be statistically significant using McNemar's test with $p < 0.05$.

Table 5.3 shows the performance comparison of the proposed method (Proposed-A, Proposed-B) with the state-of-the-art method of Inkpen's (Inkpen). In this result, we used the setting of sample-words of Table 2 to obtain the almost same experimental condition as Inkpen's. The suffix "-A" or "-B" denotes the part-of-speech mapping chosen from the two in Table 3. The suffix "+Doc" denotes that we used document-based features in addition to baseline and window features given in Table 1. Similarly, "+Syn" denotes the incorporation of syntax-based features.

The proposed method outperformed the state-of-the-art of Inkpen's against the test sets of nouns and adjectives. Both of Proposed A and Proposed-B make the use of the essentially same feature set as the Inkpen's method. We suppose that the improvement came from the fact that we used a domain-specific corpus when PMI scores are calculated, whereas Inkpen used web derived data of general domain.

Furthermore, the incorporation of document-based features slightly improved the performance of the proposed method in nouns and adjectives. However, the same incorporation was not effective for the verbs in the test set of sample-words.

In all-words task, in contrast to sample-words task, the incorporation of both of document-based and syntax-based features improved the accuracy significantly.

6 DISCUSSION AND FUTURE DIRECTIONS

6.1 *Differences between Sample-words and All-words*

Our models showed substantially different results between sample-words and all-words tasks. An important difference was that syntax-based features improved the results of all-words task. We suppose that this is caused by the difference of quality of candidate sets. In fact, in sample-words

Table 5. Accuracy for sample-words task on BLLIP corpus of Wall Street Journal 1987

	Acc. (noun and adj.)	Acc. (all)
Inkpen	70.01	65.16
F0	70.01	65.16
A	70.95	58.29
B	72.03	65.03
A +Doc	71.09	58.18
B +Doc	72.19	64.63
A +Doc+Syn	65.96	50.73
B +Doc+Syn	64.13	45.49

Table 6. Accuracy for All-words task on Penn Treebank

	Acc. (noun and adj.)	Acc. (all)	#samples (noun and adj.)	#samples
A	79.23	75.07	86179	122131
A +Doc	81.20	77.00	86179	122131
A +Doc+Syn	81.59	77.47	86179	122131

task, candidate substitutions has really similar usages including subcategorization of arguments. Similar claim has been mentioned in [4], where it was claimed that Edmonds' sets were close enough to WordNet synsets. In contrast to that, candidate substitutions in all-words task are more diverse including strong polysemy that can have clearly separated word usage associated with the context. We suppose that subtle distinction of among the sets like WordNet synsets may not lead to benefit of application tasks including term expansion, while the effectiveness of syntax-based and document-based features in all-words is a promising result towards such applications.

6.2 *Contribution of document-based features*

Document-based features improved the performance both in all-words and sample-words tasks. This suggests that it is important to catch the consistency of terminology in choosing a word from synonymous candidates.

However, we should be careful about the fact that we took point-wise prediction strategy, which means that in the selection for gap, we assumed correct predictions in other gaps. In real applications, we may want to perform lexical choice for different gaps at the same time. In such cases, point-wise assumption may not be appropriate.

6.3 *Lexical variation and ambiguity*

People use different terminologies to tell the same thing, according to the writer's and/or reader's background. This fact suffers IR systems through two ways; lexical variation and lexical ambiguity. Lexical variation in natural language texts, including spelling variation, abbreviation and aliases, makes it difficult for IR systems to find the relevance between queries and documents. Lexical ambiguity, which sometimes cooccurs with lexical variation, is the problem that a term can refer to more than one thing. A typical example of lexical ambiguity is the word *pitch*, which can refer to a throw of a ball, to the property that describes the frequency of sound, or to a sticky substance secreted by trees. There is a need for disambiguating ambiguous terms in queries or documents, since they can cause wrong association between queries in IR systems.

A common approach to solving lexical variation is the expansion of terms used in documents and queries. In expansion approaches to lexical variation, there is a risk of introducing noise in inappropriately expanded terms via a non-relevant sense to the context.

To solve lexical ambiguity and lexical variation at the same time, many researchers have been trying to effectively apply word sense disambiguation (WSD) to IR. In the early attempts in applying WSD to IR, while researchers found there was ambiguous terms in real environment of IR, they achieved little improvement [13]. By introducing artificially created pseudo ambiguous words, Sanderson analyzed the effectiveness of WSD, changing the query length and the accuracy of automatic WSD [14]. He found that, as for query length, WSD was more effective on queries with less terms, such as one or two; as for WSD accuracy, he concluded that at least 90% accuracy is required to make improvements in IR performance.

An inherent limitation of WSD-based approaches to lexical variation is that it is not clear what level of granularity is most appropriate for IR. It may be even better to have different level of granularity according to the domain or topic of specific applications, than to have a single consistent criteria. For example, WordNet[10] version 3.0 gives two distinct senses to the word *Atlanta*. One refers to the capital of Georgia in the United States of America. The other refers to a historic battle during the American Civil War. On the other hand, Wikipedia provides more than thirty referents including eighteen place names and five ship names, for the word *Atlanta* [6].

Lexical substitution has been studied as a "relaxed" version of word sense disambiguation, in which one do not prepare a predefined set of senses, or sense inventory. In the approach of lexical substitution, instead of having sense inventories, one tries to find a set of possible substitution for a word and select the best candidate given a specific context of the word token [15].

6.4 *Remaining problems*

The most important part missing in this work is finer evaluation of our method. Our evaluation framework is not as accurate as those used in word sense disambiguation or paraphrasing, in which human annotated corpora or human judgement is used. Since our evaluation measure misjudges some correct answers as errors, as discussed in Section 6, the effectiveness of document-based and syntax-based features should be further investigated in the evaluation framework with human judgment such as lexical substitution tasks [15].

Meanwhile, we also can evaluate our method in terms of performance on real-world applications including machine translation and information retrieval. Real-world applications might not be affected by the noise in the training data, especially when the size of training data is large.

7 Conclusion

We have seen that in our all-words task that has larger vocabulary of candidate substitutions, rich features—including document-based and syntax-based features—improved the performance of context sensitive lexical choice.

[6] Atlanta (disambiguation). (2009, August 17). In *Wikipedia, the free encyclopedia*. Retrieved on 13:47, October 1, 2009, from http://en.wikipedia.org/w/index.php?title=Atlanta_(disambiguation)&oldid=308397379.

REFERENCES

1. Yamamoto, K.: Machine translation by interaction between paraphraser and transfer. Proceedings of the 19th International Conference on Computational Linguistics (COLING), 2002 (2002)
2. Bangalore, S., Rambow, O.: Corpus-based lexical choice in natural language generation. In: ACL. (2000)
3. Edmonds, P.: Choosing the word most typical in context using a lexical co-occurrence network. In: Proceedings of the Europiean Association of Computational Linguistics. (1997)
4. Inkpen, D.: A statistical model for near-synonym choice. ACM Trans. Speech Lang. Process. **4**(1) (2007) 2
5. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Comput. Linguist. **16**(1) (1990) 22–29
6. Gardiner, M., Dras, M.: Exploring approaches to discriminating among near-synonyms. In: Proceedings of the Australasian Language Technology Workshop. (2007)
7. Connor, M., Roth, D.: Context sensitive paraphrasing with a global unsupervised classifier. In: Proceedings of the 18th European conference on Machine Learning, Berlin, Heidelberg, Springer-Verlag (2007) 104–115
8. McCarthy, D., Navigli, R.: Semeval-2007 task 10: English lexical substitution task. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, Association for Computational Linguistics (June 2007) 48–53
9. Manning, C.D., Raghavan, P., Schtze, H.: Relevance feedback and query expansion. In: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
10. Fellbaum, C.: WordNet: An Electornic Lexical Database. Bradford Books (1998)
11. Berger, A.L., Pietra, V.J.D., Pietra, S.A.D.: A maximum entropy approach to natural language processing. Comput. Linguist. **22**(1) (1996) 39–71
12. Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The Penn Treebank: annotating predicate argument structure. In: HLT '94: Proceedings of the workshop on Human Language Technology, Morristown, NJ, USA, Association for Computational Linguistics (1994) 114–119
13. Krovetz, R., Croft, W.B.: Lexical ambiguity and information retrieval. ACM Trans. Inf. Syst. **10**(2) (1992) 115–141
14. Sanderson, M.: Word sense disambiguation and information retrieval. In: SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, Springer-Verlag New York, Inc. (1994) 142–151
15. McCarthy, D., Navigli, R.: The English lexical substitution task. Language Resources and Evaluation (2009)

Yusuke Matsubara
Department of Computer Science,
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo,
Japan
E-mail: <matubara@is.s.u-tokyo.ac.jp>

Jun'ichi Tsujii
Department of Computer Science,
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo,
Japan
and
National Centre for Text Mining,
Manchester Interdisciplinary Biocentre,
131 Princess Street, Manchester,
UK
E-mail: <tsujii@is.s.u-tokyo.ac.jp>