MRD-based Word Sense Disambiguation: Further$_{\#2}$ Extending$_{\#1}$ Lesk

Timothy Baldwin,♠ Su Nam Kim,♠ Francis Bond,♡ Sanae Fujita,♢
David Martinez♠ and Takaaki Tanaka♢

♠ CSSE
University of Melbourne
VIC 3010 Australia

♡ NICT
3-5 Hikaridai, Seika-cho
Soraku-gun, Kyoto
619-0289 Japan

♢ NTT CS Labs
2-4 Hikari-dai, Seika-cho
Soraku-gun, Kyoto
619-0237 Japan

Abstract

This paper reconsiders the task of MRD-based word sense disambiguation, in extending the basic Lesk algorithm to investigate the impact on WSD performance of different tokenisation schemes, scoring mechanisms, methods of gloss extension and filtering methods. In experimentation over the Lexeed Sensebank and the Japanese Senseval-2 dictionary task, we demonstrate that character bigrams with sense-sensitive gloss extension over hyponyms and hypernyms enhances WSD performance.

1 Introduction

The aim of this work is to develop and extend word sense disambiguation (WSD) techniques to be applied to all words in a text. The goal of WSD is to link occurrences of ambiguous words in specific contexts to their meanings, usually represented by a machine readable dictionary (MRD) or a similar lexical repository. For instance, given the following Japanese input:

(1) おとなしい 犬 を 飼いたい
quiet dog ACC want to keep
"(I) want to keep a quiet dog"

we would hope to identify each component word as occurring with the sense corresponding to the indicated English glosses.

WSD systems can be classified according to the knowledge sources they use to build their models. A top-level distinction is made between supervised and unsupervised systems. The former rely on training instances that have been hand-tagged, while the latter rely on other types of knowledge, such as lexical databases or untagged corpora. The Senseval evaluation tracks have shown that supervised systems perform better when sufficient training data is available, but they do not scale well to all words in context. This is known as the knowledge acquisition bottleneck, and is the main motivation behind research on unsupervised techniques (Mihalcea and Chklovski, 2003).

In this paper, we aim to exploit an existing lexical resource to build an all-words Japanese word-sense disambiguator. The resource in question is the Lexeed Sensebank (Tanaka et al., 2006) and consists of the 28,000 most familiar words of Japanese, each of which has one or more basic senses. The senses take the form of a dictionary definition composed from the closed vocabulary of the 28,000 words contained in the dictionary, each of which is further manually sense annotated according to the Lexeed sense inventory. Lexeed also has a semi-automatically constructed ontology.

Through the Lexeed sensebank, we investigate a number of areas of general interest to the WSD community. First, we test extensions of the Lesk algorithm (Lesk, 1986) over Japanese, focusing specifically on the impact of the overlap metric and segment representation on WSD performance. Second, we propose further extensions of the Lesk algorithm that make use of disambiguated definitions. In this, we shed light on the relative benefits we can expect from hand-tagging dictionary definitions, i.e. in introducing "semi-supervision" to the disambiguation task. The proposed method is language independent, and is equally applicable to the Extended WordNet[1] for English, for example.

2 Related work

Our work focuses on unsupervised and semi-supervised methods that target all words and parts of speech (POS) in context. We use the term "unsupervised" to refer to systems that do not use hand-tagged example sets for each word, in line with the standard usage in the WSD literature (Agirre and Edmonds, 2006). We blur the supervised/unsupervised boundary somewhat in combining the basic unsupervised methods with hand-tagged definitions from Lexeed, in order to measure the improvement we can expect from sense-tagged data. We qualify our use of hand-tagged definition

[1] http://xwn.hlt.utdallas.edu

Baldwin, Timothy, Su Nam Kim, Francis Bond, Sanae Fujita, David Martinez and Takaaki Tanaka (2008) MRD-based Word Sense Disambiguation: Further Extending Lesk, In Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP 2008), Hyderabad, India.

sentences by claiming that this kind of resource is less costly to produce than sense-annotated open text because: (1) the effects of discourse are limited, (2) syntax is relatively simple, (3) there is significant semantic priming relative to the word being defined, and (4) there is generally explicit meta-tagging of the domain in technical definitions. In our experiments, we will make clear when hand-tagged sense information is being used.

Unsupervised methods rely on different knowledge sources to build their models. Primarily the following types of lexical resources have been used for WSD: MRDs, lexical ontologies, and untagged corpora (monolingual corpora, second language corpora, and parallel corpora). Although early approaches focused on exploiting a single resource (Lesk, 1986), recent trends show the benefits of combining different knowledge sources, such as hierarchical relations from an ontology and untagged corpora (McCarthy et al., 2004). In this summary, we will focus on a few representative systems that make use of different resources, noting that this is an area of very active research which we cannot do true justice to within the confines of this paper.

The Lesk method (Lesk, 1986) is an MRD-based system that relies on counting the overlap between the words in the target context and the dictionary definitions of the senses. In spite of its simplicity, it has been shown to be a hard baseline for unsupervised methods in Senseval, and it is applicable to all-words with minimal effort. Banerjee and Pedersen (2002) extended the Lesk method for WordNet-based WSD tasks, to include hierarchical data from the WordNet ontology (Fellbaum, 1998). They observed that the hierarchical relations significantly enhance the basic model. Both these methods will be described extensively in Section 3.1, as our approach is based on them.

Other notable unsupervised and semi-supervised approaches are those of McCarthy et al. (2004), who combine ontological relations and untagged corpora to automatically rank word senses in relation to a corpus, and Leacock et al. (1998) who use untagged data to build sense-tagged data automatically based on monosemous words. Parallel corpora have also been used to avoid the need for hand-tagged data, e.g. by Chan and Ng (2005).

3 Background

As background to our work, we first describe the basic and extended Lesk algorithms that form the core of our approach. Then we present the Lexeed lexical resource we have used in our experiments, and finally we outline aspects of Japanese relevant for this work.

3.1 Basic and Extended Lesk

The original Lesk algorithm (Lesk, 1986) performs WSD by calculating the relative word overlap between the context of usage of a target word, and the dictionary definition of each of its senses in a given MRD. The sense with the highest overlap is then selected as the most plausible hypothesis.

An obvious shortcoming of the original Lesk algorithm is that it requires that the exact words used in the definitions be included in each usage of the target word. To redress this shortcoming, Banerjee and Pedersen (2002) extended the basic algorithm for WordNet-based WSD tasks to include hierarchical information, i.e. expanding the definitions to include definitions of hypernyms and hyponyms of the synset containing a given sense, and assigning the same weight to the words sourced from the different definitions.

Both of these methods can be formalised according to the following algorithm, which also forms the basis of our proposed method:

---

for each word w_i in context $\mathbf{w} = w_1 w_2 ... w_n$ **do**
  for each sense $s_{i,j}$ and definition $\mathbf{d}_{i,j}$ of w_i **do**
    $score(s_{i,j}) = overlap(\mathbf{w}, \mathbf{d}_{i,j})$
  end for
  $s_i^* = \arg\max_{\mathrm{j}}\ score(s_{i,j})$
end for

3.2 The Lexeed Sensebank

All our experimentation is based on the Lexeed Sensebank (Tanaka et al., 2006). The Lexeed Sensebank consists of all Japanese words above a certain level of familiarity (as defined by Kasahara et al. (2004)), giving rise to 28,000 words in all, with a total of 46,000 senses which are similarly filtered for similarity. The sense granularity is relatively coarse for most words, with the possible exception of light verbs, making it well suited to open-domain applications. Definition sentences for these senses were rewritten to use only the closed vocabulary of the 28,000 familiar words (and some function words). Additionally, a single example sentence was manually constructed to exemplify each of the 46,000 senses, once again using the closed vocabulary of the Lexeed dictionary. Both the definition sentences and example sentences were then manually sense annotated by 5 native speakers of Japanese, from which a majority sense was extracted.

In addition, an ontology was induced from the Lexeed dictionary, by parsing the first definition sentence for each sense (Nichols et al., 2005). Hypernyms were determined by identifying the highest scoping real predicate (i.e. the genus). Other relation types such as synonymy and domain were also induced based on trigger patterns in the definition sentences, although these are too few to be useful in our research. Because each word is sense tagged, the relations link senses rather than just words.

3.3 Peculiarities of Japanese

The experiments in this paper focus exclusively on Japanese WSD. Below, we outline aspects of Japanese which are relevant to the task.

First, Japanese is a non-segmenting language, i.e. there is no explicit orthographic representation of word boundaries. The native rendering of (1), e.g., is おとなしい犬を飼いたい. Various packages exist to automatically segment Japanese strings into words, and the Lexeed data has been pre-segmented using ChaSen (Matsumoto et al., 2003).

Second, Japanese is made up of 3 basic alphabets: hiragana, katakana (both syllabic in nature) and kanji (logographic in nature). The relevance of these first two observations to WSD is that we can choose to represent the context of a target word by way of characters or words.

Third, Japanese has relatively free word order, or strictly speaking, word order within phrases is largely fixed but the ordering of phrases governed by a given predicate is relatively free.

4 Proposed Extensions

We propose extensions to the basic Lesk algorithm in the orthogonal areas of the scoring mechanism, tokenisation, extended glosses and filtering.

4.1 Scoring Mechanism

In our algorithm, *overlap* provides the means to score a given pairing of context $\mathbf{w}$ and definition $\mathbf{d}_{i,j}$. In the original Lesk algorithm, *overlap* was simply the sum of words in common between the two, which Banerjee and Pedersen (2002) modified by squaring the size of each overlapping sub-string. While squaring is well motivated in terms of preferring larger substring matches, it makes the algorithm computationally expensive. We thus adopt a cheaper scoring mechanism which normalises relative to the length of $\mathbf{w}$ and $\mathbf{d}_{i,j}$, but ignores the length of sub-string matches. Namely, we use the Dice coefficient.

4.2 Tokenisation

Tokenisation is particularly important in Japanese because it is a non-segmenting language with a logographic orthography (kanji). As such, we can chose to either word tokenise via a word splitter such as ChaSen, or character tokenise. Character and word tokenisation have been compared in the context of Japanese information retrieval (Fujii and Croft, 1993) and translation retrieval (Baldwin, 2001), and in both cases, characters have been found to be the superior representation overall.

Orthogonal to the question of whether to tokenise into words or characters, we adopt an n-gram segment representation, in the form of simple unigrams and simple bigrams. In the case of word tokenisation and simple bigrams, e.g., example (1) would be represented as { おとなしい犬 , 犬を , を飼いたい }.

4.3 Extended Glosses

The main direction in which Banerjee and Pedersen (2002) successfully extended the Lesk algorithm was in including hierarchically-adjacent glosses (i.e. hyponyms and hypernyms). We take this a step further, in using both the Lexeed ontology and the sense-disambiguated words in the definition sentences.

The basic form of extended glossing is the simple Lesk method, where we take the simple definitions for each sense $s_{i,j}$ (i.e. without any gloss extension).

Next, we replicate the Banerjee and Pedersen (2002) method in extending the glosses to include words from the definitions for the (immediate) hypernyms and/or hyponyms of each sense $s_{i,j}$.

An extension of the Banerjee and Pedersen (2002) method which makes use of the sense-annotated definitions is to include the words in the definition of each sense-annotated word d_k contained in definition $\mathbf{d}_{i,j} = d_1 d_2 ... d_m$ of word sense $s_{i,j}$. That is, rather than traversing the ontology relative to each word sense candidate $s_{i,j}$ for the target word w_i, we represent each word sense via the original definition plus all definitions of word senses contained in it (weighting each to give the words in the original definition greater import than those from definitions of those word senses). We can then optionally adopt a similar policy to Banerjee and Pedersen (2002) in expanding each sense-annotated word d_k in the original definition relative to the ontology, to include the immediate hypernyms and/or hyponyms.

We further expand the definitions (+extdef) by adding the full definition for each sense-tagged word in the original definition. This can be combined with the Banerjee and Pedersen (2002) method by

also expanding each sense-annotated word d_k in the original definition relative to the ontology, to include the immediate hypernyms (+hyper) and/or hyponyms (+hypo).

4.4 Filtering

Each word sense in the dictionary is marked with a word class, and the word splitter similarly POS tags every definition and input to the system. It is natural to expect that the POS tag of the target word should match the word class of the word sense, and this provides a coarse-grained filter for discriminating homographs with different word classes.

We also experiment with a stop word-based filter which ignores a closed set of 18 lexicographic markers commonly found in definitions (e.g. 略 [*ryaku*] "an abbreviation for ..."), in line with those used by Nichols et al. (2005) in inducing the ontology.

5 Evaluation

We evaluate our various extensions over two datasets: (1) the example sentences in the Lexeed sensebank, and (2) the Senseval-2 Japanese dictionary task (Shirai, 2002).

All results below are reported in terms of simple precision, following the conventions of Senseval evaluations. For all experiments, precision and recall are identical as our systems have full coverage.

For the two datasets, we use two baselines: a random baseline and the first-sense baseline. Note that the first-sense baseline has been shown to be hard to beat for unsupervised systems (McCarthy et al., 2004), and it is considered supervised when, as in this case, the first-sense is the most frequent sense from hand-tagged corpora.

5.1 Lexeed Example Sentences

The goal of these experiments is to tag all the words that occur in the example sentences in the Lexeed Sensebank. The first set of experiments over the Lexeed Sensebank explores three parameters: the use of characters vs. words, unigrams vs. bigrams, and original vs. extended definitions. The results of the experiments and the baselines are presented in Table 1.

First, characters are in all cases superior to words as our segment granularity. The introduction of bigrams has a uniformly negative impact for both characters and words, due to the effects of data sparseness. This is somewhat surprising for characters, given that the median word length is 2 characters, although the difference between character unigrams and bigrams is slight.

Extended definitions are also shown to be superior to simple definitions, although the relative increment in making use of large amounts of sense annotations is smaller than that of characters vs. words, suggesting that the considerable effort in sense annotating the definitions is not commensurate with the final gain for this simple method.

Note that at this stage, our best-performing method is roughly equivalent to the unsupervised (random) baseline, but well below the supervised (first sense) baseline.

Having found that extended definitions improve results to a small degree, we turn to our next experiment were we investigate whether the introduction of ontological relations to expand the original definitions further enhances our precision. Here, we persevere with the use of word and characters (all unigrams), and experiment with the addition of hypernyms and/or hyponyms, with and without the extended definitions. We also compare our method directly with that of Banerjee and Pedersen (2002) over the Lexeed data, and further test the impact of the sense annotations, in rerunning our experiments with the ontology in a sense-*insensitive* manner, i.e. by adding in the union of word-level hypernyms and/or hyponyms. The results are described in Table 2. The results in brackets are reproduced from earlier tables.

Adding in the ontology makes a significant difference to our results, in line with the findings of Banerjee and Pedersen (2002). Hyponyms are better discriminators than hypernyms (assuming a given word sense has a hyponym – the Lexeed ontology is relatively flat), partly because while a given word sense will have (at most) one hypernym, it often has multiple hyponyms (if any at all). Adding in hypernyms or hyponyms, in fact, has a greater impact on results than simple extended definitions (+extdef), especially for words. The best overall results are produced for the (weighted) combination of all ontological relations (i.e. extended definitions, hypernyms and hyponyms), achieving a precision level above both the unsupervised (random) and supervised (first-sense) baselines.

In the interests of getting additional insights into the import of sense annotations in our method, we ran both the original Banerjee and Pedersen (2002) method and a sense-insensitive variant of our proposed method over the same data, the results for which are also included in Table 2. Simple hyponyms (without extended definitions) and word-based segments returned the best results out of all the variants tried, at a precision of 0.656. This compares with a precision of 0.683 achieved for the best

	UNIGRAMS		BIGRAMS	
	ALL WORDS	POLYSEMOUS	ALL WORDS	POLYSEMOUS
Simple Definitions				
CHARACTERS	0.523	0.309	0.486	0.262
WORDS	0.469	0.229	0.444	0.201
Extended Definitions				
CHARACTERS	**0.526**	**0.313**	0.529	0.323
WORDS	0.489	0.258	0.463	0.227

Table 1: Precision over the Lexeed example sentences using simple/extended definitions and word/character unigrams and bigrams (best-performing method in **boldface**)

		ALL WORDS	POLYSEMOUS
UNSUPERVISED BASELINE:		0.527	0.315
SUPERVISED BASELINE:		0.633	0.460
Banerjee and Pedersen (2002)		0.648	0.492
Ontology expansion (sense-sensitive)			
	simple	(0.469)	(0.229)
	+extdef	(0.489)	(0.258)
	+hypernyms	0.559	0.363
W	+hyponyms	0.655	0.503
	+def +hyper	0.577	0.386
	+def +hypo	0.649	0.490
	+def +hyper +hypo	**0.683**	**0.539**
	simple	(0.523)	(0.309)
	+extdef	(0.526)	(0.313)
	+hypernyms	0.539	0.334
C	+hyponyms	0.641	0.481
	+def +hyper	0.563	0.365
	+def +hypo	0.671	0.522
	+def +hyper +hypo	0.671	0.522
Ontology expansion (sense-insensitive)			
	+hypernyms	0.548	0.348
	+hyponyms	0.656	0.503
W	+def +hyper	0.551	0.347
	+def +hypo	0.649	0.490
	+def + hyper +hypo	0.631	0.464
	+hypernyms	0.537	0.332
	+hyponyms	0.644	0.485
C	+def +hyper	0.542	0.335
	+def +hypo	0.644	0.484
	+def + hyper +hypo	0.628	0.460

Table 2: Precision over the Lexeed example sentences using ontology-based gloss extension (with/without word sense information) and word (W) and character (C) unigrams (best-performing method in **boldface**)

of the sense-sensitive methods, indicating that sense information enhances WSD performance. This reinforces our expectation that richly annotated lexical resources improve performance. With richer information to work with, character based methods uniformly give worse results.

While we don't present the results here due to reasons of space, POS-based filtering had very little impact on results, due to very few POS-differentiated homographs in Japanese. Stop word filtering leads to a very slight increment in precision across the board (of the order of 0.001).

		ALL WORDS	POLYSEMOUS
Baselines			
	Unsupervised (random)	0.310	0.260
	Supervised (first-sense)	0.577	0.555
Ontology expansion (sense-sensitive)			
W	+def +hyper +hypo	**0.624**	**0.605**
C	+def +hyper +hypo	**0.624**	**0.605**
Ontology expansion (sense-insensitive)			
W	+def +hyper +hypo	0.602	0.581
C	+def +hyper +hypo	0.593	0.572

Table 3: Precision over the Senseval-2 data

5.2 Senseval-2 Japanese Dictionary Task

In our second set of experiments we apply our proposed method to the Senseval-2 Japanese dictionary task (Shirai, 2002) in order to calibrate our results against previously published results for Japanese WSD. Recall that this is a lexical sample task, and that our evaluation is relative to Lexeed re-annotations of the same dataset, although the relative polysemy for the original data and the re-annotated version are largely the same (Tanaka et al., 2006). The first sense baselines (i.e. sense skewing) for the two sets of annotations differ significantly, however, with a precision of 0.726 reported for the original task, and 0.577 for the re-annotated Lexeed variant. System comparison (Senseval-2 systems vs. our method) will thus be reported in terms of error rate reduction relative to the respective first sense baselines.

In Table 3, we present the results over the Senseval-2 data for the best-performing systems from our earlier experiments. As before, we include results over both words and characters, and with sense-sensitive and sense-insensitive ontology expansion.

Our results largely mirror those of Table 2, although here there is very little to separate words and characters. All methods surpassed both the random and first sense baselines, but the relative impact

of sense annotations was if anything even less pronounced than for the example sentence task.

Both sense-sensitive WSD methods achieve a precision of 0.624 over all the target words (with one target word per sentence), an error reduction rate of 11.1%. This compares favourably with an error rate reduction of 21.9% for the best of the WSD systems in the original Senseval-2 task (Kurohashi and Shirai, 2001), particularly given that our method is semi-supervised while the Senseval-2 system is a conventional supervised word sense disambiguator.

6 Conclusion

In our experiments extending the Lesk algorithm over Japanese data, we have shown that definition expansion via an ontology produces a significant performance gain, confirming results by Banerjee and Pedersen (2002) for English. We also explored a new expansion of the Lesk method, by measuring the contribution of sense-tagged definitions to overall disambiguation performance. Using sense information doubles the error reduction compared to the supervised baseline, a constant gain that shows the importance of precise sense information for error reduction.

Our WSD system can be applied to all words in running text, and is able to improve over the first-sense baseline for two separate WSD tasks, using only existing Japanese resources. This full-coverage system opens the way to explore further enhancements, such as the contribution of extra sense-tagged examples to the expansion, or the combination of different WSD algorithms.

For future work, we are also studying the integration of the WSD tool with other applications that deal with Japanese text, such as a cross-lingual glossing tool that aids Japanese learners reading text. Another application we are working on is the integration of the WSD system with parse selection for Japanese grammars.

Acknowledgements

This material is supported by the Research Collaboration between NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and the University of Melbourne. We would like to thank members of the NTT Machine Translation Group and the three anonymous reviewers for their valuable input on this research.

References

Eneko Agirre and Philip Edmonds, editors. 2006. *Word Sense Disambiguation: Algorithms and Applications*. Springer, Dordrecht, Netherlands.

Timothy Baldwin. 2001. Low-cost, high-performance translation retrieval: Dumber is better. In *Proc. of the 39th Annual Meeting of the ACL and 10th Conference of the EACL (ACL-EACL 2001)*, pages 18–25, Toulouse, France.

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proc. of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002)*, pages 136–45, Mexico City, Mexico.

Yee Seng Chan and Hwee Tou Ng. 2005. Scaling up word sense disambiguation via parallel texts. In *Proc. of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pages 1037–42, Pittsburgh, USA.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA.

Hideo Fujii and W. Bruce Croft. 1993. A comparison of indexing techniques for Japanese text retrieval. In *Proc. of 16th International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, pages 237–46, Pittsburgh, USA.

Kaname Kasahara, Hiroshi Sato, Francis Bond, Takaaki Tanaka, Sanae Fujita, Tomoko Kanasugi, and Shigeaki Amano. 2004. Construction of a Japanese semantic lexicon: Lexeed. In *Proc. of SIG NLC-159*, Tokyo, Japan.

Sadao Kurohashi and Kiyoaki Shirai. 2001. SENSEVAL-2 Japanese tasks. In *IEICE Technical Report NLC 2001-10*, pages 1–8. (in Japanese).

Claudia Leacock, Martin Chodorow, and George A. Miller. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–65.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proc. of the 1986 SIGDOC Conference*, pages 24–6, Ontario, Canada.

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka, and Masayuki Asahara. 2003. *Japanese Morphological Analysis System ChaSen Version 2.3.3 Manual*. Technical report, NAIST.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant senses in untagged text. In *Proc. of the 42nd Annual Meeting of the ACL*, pages 280–7, Barcelona, Spain.

Rada Mihalcea and Timothy Chklovski. 2003. Open Mind Word Expert: Creating Large Annotated Data Collections with Web Users' Help. In *Proceedings of the EACL 2003 Workshop on Linguistically Annotated Corpora (LINC 2003)*, pages 53–61, Budapest, Hungary.

Eric Nichols, Francis Bond, and Daniel Flickinger. 2005. Robust ontology acquisition from machine-readable dictionaries. In *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI-2005)*, pages 1111–6, Edinburgh, UK.

Kiyoaki Shirai. 2002. Construction of a word sense tagged corpus for SENSEVAL-2 japanese dictionary task. In *Proc. of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 605–8, Las Palmas, Spain.

Takaaki Tanaka, Francis Bond, and Sanae Fujita. 2006. The Hinoki sensebank — a large-scale word sense tagged corpus of Japanese —. In *Proc. of the Workshop on Frontiers in Linguistically Annotated Corpora 2006*, pages 62–9, Sydney, Australia.