# Japanese all-words WSD system using the Kyoto Text Analysis ToolKit

**Hiroyuki Shinnou  Kanako Komiya  Minoru Sasaki  Shinsuke Mori**
Ibaraki University, Department of Computer and Information Sciences
4-12-1 Nakanarusawa, Hitachi, Ibaraki 316–8511, Japan
hiroyuki.shinnou.0828@vc.ibaraki.ac.jp
{kanako.komiya.nlp, minoru.sasaki.01}@vc.ibaraki.ac.jp
Kyoto University, Yoshida Honcho, Sakyouku, Kyoto 606–8501, Japan
mori.shinsuke.8u@kyoto-u.ac.jp

## Abstract

In this paper, we discuss Japanese all-words word sense disambiguation (WSD) and propose a new system KyWSD to achieve it. KyWSD uses the Kyoto Text Analysis ToolKit, a learning system building a Japanese morphological analysis model. It accepts plain Japanese text, segments it into words, and assigns a sense to each segmented word. KyWSD is open source software that can serve as the baseline system for a Japanese all-words WSD system. Therefore, it can be useful for several Japanese semantic analysis systems and is an advancement in all-words WSD technology. Furthermore, we show that Japanese all-words WSD involves a peculiar problem different from those of general WSD and that KyWSD is adaptable and highly precise.

## 1 Introduction

Word-sense disambiguation (WSD) is a basic procedure of semantic analysis, but it has not been widely used in practice. This is because current WSD systems adopt a supervised learning approach, limiting WSD target words. WSD for all words called "all-words WSD" has been studied for a long time (Navigli, 2009). However, a sense in many all-words WSD systems is defined as a concept, resulting in coarse granularity. Furthermore, the target language is generally English. Japanese all-words WSD has not been achieved, preventing easy access to it. Given this background, we created a Japanese all-words WSD system called *KyWSD*[1]. KyWSD is useful for several Japanese semantic analysis systems. Using it, we can add sense features when we use a learning method to solve various NLP tasks, thereby improving precision.

The substance of KyWSD is a model built using the Kyoto Text Analysis ToolKit (KyTea)[2], a learning system. By executing KyTea using this model, KyWSD accepts plain Japanese text, segments it into words, and assigns a sense to each segmented word (Neubig et al., 2011). Briefly KyTea is a system learning a morphological analysis model. We build KyWSD using KyTea because all-words WSD can be regarded as a kind of morphological analysis. Therefore, KyTea contains a mechanism for learning a model to adapt to a target domain. The ability to use this mechanism provides KyWSD with high adaptability. For example, adding training data to KyWSD, senses to all words, but a target sense. Thus, KyWSD is an appropriate system for domain adaptation. As seen above, KyWSD provides great value as new use of KyTea.

We evaluated KyWSD using a Japanese dictionary task in Senseval-2 (Kiyoaki Shirai, 2001). Adding training data of this task to its original training data enabled KyWSD to perform better than a general supervised support vector machine (SVM) based learning method. This evaluation revealed a peculiar problem of Japanese all-words WSD through which it differs from general WSD.

[1] KyTea for WSD.

[2] http://www.phontron.com/kytea/

## 2 Related Work

The availability of a supervised learning method for all-words WSD typically requires specifying the domain. Some systems using a supervised learning method have used all-words WSD tasks of SemEval-07 (Navigli et al., 2007), but these systems have a problem with scalability.

All-words WSD methods not using a supervised learning method are divided into two types: knowledge based methods and unsupervised learning methods (Kulkarni et al., 2010).

Lesk's method (Lesk, 1986), a well known classical knowledge based method uses a dictionary in which each sense of every word is provided with definition sentences. Lesk's method counts the overlapping words that are between the words used in the definition sentence and words that are surrounding the target word in the test sentence. Finally, the sense with the largest overlapping is selected. However, a knowledge based method generally cannot make use of the distribution of senses, resulting in low precision.

There are various unsupervised learning methods (Yarowsky, 1995; Izquierdo-Beviá et al., 2006; Zhong and Ng, 2009). Recently, methods using a generative model have been studied (Boyd-Graber et al., 2007; Tanigaki et al., 2013; Tanigaki et al., 2015; Komiya et al., 2015). These methods have higher precision than knowledge based methods, in general, and can be expected to improve in the future. However, current unsupervised learning methods have the problem that the sense assigned to a word is a concept, because such a method essentially uses the following heuristic: "If the context surrounding the sense $a$ is similar to the context surrounding the sense $b$, then $a$ is similar to $b$." In general, $a$ and $b$ are ambiguous, so we must measure the distance between $a$ and $b$ to use this heuristic. In the case which $a$ and $b$ are concepts, we can measure that distance. However, if $a$ and $b$ are senses defined in a dictionary, we cannot. This is the problem of sense granularity. In general, a sense defined in a dictionary is finer than a concept. Therefore, it is more difficult to assign a sense defined in a dictionary to a word than to assign a concept. KyWSD does the former using "Iwanami Kokugo Jiten[3]."

Furthermore, we must note that the input-output for an all-words WSD system using an unsupervised learning method is different from that of a general WSD system. The input of the former is a corpus, and the output is the same corpus, in which all words are assigned senses. When we input a sentence including a WSD target word to the system, it cannot assign a sense to the target word. This means that we cannot use these all-words WSD systems as general WSD systems.

Recently, methods for using the distributed representation of a word sense for all-words WSD have been studied (Chen et al., 2014)(Neelakantan et al., 2014). Here, we denote the distributed representation of the $i^{th}$ sense of the word $w$ as $s_i$, and the distributed representation of the context of $w$ as $v$. By measuring the similarity between $s_i$ and $v$, we choose the $s_i$ with the greatest similarity as the sense of $w$. This method is knowledge based and therefore has low precision. In general, such knowledge based methods lack the precision of a most frequent sense (MFS) method. A method for estimating MFS using the distributed representation of a word has been studied for this reason (Bhingardive et al., 2015).

KyWSD was constructed using a supervised method. Moreover, a sense in KyWSD is not a concept, but a sense in a dictionary.

Hatori et al. uses a similar supervised approach as KyWSD by treating the all-word WSD task as a sequence labelling problem (Hatori et al., 2008). They also regarded all-words WSD as a sequential labeling problem. To solve it, they used a conditional random field (CRF), but KyWSD uses pointwise prediction. This is the essential difference. Assigning a sense defined in a dictionary to a word, the sense $s$ of a word $w$ is not assigned to any word other than $w$. For this reason, we need not look fully sequentially for all-words WSD to assign a sense defined in a dictionary. Pointwise prediction is all that is required.

## 3 KyTea

All-words WSD is a same problem as part of speech tagging. For example, we do all-words WSD for following word segmented Japanese sentece:

/ 国民/ の/ 声/ を/ 聞く/

[3]"岩波国語辞典 (Iwanami Kokugo Jiten)" is used as a standard dictionary for Japanese WSD task.

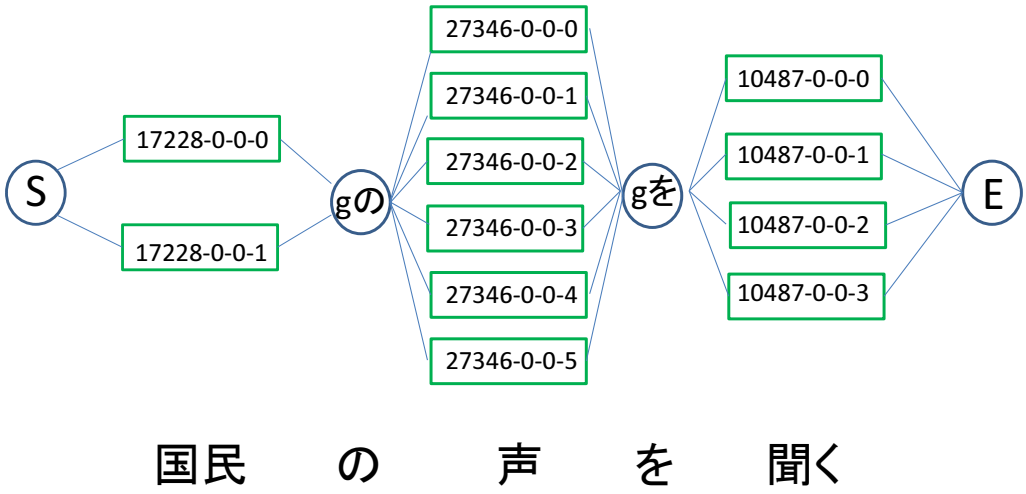


Figure 1: Directed graph for all-words WSD

The word “国民”, “声”, and “聞く” in the sentence has multi senses. Concretely, the sense IDs are ’17228-0-0-0’ and ’17228-0-0-1’ for the word “国民”, ’27346-0-0-0’,’27346-0-0-1’, · · ·, ’27346-0-0-5’ for the word “声”, and ’10487-0-0-0’,’10487-0-0-1’ and ’10487-0-0-3’ for the word “聞く”. The problem of all-words WSD is the estimate of the correct combination of these sense IDs. Regarding the sense ID as the part of speech, all-words WSD is a same problem as part of speech tagging. That is, to solve all-words WSD, we make a directed graph like Figure 1, and then estimate the optimum path from the start node ’S’ to the end node ’E’. To do this effectively, we uses KyTea.

KyTea is essentially a system for building a model of a word segmentation system. In general, a word segmentation problem can be modeled as a sequential labeling problem. However, KyTea models such a problem as a binary classification problem that judges whether each two characters are segmented or not. In learning, KyTea uses only n-grams surrounding the target place using a linear SVM or logistic regression. The training data of KyTea are very simple, just a word-segmented text. Hence, it is easy to scale the model up and adapt the model to another target domain.

We can create a tagger system using KyTea. When a tag is given to a segmented word in the training data, KyTea learns the model for assigning the tag to each segmented word. If the tag is the part of speech, KyTea learns a general morphological analysis model. The tag can be used for far more than the part of speech. For example, the pronunciation and BIO tags for a name recognition task have been used (Neubig and Mori, 2010)(Sasada et al., 2015).

In this study, we set the sense of the word as the tag. Using this setting, we can build an all-words WSD system based on a sense-tagged corpus and KyTea.

## 4 KyWSD

### 4.1 System Overview

KyWSD is a Japanese all-words WSD system, and we can try it on the following demonstration site:

`http://nlp.dse.ibaraki.ac.jp/~shinnou/cgi-bin/demo.html`

The Figure 2 shows a demonstration of KyWSD. Input a Japanese sentence in the text field, and push the ’KyWSD’ button. The analysis result by KyWSD will be shown. In this demonstration site, the given sentence is segmented into words, and the part of speech and the sense ID for each segmented word are assigned.

Note that this demonstration site is just built in order to get the picture of KyWSD. When we use


Figure 2: Demonstration of KyWSD

KyWSD for real, commands are used in the character terminal, that is CUI interface. We can get a set of KyWSD from the following:

```
http://nlp.dse.ibaraki.ac.jp/
    ~shinnou/wsd/kywsd.zip
```

KyWSD is essentially a model of KyTea. Therefore, KyWSD works under the operating system supported by KyTea, Linux, Windows, and Mac OS.

We show an example of KyWSD execution in Figure 2. The `wsd.mod` is the model learned by KyTea. The input is the plain Japanese text file (`sample.txt`), and the output is that shown in Figure 1, i.e., the input texts are segmented into words, and the part of speech and the sense are assigned to each segmented word. However, a sense is assigned for content words, including nouns and verb or adjective, stems, but for other kinds of words the sense s set to "0." The "UNK" in Figure 1 means that the word "ＢＨ" appears in neither the training data nor the dictionary.

KyWSD can omit the first tags, i.e., the POS tags with the option `-notag 1`. In addition, it can output the confidence degree with the option `-out conf`. The confidence degree is the probability when option wsd.mod is used because logistic regression is used for the estimation. For example, KyWSD outputs the word senses for the word "意味" (meaning) as follows.

```
意味/2843-0-0-1&2843-0-0-2&2843-0-0-3
```

This shows that the senses of "意味" are three, i.e., 2843-0-0-1, 2843-0-0-2, and 2843-0-0-3. The confidence degree of the word sense of "意味" is as follows.

```
0.807761&0.108979&0.0807573
```

This shows that the probabilities of the word senses 2843-0-0-1,2843-0-0-2, and 2843-0-0-3 are 0.807761, 0.108979, and 0.0807573, respectively. This degree enables KyWSD to use active learning easily.

### 4.2 Building of KyWSD

KyWSD is built by providing a sense-tagged corpus to KyTea as training data. As the sense-tagged corpus, we used a corpus developed by Okumura Laboratory at Tokyo Institute of Technology. This corpus consists of core data of the Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa, 2007). It contains 1,980 documents of six genres, and all multisense words in them are assigned a sense defined in the dictionary "Iwanami Kokugo Jiten." There are 114,696 assigned senses and 4,916 types of senses.

It is easy to translate the above corpus into training data for KyTea, but there is a problem, Japanese inflection. A Japanese verb and adjective consist of

```
> cat sample.txt
野球のＢＨの正式呼び名と意味を教えてください。

> kytea -model wsd.mod < sample.txt
野球/名詞-普通名詞/51783-0-0-0 の/助詞-格助詞/0 ＢＨ/UNK/UNK の/助詞-格助詞/0
正式/形状詞-一般/0 呼び名/名詞-普通名詞/53605-0-0-0 と/助詞-格助詞/0
意味/名詞-普通名詞/2843-0-0-1 を/助詞-格助詞/0 教え/動詞-一般-語幹/5541-0-0-2
て/助詞-接続助詞/0 くださ/動詞-非自立可能-語幹/13445-0-0-2 い/動詞-非自立可能-語尾/0
。/補助記号-句点/0

>kytea -model wsd.mod -notag 1 -out conf < sample.txt
野球/51783-0-0-0 の/0&39930-0-1-3&40065-0-0-0 ＢＨ/UNK
の/0&39930-0-1-3&39930-0-1-1 正式/0 呼び名/53605-0-0-0
と/0&37713-0-0-1&37446-0-0-2 意味/2843-0-0-1&2843-0-0-2&2843-0-0-3
を/0 教え/5541-0-0-2&5541-0-0-1&5541-0-0-3 て/0&35369-0-0-0
くださ/13445-0-0-2&0 い/0&1707-0-0-2&52935-0-0-3 。/0

    … (omit) …

1 0.999999&7.94354e-07&1.23533e-07 1 1&6.47248e-08&3.92486e-08 1 1
1&1.8927e-09&1.8105e-09 0.807761&0.108979&0.0807573 1
0.863406&0.135187&0.0012201 1&4.35077e-09 0.999236&0.00076433
0.999999&1.22639e-07&8.67671e-08 1
```

Figure 3: Example of KyWSD execution

a stem and a desinence, and the desinence changes depending on modality and tense. In general, a Japanese word segmentation system regards the combined stem and desinence as one word, sometimes resulting in a word having different character sequences. KyTea recognizes these words as different words. For example, the word “書く (write)” changes to “書か (write)” + “ない (not)” when the word is used in the negative form. The word “書く (write)” and “書か (write)” are essentially the same, but have different character sequences. To overcome this problem, we define the stem as the word, i.e. all verbs and adjectives in the corpus are separated into the stem and the desinence. Note that the SA-row irregular verb “する” is an exception, because its stem is not fixed. However, the KA-row irregular verb “来る” is not an exception, because the sound of its stem is not fixed, but its character is fixed as “来”.

KyTea can use dictionaries in learning. A sense of the word not appearing in the training data is assigned by the dictionary. By registering MFS for a word in the dictionary, KyWSD can output MFS as the default sense. KyWSD registers the first sense of a word in “Iwanami Kokugo Jiten.”

## 5 Evaluation

### 5.1 Precision

We evaluated the precision of KyWSD, but it is difficult to measure the precision of an all-words WSD system. Here, we investigate the precision of KyWSD using test data of a Japanese dictionary task in Senseval-2 (Kiyoaki Shirai, 2001). This task has 100 WSD target words (50 nouns and 50 verbs). For each target word, 100 test instances are provided for a total of 10,000 test instances.

First, we investigate the precision of a standard method, a supervised learning method using an SVM. For each target word, 175 training instances are provided on average. Using these training data and the following six features (`e1` to `e6`) for WSD,

[4] we build the SVM classifier for each target word.

**e1:** the word $w_{i-1}$
**e2:** the word $w_{i+1}$
**e3:** two content words in front of $w_i$
**e4:** two content words behind $w_i$
**e5:** thesaurus ID number of e3
**e6:** thesaurus ID number of e4

In 10,000 test instances, 7,244 instances were identified correctly using the above SVM classifiers. This means that the precision (i.e., F-value) of a standard supervised method is 0.7244.

Next, we translate test data to plain text and input it to KyWSD. As a result, every words in the text is assigned its sense. If the target word in a test instance is correctly segmented, and the correct sense is assigned to the word, then we judge it to be a correct answer. Among 10,000 test instances, KyWSD correctly segmented 9,935 target words, and correctly assigned 6,258 senses to them. That is, the precision is 0.6571, the recall is 0.6528, and the F-value is 0.6549.

The F-value of KyWSD is lower than that of an SVM. One reason is that the problem setting of all-words WSD is more difficult than that of general WSD. In general WSD, the sense list $L_w$ of the target word $w$ is given in advance, requiring us to select only one sense in $L_w$. In contrast, $L_w$ is not given in all-words WSD. In Japanese, there are many words with the same character sequence. Therefore, the real sense list $L'_w$ of the target word $w$ in all-words WSD is larger than the $L_w$ in general WSD.

For example, the Japanese word "間" has six types of pronunciation: "あい (21)", "あいだ (105)", "あわい (1432)", "かん (9518)", "けん (15147)" and "ま (48408)" [5]. Each of these six words is listed in "Iwanami Kokugo Jiten" as one word. The word "間" is one of the target words in the Japanese dictionary task in Senseval-2, but the sense listed for this word is that of the word "あいだ (105)" only. This problem has been ignored in conventional Japanese WSD. However, it is a serious problem in Japanese all-words WSD, and we must take measures to address it.

In the above experiment, KyWSD output 1,372 incorrect senses because KyWSD selected a sense not belonging to $L_w$. If KyWSD does not select such senses, the number of evaluation target instances changes to 8,563, and the correct answers for them number 6,258. Therefore, the precision of KyWSD is 0.7623 and the F-value is 0.7076.

### 5.2 Adaptability

The principal advantage of KyWSD is its ease of adaptation. The new adapted model can be learned as a consequence of adding training data to the current model. In this section, we show this using the above experimental data. In the above experiment, KyWSD did not use the training data provided by that task. Here, we adapt the model of KyWSD by using it for that task. Note that only senses of target words are included in the training data.

As a result, among 10,000 test instances, the new KyWSD correctly segmented 9,938 target words, and assigned 6,986 correct senses for them. That is, the precision is 0.7030, the recall is 0.6986, and the F-value is 0.7008. Moreover, as explained above, ignoring senses not in the sense list provided by that task, there are 6,986 correct senses for 8,953 answered instances. Therefore, the precision is 0.7803, and the F-value is improved to 0.7395. This value is better than that of a supervised learning method using an SVM.

### 5.3 Use for document classification

In this section, we apply KyWSD to document classification.

In document classification, a document is translated to a vector using a bag-of-words model. That is, the learning feature is each word. A word is given a sense using KyWSD. Thus, the sense is added to the learning features.

We downloaded 316 documents from the netnews site:`http://news.goo.ne.jp/`. This document set has five categories: politics, economics, national, society and sports. The classifier is learned using naive Bayes method. We evaluated it using leave-one-out cross validation. Using words as the learning feature, the number of correct classifica-

[4] Suppose that the target word is $w_i$ which is the $i$-th word in the sentence.

[5] The number in a parenthesis means the word ID in "Iwanami Kokugo Jiten."

tions was 246. Using words and senses as the learning feature, the number was 247.

This improvement is only slight, but it is very easy to add a sense to the learning features using KyWSD. Therefore, KyWSD can be used for far more than document classification.

## 6 Conclusion

In this paper, we introduced the Japanese all-words WSD system called KyWSD, which we produced and launched. KyWSD uses KyTea, a learning system for building a Japanese morphological analysis model. KyWSD provides great value as new use of KyTea. KyWSD estimates senses using pointwise prediction. It is simple, and adapting the model to another domain is easy. Through experiments, we showed that the precision of KyWSD is comparable to that of a supervised learning method, and that Japanese all-words WSD has a peculiar problem different from those of general WSD.

KyWSD is useful for many Japanese semantic analysis systems, and can add senses to the learning features of various NLP learning systems. It clearly deserves further attention.

## Acknowledgments

The work reported in this article was supported by the NINJAL collaborative research project ‘ Development of all-words WSD systems and construction of a correspondence table between WLSP and IJD by these systems.’

## References

Sudha Bhingardive, Dhirendra Singh, V Redkar Murthy, Hanumant Redkar, and Pushpak Bhattacharyya. 2015. Unsupervised Most Frequent Sense Detection using Word Embeddings. In *HLT-NAACL-2015*, pages 1238–1243.

Jordan L Boyd-Graber, David M Blei, and Xiaojin Zhu. 2007. A topic model for word sense disambiguation. In *EMNLP-CoNLL-2007*, pages 1024–1033.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A Unified Model for Word Sense Representation and Disambiguation. In *EMNLP-2014*, pages 1025–1035.

Jun Hatori, Yusuke Miyao, and Jun’ichi Tsujii. 2008. Word Sense Disambiguation for All Words using Tree-Structured Conditional Random Fields. In *COLING-2008*, pages 43–46.

Rubén Izquierdo-Beviá, Lorenza Moreno-Monteagudo, Borja Navarro, and Armando Suárez. 2006. Spanish all-words semantic class disambiguation using Cast3LB corpus. In *MICAI 2006: Advances in Artificial Intelligence*, pages 879–888.

Kiyoaki Shirai. 2001. SENSEVAL-2 Japanese Dictionary Task. In *SENSEVAL-2*, pages 33–36.

Kanako Komiya, Yuto Sasaki, Hajime Morita, Hiroyuki Shinnou, Minoru Sasaki, and Yoshiyuki Kotani. 2015. Surrounding Word Sense Model for Japanese All-words Word Sense Disambiguation. In *PACLIC-29*, pages 35–43.

Anup Kulkarni, Mitesh M Khapra, Saurabh Sohoney, and Pushpak Bhattacharyya. 2010. CFILT: Resource conscious approaches for all-words domain specific WSD. In *SemEval-2010*, pages 421–426.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *the 5th annual international conference on Systems documentation*, pages 24–26.

Kikuo Maekawa. 2007. Design of a Balanced Corpus of Contemporary Written Japanese. In *Symposium on Large-Scale Knowledge Resources (LKR2007)*, pages 55–58.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. SemEval-2007 Task 07: Coarse-grained English All-words Task. In *SemEval-2007*, pages 30–35.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. In *EMNLP-2014*, pages 1059–1069.

Graham Neubig and Shinsuke Mori. 2010. Word-based Partial Annotation for Efficient Corpus Construction. In *LREC-2010*.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis. In *ACL-HLT-2011*, pages 529–533.

Tetsuro Sasada, Shinsuke Mori, Tatsuya Kawahara, and Yoko Yamakata. 2015. Named Entity Recognizer Trainable from Partially Annotated Data. In *PACLING-2015*.

Koichi Tanigaki, Mitsuteru Shiba, Tatsuji Munaka, and Yoshinori Sagisaka. 2013. Density maximization in context-sense metric space for all-words wsd. In *ACL-2013*, pages 884–893.

Koichi Tanigaki, Shuichi Tokumoto, Tatsuji Munaka, and Yoshinori Sagisaka. 2015. Hierarchical bayesian

word sense disambiguation for mapping context space to sense space (in japanese). In *IPSJ SIG on NLP*, pages NL–220–5.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL-95*, pages 189–196.

Zhi Zhong and Hwee Tou Ng. 2009. Word sense disambiguation for all words without hard labor. In *IJCAI-2009*, pages 1616–1622.