ترجمه فا
TarjomeFa.Com

ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

# A Generalized Method for Word Sense Disambiguation based on Wikipedia

Chenliang Li, Aixin Sun, and Anwitaman Datta

School of Computer Engineering,
Nanyang Technological University, Singapore
{lich0020|axsun|anwitaman}@ntu.edu.sg

**Abstract.** In this paper we propose a general framework for word sense disambiguation using knowledge latent in Wikipedia. Specifically, we exploit the rich and growing Wikipedia corpus in order to achieve a large and robust knowledge repository consisting of keyphrases and their associated candidate topics. Keyphrases are mainly derived from Wikipedia article titles and anchor texts associated with wikilinks. The disambiguation of a given keyphrase is based on both the commonness of a candidate topic and the context-dependent relatedness where unnecessary (and potentially noisy) context information is pruned. With extensive experimental evaluations using different relatedness measures, we show that the proposed technique achieved comparable disambiguation accuracies with respect to state-of-the-art techniques, while incurring orders of magnitude less computation cost.

**Keywords:** Word Sense Disambiguation, Wikipedia, Context Pruning

## 1 Introduction

Word sense disambiguation (WSD) is the problem of identifying the sense (meaning) of a word within a specific context. In our daily life, our brain subconsciously relates an ambiguous word to an appropriate meaning based on the context it appears. In natural language processing, word sense disambiguation is thus the task of automatically determining the meaning of a word by considering the associated context(s). It is a complicated but crucial task in many areas such as topic detection and indexing [7, 13], cross-document co-referencing [2, 18], and web people search [1, 12, 22]. Given the current explosive growth of online information and content, an efficient and high-quality disambiguation method with high scalability is of vital importance.

Two main approaches can be found in the literature that try to address the issue, namely *knowledge-based* methods and *supervised machine learning* methods. The former relies primarily on dictionaries, thesauri, or lexical knowledge bases, e.g., a sense inventory consisting of words/phrases and definitions of their possible senses. The Lesk algorithm [11] is the seminal algorithm of such kind, with the assumption that the words referring to the same meaning share a common topic in their neighborhood. Following this idea, a lot of works attempted

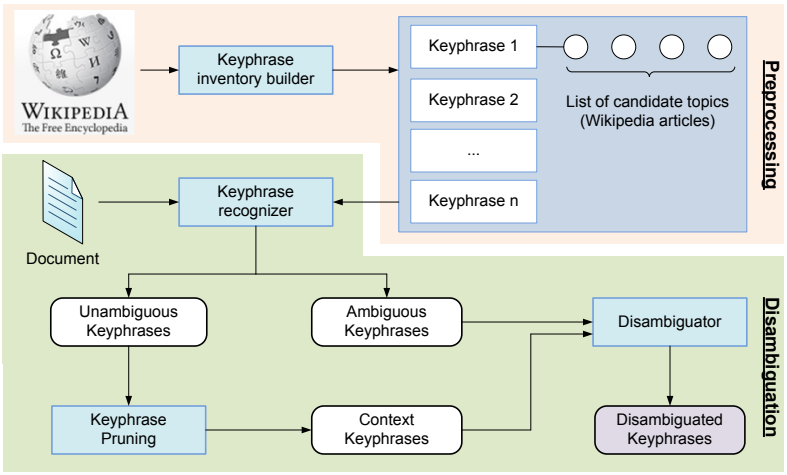


**Fig. 1.** The framework of keyphrase disambiguation based on Wikipedia.

to identify the correct meaning for a word by maximizing the agreement between the dictionary definitions and the contextual terms of the given ambiguous word. Within the disambiguation process, a high-quality sense inventory is a critical factor that affects the performance. However, building such a large-scale, machine-readable lexical resource is tedious and laborious. Thus, the knowledge acquisition bottleneck is the main problem limiting the performance of such systems. The second method based on supervised machine learning attempts to derive a set of local and global contextual features from a manually sense-tagged dataset and to integrate these training examples into a machine learning classifier. Many machine learning techniques have been applied to WSD, and shown to be successful [6, 10, 17]. Nevertheless, machine learning methods too suffer from the knowledge acquisition bottleneck since they require substantial amounts of training examples.

In this paper, we propose a generalized method exploring the use of Wikipedia as the lexical resource for disambiguation. Wikipedia is the largest online encyclopedia and collaborative knowledge repository in the world with over 3.2M articles in English alone. It provides with a reasonably broad if not exhaustive coverage of topics, in comparison to many other knowledge bases. Previous study has found that the quality of Wikipedia articles is comparable to the editor-based encyclopedia [5]. Because of its massive scale of collaboration as well as usage, Wikipedia has become a fruitful resource in many research areas in recent years.

The proposed disambiguation framework is illustrated in Figure 1. Three key components, Wikipedia inventory, keyphrase identification and pruning, and sense disambiguator are developed in our work for disambiguation. Specifically, we build a word sense inventory by extracting the polysemy, synonym and hyperlinks encoded in Wikipedia. Each entry in the inventory is a keyphrase which refers to at least one Wikipedia article. To be detailed in Section 3.1, a keyphrase

is either a Wikipedia article title, or the surface form (or anchor text) of a wikilink. Those keyphrases, each of which refers to exactly one Wikipedia article, are unambiguous keyphrases. Some keyphrases are ambiguous; each of which refers to multiple Wikipedia articles (i.e., candidate topics/senses, shown in Figure 1). Given a document, the unambiguous keyphrases recognized from the document serve as context information to disambiguate the ambiguous keyphrases. In between, the keyphrase pruning helps identify the most important keyphrases in the context of the occurrence of the given ambiguous keyphrase for disambiguation, and it can largely filter out the noise and improve efficiency of the system. The disambiguator is the core component of our framework. It aims to balance the agreement between the context of the ambiguous keyphrase and the context of each candidate sense. Empirical evaluations based on a ground-truth dataset illustrate that our method outperforms other state-of-the-art approaches in terms of both effectiveness and efficiency. Moreover, since the Wikipedia inventory we create relies on the rich semantic information contained in Wikipedia, our approach avoids the traditional knowledge acquisition bottleneck and is applicable to any domain of varying size. It can be plugged into the existing works which require to address word sense disambiguation as well as potential applications.

Our approach is general enough in several senses: given rather exhaustive coverage of Wikipedia topics, the Wikipedia inventory is domain independent; given Wikipedia's growing popularity in other languages, our approach can be readily reused across different languages; and finally, the modular framework allows for using different relatedness measures suiting different application needs.

The rest of this paper is structured as follows: Section 2 reviews related works. Section 3 introduces our approach along with the individual components in the proposed framework. In Section 4, we present and discuss the experimental results. Finally, we conclude in Section 5.

## 2 Related Work

Many recent works explore Wikipedia to enhance text mining tasks, such as semantic relatedness measure [15, 19], text classification and clustering [4, 9, 21], and topic detection [7, 13, 14, 16]. Among these studies, we review the related works involving word sense disambiguation and semantic relatedness measures.

Strube and Ponzetto used Wikipedia for measuring semantic relatedness [19]. Their method searches the Wikipedia articles that contain the specific word in their titles, and measure the relatedness by taking the path length measure in the Wikipedia category hierarchy, text overlap, as well as their probability of occurrence. Milne and Witten developed a light-weight measure of semantic relatedness based on the Wikipedia links, called Wikipedia Link-based Measure (WLM) [15]. First, they identified the Wikipedia articles that related to the term; then, they compute the relatedness of two terms by their mapped Wikipedia articles as follow:

$$relatedness(a, b) = \frac{\log(\max(|A|, |B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|, |B|))} \quad (1)$$

where $a$ and $b$ are the two Wikipedia articles, $A$ and $B$ are the sets of all Wikipedia articles that link to $a$ and $b$ respectively, and $W$ is the set of the entire Wikipedia articles. Due to its high accuracy and low cost, it is commonly used in existing works [8, 13, 16]. In our work, we employ *WLM* as an option to calculate the semantic relatedness between two Wikipedia articles efficiently. Since this method focuses on the hyperlinks within Wikipedia articles, we also investigate Dice and Jaccard measures over the hyperlinks for disambiguation in our study.

Wikify! [14] tries to annotate keyphrases in a document with Wikipedia topics where keyphrase disambiguation is a key step. Both knowledge-based and data-driven algorithms were used in Wikify!. The knowledge-based method, inspired by the Lesk algorithm [11], utilizes the occurrences of ambiguous keyphrases and the contextual information. However, the standalone method performed worse than the baseline method using the most common sense. The data-driven method learns classifier with a number of features, such as part-of-speech and the local contextual words. They then combined the two algorithms by using voting scheme. Most significantly, the method is computationally expensive since it extracts a training feature vector for each ambiguous keyphrase from all its occurrences in the whole Wikipedia.

Medelyan *et al.* [13] utilized both relatedness and commonness measures. For a given document, all keyphrases, each of which uniquely maps to one Wikipedia topic are identified as the context. The context is used to then disambiguate the keyphrases that each can map to more than one Wikipedia topic. In their work, relatedness to the context for each candidate topic of an ambiguous keyphrase is computed by *WLM*. For a candidate topic $t$, *Commonness* for a given keyphrase $k$ is the priori probability of the keyphrase $k$ referring to the candidate topic $t$, i.e., $P(t|k)$ [14]. With the two measures, a score is computed for each candidate topic $t$ for a given keyphrase $k$ using the following equation.

$$Score(t,k) = \frac{\sum_{c \in C} relatedness(t,c)}{|C|} \times P(t|k) \tag{2}$$

In this equation, $C$ denotes the context of the keyphrase $k$. Observe that all context keyphrases in [13] are treated equally. Evaluated on 100 Wikipedia articles, the proposed method outperformed the most common sense baseline by a significant 2.4 percent in F-measure.

Naturally, some keyphrases are more related to the context than others especially when a document covers multiple topics. Milen and Witten [16] proposed to weigh the context keyphrases based on their relatedness to each other as well as their *keyphraseness*. Specifically, if the context is cohesive, then the relatedness measure becomes more relevant; while *commonness* is more useful when the context is diverse. Their empirical study showed that C4.5 classifier achieved the better performance than Medelyan *et al.*'s approach.

While the works from Medelyan *et al.*, Milne and Witten achieve a promising performance among the existing approaches to date, they rely on the context relatedness by taking all unambiguous keyphrases identified in the given document

into account, which is not efficient. As a document often contains some noise, i.e., webpages, not all unambiguous keyphrases are equally useful for expressing the thread of the document, and some of them may even lower accuracy besides wasting computational resources. Although Milen and Witten applied a weighting scheme to highlight the more semantic related context keyphrases, it inevitably incurs additional cost. In this work, we apply a pruning scheme picking the most important keyphases for further processing. This non-trivial step filters out shallow keyphrases and significantly reduces noise, which leads to both better efficiency as well as accuracy. Moreover, existing methods were defined and evaluated by using a specific relatedness measure. Here, we develop a generalized algorithm that can be adaptive to different relatedness measures.

## 3 Disambiguation Framework

In this section, we provide concrete description of the three core components realized in order to achieve disambiguation, namely: *Wikipedia inventory*, *keyphrase identification and pruning*, and *disambiguator*, in that sequence respectively, since it coincides with the order of their usage in our framework.

### 3.1 Wikipedia Inventory

The Wikipedia inventory consists of keyphrases and their associated candidate topics. The keyphrases are from two sources, namely, Wikipedia article titles and anchor texts of wikilinks.

In Wikipedia, each article describes a single topic and is titled using the name which is most commonly used to refer to the topic[1]. Hence, the titles of Wikipedia articles are included in our Wikipedia inventory as keyphrases, each of which refers to the associated Wikipedia article as its candidate topic[2]. Note that, Wikipedia pages for administration or maintenance purposes (e.g., discussion, talk, user pages), are excluded, but the redirect pages are included. A redirect page in Wikipedia redirects the page title to the target article with the preferred title given the two titles referring to the same topic. Such redirection can help us deal with synonym (alternative names), abbreviations, spelling variations, and misspellings. Naturally, target article of the redirection is the candidate topic for the title of a redirect page as a keyphrase in the inventory.

Based on the Wikipedia policy, wikilinks (or hyperlinks) in Wikipedia should be created to relevant topics of the article, technical terms mentioned, or for proper names that are likely to be unfamiliar to readers[3]. Thus, the anchor texts and the linked articles of hyperlinks are semantic associations built by the wisdom of crowd of Wikipedia contributors. Note that, anchor text is the surface form of a hyperlink which may not always match the title of the linked article.

[1] `http://en.wikipedia.org/wiki/Wikipedia:TITLE`

[2] From now on, we use Wikipedia article, candidate sense, sense, candidate topic, Wikipedia topic equivalent interchangeably.

[3] `http://en.wikipedia.org/wiki/Wikipedia:Linking`

Hence the anchor texts enrich the keyphrase inventory largely by polysemy, associative relatedness and social relatedness reflected by them [8]. The anchor text and its linked article is added in our Wikipedia inventory as keyphrase and its candidate topic respectively.

Wikipedia disambiguation pages are designed to disambiguate a number of similar topics which may be referred to by a single ambiguous term. The titles of such pages are normally one of the ambiguous terms, followed by the tag `disambiguation`. The candidate topics are listed in the page, each with a short description about it. We adopt the heuristic by Turdakov and Velikhov [20] to extract the candidate topics from each disambiguation page. When an ambiguous term already exists in the inventory as a keyphrase, we update its list of candidate topics with the ones extracted from corresponding disambiguation page.

In summary, Wikipedia keyphrase inventory is created by taking Wikipedia article titles, processing redirected pages, parsing disambiguation pages and extracting of hyperlinks. In the inventory, if a keyphrase is associated with exactly one topic (or article), we call it unambiguous keyphrase. An ambiguous keyphrase is associated with more than one topic.

### 3.2 Keyphrase Identification and Pruning

We parse the input document and extract all keyphrases that are also present in the inventory, with preference for longer ones. For instance, given a sentence "The Java Sea is ...", we extract a keyphrase *java sea* instead of *java*. For the unambiguous keyphrases extracted, their associated Wikipedia topics are obtained directly from the inventory. These Wikipedia topics help us understand the topics covered by the document, and provide context to determine the sense of the ambiguous keyphrases extracted.

However, a document may cover very diverse topics. Thus, not all identified unambiguous keyphrases are equally important for disambiguation. While the related keyphrases can help identify the correct sense of an ambiguous keyphrase, the unrelated ones may hurt the disambiguation accuracy and incur additional computational cost. This calls for an appropriate pruning scheme for both effectiveness and efficiency.

We use the *keyphraseness* measure to quantify the importance of a keyphrase as in [7, 14]. For a given unambiguous keyphrase, *keyphraseness* is the priori probability that a keyphrase is used as anchor text, no matter where it appears. Based on this measure, we select the top $M$ keyphrases with the highest *keyphraseness* values to form *context keyphrases*. The ambiguous keyphrases identified from the document are then disambiguated using the context keyphrases. In our experiments, we shall evaluate the impact of $M$ on the effectiveness and efficiency of disambiguation.

### 3.3 Disambiguator

For a given ambiguous keyphrase $k$, not all context keyphrases are equally important for disambiguation as some are more semantically related to $k$ than

others. For example, keyphrase *Albert Einstein* appears in the Wikipedia article *Google Search*[4] as an example for the introduction of Google Doodle feature. Obviously there is very little relatedness (if any) between the genius in science and the search engine giant. Nevertheless, due to its high keyphraseness value, *Albert Einstein* is often selected as one of the context keyphrases.

Since each keyphrase (or one of its candidate topics) refers to one Wikipedia article, the computation of relatedness between two keyphrases (or candidate topics) can therefore be reduced to the problem of computing relatedness between their associated Wikipedia articles. A few measures have been reported in the literature to measure semantic relatedness between two Wikipedia articles, mainly based on wikilinks, such as Dice, Jaccard, and WLM [15] measures (see Section 2). As a generic framework, our proposed method can use any such measure and in our following discussion we use $Relatedness(k, k')$ to denote the relatedness between two keyphrases $k$ and $k'$ (or candidate topic $t$).

Recall that a document may cover many diverse topics, which is often reflected by its $M$ context phrases. That is, some context phrases from $M$ may not be strongly related to the other context phrases. Similar to that in [16], a context keyphrase is weighted by its relatedness to all other context keyphrases, shown in the following equation. In this equation, $C$ denotes the set of context keyphrases and $|C| \leq M$.

$$Weight(k, C) = \frac{\sum_{k' \in C \setminus k} Relatedness(k, k')}{|C| - 1} \tag{3}$$

With the defined weight, the relatedness between a candidate topic $t$ to the entire context $C$ is computed in Equation 4. Similar contextual similarity has been adopted in [2, 3, 11, 14].

$$Relatedness(t, C) = \frac{\sum_{k \in C} Weight(k, C) \times Relatedness(t, k)}{\sum_{k \in C} Weight(k, C)} \tag{4}$$

Discussed in Section 2, *commonness* is the priori probability of a keyphrase referring to a specific topic. Existing works already show the effectiveness of commonness measure. In our framework, we balance the relatedness and commonness using an exponential factor $c$. Given a keyphrase $k$ to be disambiguated, let $C_k$ be the set of candidate topics of $k$. We assign topic $t_o$ as the disambiguated topic to $k$ which maximizes both relatedness and commonness with the pre-specified parameter $c$, shown in Equation 5.

$$t_o = \arg\max_{t \in C_k} \left( Relatedness(t, C)^c \times P(t|k) \right) \tag{5}$$

Thus, our framework involves two parameters: $M$ for the size of the context, and $c$ for balancing the relatedness and commonness. A smaller $M$ keeps the more useful topics for disambiguation and improves the efficiency, with the risk of filtering away helpful topics as well. A larger $M$, on the other hand, may bring

[4] `http://en.wikipedia.org/wiki/Google_Search`

in more useful topics as well as noise, and certainly is more computationally costly. As for the scaling factor $c$, it gives the flexibility of adjusting the impact of relatedness measure based on various relatedness definitions (e.g., Jaccard and WLM). In the following section, we illustrate the impact of the two parameters empirically.

## 4 Experiments

We conducted two sets of experiments. In the first, we evaluate the disambiguation accuracy of the proposed technique and the impact of varying the two parameters $M$ and $c$ on the three types of relatedness measures, namely, *Dice*, *Jaccard* and *WLM*. In the second set of experiments, we compare the proposed technique with three state-of-the-art methods and two baseline methods. Next we report our findings.

### 4.1 Dataset and Performance Metric

We used the English Wikipedia dump released on 30 January, 2010[5] to build the keyphrase inventory. In this dump, there are 3,246,821 articles and 266,625,017 hyperlinks among them. The resulting inventory consists of 6,168,269 unambiguous keyphrases and 526,081 ambiguous keyphrases respectively. For the latter, each keyphrase refers to 4.22 candidate topics on an average.

All evaluated disambiguation method assigns each ambiguous keyphrase $p$ to exactly one candidate topic $t$. We report the *accuracy* of the assignments, i.e., the ratio of the correct assignments for all ambiguous keyphrases involved in the evaluation[6]. The correct assignments are predetermined by human annotations (wikilinks which have been made collaboratively) in our experiments.

### 4.2 Evaluation of the Proposed Method

To evaluate the disambiguation accuracy of the proposed method and the impact of the parameter settings, we randomly selected 500 articles from the Wikipedia dump, such that, each selected article contained at least 50 unambiguous keyphrases. Such a selection criterion allowed us to evaluate a relatively large range of $M$ values. Recall that our proposed method involves two parameters $M$ and $c$, and a relatedness measure. $M$ determines the number of related keyphrases involved in the computation and $c$ balances the commonness and relatedness measure.

The selected 500 articles contained $15,298$ ambiguous keyphrases in total. Figure 2 reports the disambiguation accuracy of the proposed methods by varying $M$ and $c$ on the three relatedness measures. $M$ was varied from 5 to 50 with

[5] `http://download.wikimedia.org/enwiki/20100130/`.

[6] Note that as each ambiguous keyphrase cannot have more than one sense in a given context, the accuracy reported here is the same as both precision and recall.

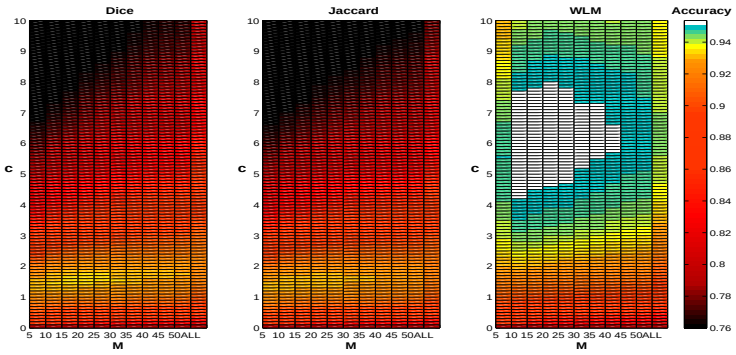


**Fig. 2.** Accuracy of varying $M$ and $c$ with Dice, Jaccard and WLM

a step of 5, and *All* took all unambiguous keyphrases into account. $c$ was varied from 0 to 10 with a step of 0.1. Note that when $c = 0$, our method reduces to the 'most common sense' scenario. We made the following observations on the experimental results.

- Parameter $c$ significantly affected the results for all relatedness measures. For Dice and Jaccard, best accuracies were achieved when $c = 1.5$ for a fairly large span of $M$ values from 10 to 40. For WLM, the best accuracies were achieved when $c$ was in the range of 5 to 7 and $M$ was between 10 and 40.
- A larger $M$ did not necessarily lead to better accuracy. In particular, accuracies dropped for all settings when $M = All$. This is consistent with what we have discussed earlier, that not all unambiguous keyphrases are useful for disambiguation. Many of them may bring in more noise than benefit. The other implication of obtaining high accuracy for relatively small values of $M$ is that, even very few unambiguous keyphrases provide adequate clues for disambiguation.

To better understand the impact of $c$ on the three relatedness measures, as a case study, we calculated the pair-wise relatedness between the Wikipedia article *Google* and all its 235 out-going neighbors, using Dice, Jaccard and WLM respectively. Table 1 reports the mean, standard deviation (std) and coefficient of variation (CV) of these 235 pair-wise relatedness. Observe that the relatedness values by Dice and Jaccard are widely scattered; while WLM generates a narrow dispersion of relatedness values. This is consistent with the previous observation that a larger $c$ obtains a better disambiguation ability with WLM. The experimental results and the case study also illustrate that our method can generalize well for different settings.

### 4.3 Comparison with Other Methods

In this set of experiments, we compare our method with three state-of-the-art methods and two baseline methods for both effectiveness and efficiency. Specifi-

**Table 1.** Relatedness distribution using Dice, Jaccard and WLM

| Relatedness | Mean | Std | CV |
|---|---|---|---|
| Dice | 0.0158 | 0.0201 | 1.2709 |
| Jaccard | 0.0081 | 0.0106 | 1.3074 |
| WLM | 0.4174 | 0.1583 | 0.3793 |

**Table 2.** Statistics on datasets

| Dataset | #articles | #unambiguous | #ambiguous | #candidates |
|---|---|---|---|---|
| Training | 500 | 59,027 | 15,298 | 707,016 |
| Validation | 100 | 13,442 | 3,800 | 178,306 |
| Evaluation | 200 | 24,872 | 7,614 | 354,592 |

cally, we compared our method with the methods reported in Milen and Witten (M&W)[7] [16], and Medelyan *et al.* [13]. The former builds machine learning classifiers to disambiguate the keyphrases and the latter maximizes the balance between commonness and relatedness using equal weight (See Section 2). To build the classifiers, we used C4.5 and Bagged C4.5 using Weka library[8]. The two baseline methods are *Random sense* and *Most common sense* which simply assign topics to ambiguous keyphrases randomly and to the most common sense respectively.

We used the dataset of 500 articles that was used in the previous section (Section 4.2) for classifier training. The trained classifiers are validated using another set of randomly selected 100 articles. For a fair comparison, all methods were evaluated on another set of 200 randomly selected articles which has no overlap with the articles used in training, validation. The statistics of the three datasets are reported in Table 2.

The disambiguation accuracy and execution time of the evaluated methods are reported in Table 3. Note that, for *Random sense*, the result is averaged over 10 runs. For the proposed method, we report the performance using 9 sets of parameter settings on relatedness measure, $M$ and $c$, respectively. The parameters were set according to the findings in Section 4.2.

**Effectiveness**. Overall, the proposed method with WLM achieved the best performance among all methods. Specifically, the best accuracy 94.19% was achieved with *WLM* and $M = 15$, $c = 6.0$. The symbol $^*$ indicates the change is significant according to the paired $t$-test at the level of $p < 0.001$, compared to the best accuracy. The methods with Dice and Jaccard yield competitive accuracies. M&W with C4.5 classifier performed marginally better than Medelyan *et al.*. While classifier bagging improved the accuracy by 0.3% in [16], it degraded the performance by 0.66% in our experiments. All these methods, on the other hand, significantly outperformed the two baselines. Specifically, most common

[7] Two classifiers with the best performance in their work are evaluated here: C4.5 and bagged C4.5.

[8] `http://www.cs.waikato.ac.nz/ml/weka/`

**Table 3.** Disambiguation accuracy and execution time on the evaluation set

| Method | Accuracy(%) | Time(second) |
|---|---|---|
| Random sense | 18.34* | 14 |
| Most common sense | 78.28* | 42 |
| Medelyan *et al.* | 86.07* | 5,438 |
| M&W with C4.5 | 86.25* | 5,810 |
| M&W with bagged C4.5 | 85.59* | 5,877 |
| Dice(M=5,c=1.5) | 92.01* | 137 |
| Dice(M=10,c=1.5) | 92.65* | 265 |
| Dice(M=15,c=1.5) | 92.50* | 392 |
| Jaccard(M=5,c=1.5) | 91.99* | 136 |
| Jaccard(M=10,c=1.5) | 92.58* | 266 |
| Jaccard(M=15,c=1.5) | 92.46* | 392 |
| WLM(M=5,c=6.0) | 93.63* | 140 |
| WLM(M=10,c=6.0) | 94.17 | 273 |
| WLM(M=15,c=6.0) | **94.19** | 399 |

sense delivered an accuracy of 78.28%, and random guess had a mere 18.34% accuracy.

**Efficiency**. Table 3 also reports the execution time by each method evaluated, ignoring the time taken for data loading and classifier training. All experiments were conducted on the same workstation with a 2.40GHz Xeon quad-core CPU and 24GB of RAM. Observe that our method outperformed the state-of-the-art methods significantly in terms of efficiency. With $M = 15$ and 5, our method was 14 and 40 times faster than Medelyan *et al.* and M&W, respectively. Moreover, by setting $M$ to 5 instead of 15, the proposed method speed up 2.8 times with less than 1% of drop in accuracy, for all three relatedness settings.

## 5 Conclusion

Word sense disambiguation is a key problem to address in many applications in the areas of Natural Language Processing, Information Retrieval and others. The large scale and high quality knowledge in Wikipedia enables a domain independent knowledge repository for word sense disambiguation. In this paper, we propose a general framework (which can accommodate diverse relatedness measures, is domain independent, and potentially can be applied for other languages) to utilize Wikipedia for word/keyphrase sense disambiguation using both commonness and relatedness measures. We show that pruning of unnecessary or potentially noisy context make the disambiguation process orders of magnitude faster than existing methods while achieving comparable (if not better) disambiguation accuracy.

**Acknowledgments.** This work is supported in part by the Agency for Science, Technology and Research (A*STAR) SERC Grant No: 072 134 0055.

## References


1. J. Artiles, J. Gonzalo, and S. Sekine. Weps 2 evaluation campaign: overview of the web people search clustering task. In *Web People Search Evaluation Workshop (WePS), WWW Conference*, 2009.
2. A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Int'l Conf. on Computational Linguistics*, pages 79–85, 1998.
3. S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, pages 708–716, 2007.
4. E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge. In *AAAI*, pages 1301–1306, 2006.
5. J. Giles. Internet encyclopaedias go head to head. *Nature*, 438, Dec 2005.
6. A. Gliozzo, C. Giuliano, and C. Strapparava. Domain kernels for word sense disambiguation. In *ACL*, pages 403–410, 2005.
7. M. Grineva, M. Grinev, and D. Lizorkin. Extracting key terms from noisy and multitheme documents. In *WWW*, pages 661–670, 2009.
8. X. Han and J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *ACM CIKM*, pages 215–224, 2009.
9. X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou. Exploiting wikipedia as external knowledge for document clustering. In *ACM KDD*, pages 389–396, 2009.
10. Y. K. Lee and H. T. Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *EMNLP*, pages 41–48, 2002.
11. M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC*, pages 24–26, 1986.
12. G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *HLT-NAACL*, pages 33–40, 2003.
13. O. Medelyan, I. H. Witten, and D. Milne. Topic indexing with wikipedia. In *AAAI Workshop on Wikipedia and Artificial Intelligence*, 2008.
14. R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *ACM CIKM*, pages 233–242, 2007.
15. D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *AAAI Workshop on Wikipedia and Artificial Intelligence*, 2008.
16. D. Milne and I. H. Witten. Learning to link with wikipedia. In *ACM CIKM*, pages 509–518, 2008.
17. T. Pedersen. A decision tree of bigrams is an accurate predictor of word sense. In *NAACL*, pages 1–8, 2001.
18. Y. Ravin and Z. Kazi. Is hillary rodham clinton the president?: disambiguating names across documents. In *Workshop on Coreference and its Applications (CorefApp)*, pages 9–16, 1999.
19. M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, pages 1419–1424, 2006.
20. D. Turdakov and P. Velikhov. Semantic relatedness metric for wikipedia concepts based on link analysis and its application to word sense disambiguation. In *SYRCoDIS*, volume 355 of *CEUR Workshop Proceedings*, 2008.
21. P. Wang and C. Domeniconi. Building semantic kernels for text classification using wikipedia. In *ACM KDD*, pages 713–721, 2008.
22. M. Yoshida, M. Ikeda, S. Ono, I. Sato, and H. Nakagawa. Person name disambiguation by bootstrapping. In *ACM SIGIR*, pages 10–17, 2010.

ترجمه فا
TarjomeFa.Com

این مقاله، از سری مقالات ترجمه شده رایگان سایت ترجمه فا میباشد که با فرمت PDF در اختیار شما عزیزان قرار گرفته است. در صورت تمایل میتوانید با کلیک بر روی دکمه های زیر از سایر مقالات نیز استفاده نمایید:

✔ لیست مقالات ترجمه شده

✔ لیست مقالات ترجمه شده رایگان

✔ لیست جدیدترین مقالات انگلیسی ISI

سایت ترجمه فا ؛ مرجع جدیدترین مقالات ترجمه شده از نشریات معتبر خارجی