TSDW: Two-Stage Word Sense Disambiguation Using Wikipedia

Chenliang Li, Aixin Sun, and Anwitaman Datta
School of Computer Engineering, Block N4, Nanyang Technological University, Nanyang Avenue, Singapore 639798. E-mail: {lich0020, axsun, anwitaman}@ntu.edu.sg

The semantic knowledge of Wikipedia has proved to be useful for many tasks, for example, named entity disambiguation. Among these applications, the task of identifying the word sense based on Wikipedia is a crucial component because the output of this component is often used in subsequent tasks. In this article, we present a two-stage framework (called TSDW) for word sense disambiguation using knowledge latent in Wikipedia. The disambiguation of a given phrase is applied through a two-stage disambiguation process: (a) The first-stage disambiguation explores the contextual semantic information, where the noisy information is pruned for better effectiveness and efficiency; and (b) the second-stage disambiguation explores the disambiguated phrases of high confidence from the first stage to achieve better redisambiguation decisions for the phrases that are difficult to disambiguate in the first stage. Moreover, existing studies have addressed the disambiguation problem for English text only. Considering the popular usage of Wikipedia in different languages, we study the performance of TSDW and the existing state-of-the-art approaches over both English and Traditional Chinese articles. The experimental results show that TSDW generalizes well to different semantic relatedness measures and text in different languages. More important, TSDW significantly outperforms the state-of-the-art approaches with both better effectiveness and efficiency.

Introduction

Being an online collaborative knowledge repository with millions of contributors around the world, Wikipedia has grown into the largest multilingual encyclopedia. More specifically, it contains more than 21 million articles in more than 200 languages.[1] Because of its massive scale of collaboration and usage (Cho, Chen, & Chung, 2010; Stvilia, Twidale, Smith, & Gasser, 2008), as well as the high quality of its articles (Giles, 2005), Wikipedia has become a credible resource in many research fields. Particularly, Wikipedia's categorization scheme and hyperlink (wikilink) have been extensively studied in text mining, such as document clustering and classification (Gabrilovich & Markovitch, 2006; Hu, Zhang, Lu, Park, & Zhou, 2009; Malo, Sinha, Wallenius, & Korhonen, 2011; Wang & Domeniconi, 2008), semantic relatedness (Gabrilovich & Markovitch, 2007; Milne & Witten, 2008a; Strube & Ponzetto, 2006; Yeh, Ramage, Manning, Agirre, & Soroa, 2009), topic detection and indexing (Grineva, Grinev, & Lizorkin, 2009; Medelyan, Witten, & Milne, 2008), and named entity disambiguation (Bunescu & Pasca, 2006; Cucerzan, 2007; Han & Zhao, 2009), among others.

In this article, we are interested in the problem of *word sense disambiguation to Wikipedia*, or simply *disambiguation to Wikipedia* (*D2W*). Formally defined by Ratinov, Roth, Downey, and Anderson (2011), the task of D2W is to disambiguate a set of explicitly identified substrings (e.g., words or phrases) in a given document by mapping each substring to a Wikipedia page,[2] if one exists. For simplicity, we refer the explicitly identified substrings as phrases,[3] and we focus only on the phrases that each has at least one corresponding Wikipedia page. If a phrase maps to exactly one Wikipedia page, the mapping process is straightforward because the phrase is unambiguous. A phrase is ambiguous if it can be mapped to more than one Wikipedia page. The task of D2W is to disambiguate the *ambiguous phrases* in a given document to their correct Wikipedia topics based on the content of the document. The D2W problem is the fundamental problem of the Wikification task, which links words or phrases in text documents to their corresponding Wikipedia pages (Mihalcea & Csomai, 2007). D2W is also similar to the word sense disambiguation task in natural language

[1] http://meta.wikimedia.org/wiki/List_of_Wikipedias

Received April 28, 2012; revised August 10, 2012, and September 6, 2012; accepted September 6, 2012

 • Published online 12 April 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.22829

[2] Because each Wikipedia article introduces a single topic, in our discussion, we use Wikipedia article, Wikipedia page, candidate sense, sense, candidate topic, and Wikipedia topic equivalent interchangeably.

[3] We do not specifically distinguish words and phrases in our discussion.

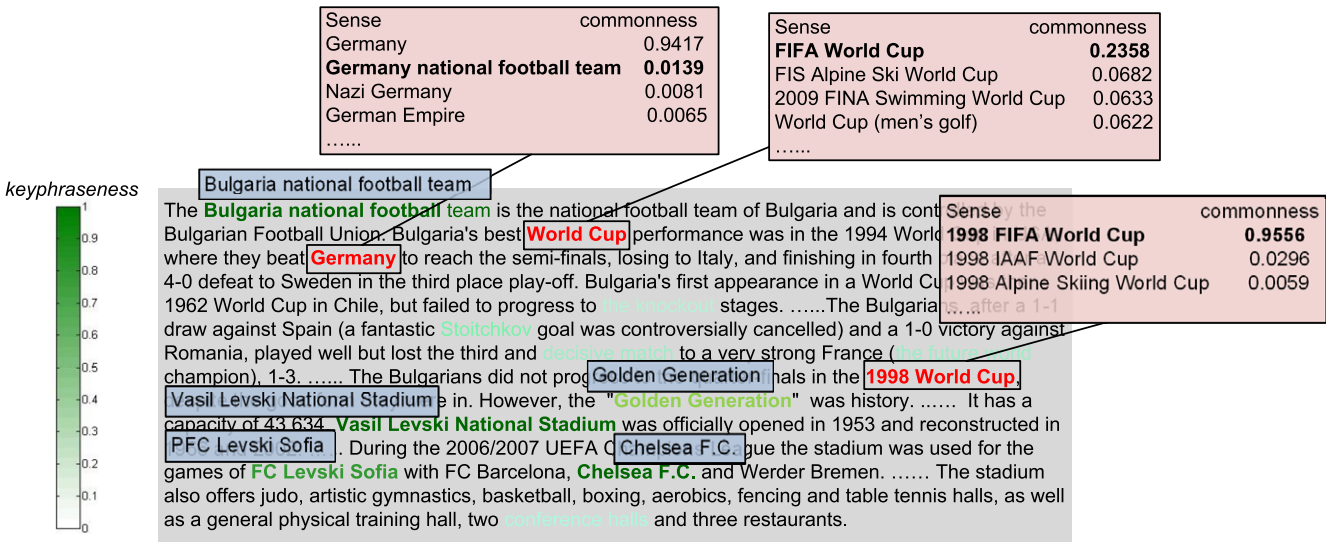

FIG. 1. Sample article with phrases disambiguated to Wikipedia topics. Ambiguous phrases are highlighted in red and boldface. The top 10 unambiguous phrases in terms of *keyphraseness* are highlighted in gradual changing green colors. The top five unambiguous phrases are highlighted in boldface with their corresponding Wikipedia topics labeled. The most probable candidate topics are listed for the ambiguous phrases along with the corresponding *commonness* values, and their correct Wikipedia topics are highlighted in boldface. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

processing, which can be an important preprocessing step in the aforementioned text mining tasks. Figure 1 shows an example of an article in our evaluation data set, where ambiguous phrases are highlighted in red and boldface, and disambiguated using our proposed approach.

Recently, many works have been proposed to address the problem of D2W (Li, Sun, & Datta, 2011; Medelyan et al., 2008; Milne & Witten, 2008b; Ratinov et al., 2011). The proposed solutions mainly focus on the approximation of the likelihood of a phrase mapping to each candidate Wikipedia page based on the semantic context of the document. The semantic context is defined to be the set of Wikipedia pages that are uniquely mapped to by the unambiguous phrases in the document (Li et al., 2011; Medelyan et al., 2008; Milne & Witten, 2008b). However, the information provided by the unambiguous phrases in a document could be very limited, leading to poor accuracy for disambiguation of some ambiguous phrases. The situation becomes apparent when short documents are processed. To augment the semantic context of such documents, Wikipedia pages of ambiguous phrases disambiguated by some simple methods have been used as additional context information to augment the semantic context of such documents. One simple method is to map an ambiguous phrase to the Wikipedia page that has the largest cosine similarity with the local neighboring words of the ambiguous phrase (Ratinov et al., 2011). However, the augmentation may bring in not only additional computational cost but also noisy information. In addition, all existing works are based on English articles only. Considering the popular usage of Wikipedia in other languages, as well as its significant contribution in many fields (Chrupala & Klakow, 2010; Kassner, Nastase, & Strube, 2008; Li, Huang, Tsuchiya, Ren, & Zhong, 2008; Shirakawa, Nakayama, Aramaki, Hara, & Nishio, 2010; Tamagawa et al., 2010; Zesch, Gurevych, & Mühlhäuser, 2007), it is important to study the performance of the disambiguation solutions to other languages in terms of both effectiveness and efficiency.

In this article, we propose a generic two-stage framework for word sense disambiguation to Wikipedia, named TSDW. TSDW consists of the following three key components: *Wikipedia inventory*, *keyphrase recognizer*, and *two-stage disambiguator*. We build a word sense inventory by extracting the polysemy, synonym, and hyperlinks encoded in Wikipedia. Each entry in the inventory is a keyphrase that refers to at least one Wikipedia article. A keyphrase is a phrase that is used either as a Wikipedia article title or anchor text of a wikilink in Wikipedia. The 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). Given a document, the unambiguous keyphrases extracted by the keyphrase recognizer from the document serve as context information to help in disambiguating the ambiguous keyphrases. Although the ambiguous keyphrases are often ignored in existing works, some of them may provide additional semantic clues, resulting in a better semantic context for disambiguation. The core component of TSDW, which distinguishes this work from our previous work (Li et al., 2011), is the two-stage disambiguator. In the first stage, it disambiguates the ambiguous keyphrases in a document by exploring semantic context defined by the unambiguous keyphrases. The quality of each disambiguation decision in the first stage is evaluated by a confidence measure. In the second stage, the disambiguated keyphrases with high confidence from the first stage are

DOI: 10.1002/asi

recruited as additional semantic context for a better disambiguation of the keyphrases with low confidence. One of the main contributions of this work is, therefore, the confidence measure. Moreover, the recruitment of high-confidence disambiguated keyphrases alleviates the data sparsity problem in the first stage. For example, as demonstrated in Figure 1, the ambiguous keyphrase *Germany* cannot be disambiguated with high confidence, because the context of the five unambiguous keyphrases[4] cannot provide adequate discriminative information. However, the ambiguous keyphrases *World Cup* and *1998 World Cup* can be easily disambiguated correctly with high confidence at the first stage. Based on the additional semantic information offered by these two disambiguations, *Germany* can be correctly disambiguated to Wikipedia topic *Germany national football team* in the second stage. Note that not all unambiguous or high-confidence disambiguated keyphrases are used as semantic context in our disambiguation process because of both effectiveness and efficiency reasons. We further illustrate this point by showing the impact of the size of the semantic context in our experiments.

We highlight that the proposed TSDW framework is generic because it can be materialized by using any semantic relatedness measure. In our experiments, we evaluate the impact of using three semantic relatedness measures including *Jaccard*, *Dice*, and Wikipedia link-based measure (*WLM*). Note that semantic relatedness measures hold varying characteristics for Wikipedia corpora in different languages. In our experiments, we have evaluated our TSDW framework using both English and Chinese versions of Wikipedia. More specifically, our empirical evaluation involves several data sets ranging from Wikipedia articles to newswire reports, and in two different languages, English and Traditional Chinese. Our experimental results show that TSDW outperforms other state-of-the-art approaches in terms of both effectiveness and efficiency, for both English and Traditional Chinese articles. In summary, we made the following contributions in this article:

1. We proposed a generic TSDW framework. TSDW is able to accommodate different semantic relatedness measures for Wikipedia in different languages, making it suitable for different application settings.
2. TSDW exploits the semantic context of both unambiguous and ambiguous keyphrases with a two-stage disambiguator. The valuable semantic information contained in the ambiguous keyphrases can help alleviate the data sparsity problem encountered by using the unambiguous keyphrases alone.
3. To the best of our knowledge, this work is the first to study the performance of existing state-of-the-art approaches and TSDW using data sets in multiple (two) languages, English and Traditional Chinese, in terms of both effectiveness and efficiency. This has significant implications for any future related works.

The rest of this article is structured as follows: The next section briefly describes related works on word sense disambiguation to Wikipedia. Then we present TSDW in detail. Next, we study the performance of TSDW and its generalization to other languages through extensive experiments. Finally, we conclude this work and discuss the future work.

Related Work

Word sense disambiguation is the task of identifying the sense of a word or phrase within a specific context. Two main approaches can be found in the literature that try to address this problem, namely, *knowledge-based* methods and *supervised machine learning* methods. The former tries to identify the correct sense by maximizing the agreement between the dictionary definition and the context of the given ambiguous word or phrase. The latter identifies the correct sense by applying a classifier trained on a set of local and global contextual features from a manually sense-tagged data set. Both approaches suffer from the knowledge acquisition bottleneck problem: Either a high-quality sense inventory or a substantial number of training examples are required. Because Wikipedia is the largest online encyclopedia and collaborative knowledge repository, it has become a paradise for the researchers in the field of natural language processing. Many studies explore the semantic resource of Wikipedia for word sense disambiguation, because Wikipedia provides both a high-quality sense inventory and a large number of human annotations. In the following subsections, we review the related works in the two directions, respectively.

Knowledge-Based Methods

Mihalcea and Csomai (2007) addressed the problem of word sense D2W in their Wikify! system. Both knowledge-based and supervised machine learning methods were investigated. The knowledge-based method, inspired by the Lesk algorithm (Lesk, 1986), uses the occurrences of ambiguous keyphrases and the contextual information. A Wikipedia topic that has the maximum overlap with the contextual words of the given ambiguous keyphrase is chosen as the correct sense. However, this method alone performed poorer than the baseline method using the most common sense. Medelyan and colleagues (2008) used both *commonness* and *relatedness* measures for disambiguation. For a candidate topic t, commonness for a given keyphrase k is defined as $P(t|k)$, that is, the prior probability of keyphrase k referring to candidate topic t (Mihalcea & Csomai). For a given document, all keyphrases, each of which uniquely maps to one Wikipedia topic, are identified as the context. The context is used then to 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

[4]The five unambiguous keyphrases with the highest *keyphraseness* values: *Bulgaria national football team*, *Vasil Levski National Stadium*, *Chelsea F.C.*, *FC Levski Sofia*, and *Golden Generation*.

DOI: 10.1002/asi

keyphrase is computed by using WLM (Milne & Witten, 2008a). A score is computed for each candidate topic t of a given keyphrase k to be the multiplication of commonness and relatedness. The topic with the highest score is chosen to be the disambiguated sense. Their approach significantly outperforms the most common sense baseline.

Recently, Ratinov et al. (2011) addressed the problem of D2W by combining both local and global approaches with supervised learning. In detail, the local context approach solves the disambiguation by choosing the topic that is the most similar to the input document containing the ambiguous keyphrase. The global approach, then, solves the problem by disambiguating the ambiguous keyphrases as being a coherent set of related topics. They implemented the local approach by using cosine similarity between the candidate topic and the context window of an ambiguous keyphrase. The global approach was implemented by measuring the relatedness between two Wikipedia topics using Wikipedia links (i.e., WLM and Pointwise Mutual Information [*PMI*]). The set of topics that is to be optimized as a coherent set is augmented by taking all topics of the ambiguous keyphrases that have been disambiguated by the local approach. It is reported that combining both local and global context approaches results in better disambiguation accuracy.

Our previous work (Li et al., 2011) on D2W filters away noisy contextual information and applies a scaling factor to accommodate the relatedness measures with varying property (i.e., the dispersion of relatedness measure). The computation required is greatly reduced because of the context pruning. Meanwhile, because noisy information is filtered away, the accuracy of disambiguation is improved as well. The experimental results showed that both better effectiveness and efficiency were achieved compared with the state-of-the-art approaches. Moreover, the scaling factor added to the relatedness measure enables the approach to generalize well to different settings.

Supervised Machine Learning Methods

Mihalcea and Csomai (2007) also studied the supervised machine learning method in their Wikify! system. The classifier is learned with a number of contextual features, such as part-of-speech tag and local neighboring words. Milne and Witten (2008b) further extended this method by considering the cohesiveness of the context (Milne & Witten, 2008b). Several machine learning-based classifiers are trained based on the relatedness to the context, the cohesiveness of the context, and commonness. Whereas Medelyan and colleagues (2008) measured the relatedness to the context by computing the averaged relatedness to all unambiguous keyphrases, Milne and Witten weighted each unambiguous keyphrase based on their relatedness to each other as well as their *keyphraseness*. Keyphraseness is the prior probability that a given keyphrase should be linked in a Wikipedia page. Higher keyphraseness indicates that the keyphrase is a concrete concept of human knowledge. They assumed that if the context is cohesive, the relatedness measure is more important for the disambiguation; otherwise, commonness would be a more significant indicator when the context is diverse. Their empirical study showed that C4.5 classifier achieved better disambiguation accuracy than Medelyan and colleagues' method.

Although the methods by Medelyan and colleagues (2008) and Milne and Witten (2008b) have achieved promising disambiguation accuracy to date, they measure the context relatedness by taking all unambiguous keyphrases identified in the given document into account. Although Milne and Witten applied a weighting scheme to highlight the more semantically related context keyphrases, it inevitably incurs additional computation. Ratinov et al. (2011) explored the context information by applying Named Entity Recognition (NER) taggers and shallow parser, that is, to restrict the context information by using named entities and nouns only. Although their approach reduces computation in the disambiguation process, the additional computation incurred by applying the NER tagger and shallow parser is expensive. Because a document often contains some noise, not all unambiguous keyphrases are equally useful for expressing the main topic of the document. Therefore, some unambiguous keyphrases may even hurt the disambiguation accuracy besides incurring computational resources. Our previous work (Li et al., 2011) applies a pruning scheme picking the most important keyphrases for use in the disambiguation process. This nontrivial step filters away shallow keyphrases and reduces noise in the context. In this article, we extend our previous approach with a two-stage disambiguation process. Specifically, we explore the semantic clue contained in the ambiguous keyphrases in the second-stage disambiguation process, by picking some high-confident disambiguation decisions from the first stage. Although Ratinov et al. adopted the predictions of ambiguous keyphrases from a local context in their global approach, they augmented the topic set with all predictions of all ambiguous keyphrases. This introduces large computational cost. Similar works were also proposed for general word sense disambiguation (Agirre & de Lacalle, 2009; Agirre & Soroa, 2009; Mihalcea, 2005), where the candidate senses of all ambiguous words are considered together. In this work, we augment the context in the second-stage disambiguation by taking only a limited number of disambiguation decisions of high confidence from the first stage.

Note that TSDW can also be configured to have more passes of redisambiguations, such as third-stage disambiguation. The idea of multiple passes of inference to alleviate the data sparsity problem is similar to bootstrapping (Hearst, 1992; Komachi, Kudo, Shimbo, & Matsumoto, 2008; Yoshida, Ikeda, Ono, Sato, & Nakagawa, 2010) and semisupervised learning (Nigam, McCallum, Thrun, & Mitchell, 2000; Su, Shirab, & Matwin, 2011). Nevertheless, in evaluation of TSDW, we show that two-stage disambiguation is adequate for D2W task and further stages yield negligible gain.

DOI: 10.1002/asi

TSDW Disambiguation Framework

The TSDW framework consists of the following three main components: Wikipedia inventory, keyphrase recognizer, and two-stage disambiguator. The first two components are mainly based on our previous work (Li et al., 2011). For completeness, we briefly describe these two components respectively. Then we present the two-stage disambiguator in detail.

Wikipedia Inventory

The Wikipedia inventory is a dictionary that consists of keyphrases and their associated candidate topics based on Wikipedia. Each candidate topic refers to a corresponding Wikipedia article. The sources for the keyphrases and associated candidate topics include: Wikipedia article titles, anchor text of wikilinks, redirect pages, and disambiguation pages in Wikipedia. In the following list, we describe each of the four sources in detail:

- Given a topic described by a Wikipedia article, the article title is chosen to be the one that is most commonly used to refer to that topic.[5] Hence the titles of Wikipedia articles are included in our Wikipedia inventory as keyphrases, each of which contains the associated Wikipedia article as its candidate topic. Note that Wikipedia pages for administration or maintenance purposes (e.g., discussion, talk, user pages) are excluded.
- Based on Wikipedia policy, wikilinks (or hyperlinks) in Wikipedia are created to help readers better understand the topic, by linking technical terms and proper names (i.e., abbreviations, synonym, spelling variations) to the related Wikipedia articles.[6] Thus, including the anchor text and the related Wikipedia articles largely enriches the keyphrase inventory, leading to an inventory of broader coverage.
- A redirect page groups the alternative names of a topic together by establishing redirect relations between the alternative names and the Wikipedia article of that topic. For instance, "U.S." is redirected to the Wikipedia article *United States* because "U.S." is an alternative name for the topic *United States*. Such redirections help us further enrich the keyphrase inventory with abbreviations, synonym, and spelling variations.
- Wikipedia disambiguation pages are designed to help readers find the topic of interest among possible candidate topics from the polysemy query. The titles of such pages are normally polysemy keyphrases, followed by 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 (2008) 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 the corresponding disambiguation page.

In summary, the Wikipedia keyphrase inventory is created by taking Wikipedia article titles, processing redirected pages, parsing disambiguation pages, and extracting 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.

Keyphrase Recognizer

Given an input document, all keyphrases that appear in the Wikipedia inventory are extracted from the document with a preference for longer phrases. For example, given a sentence segment, "Java programming language," we extract keyphrase *Java programming language* instead of two keyphrases *Java* and *programming language*. The keyphrases extracted are classified into ambiguous and unambiguous keyphrases based on the Wikipedia inventory. For the unambiguous keyphrases extracted, their associated Wikipedia topics are obtained directly from the inventory. These Wikipedia topics provide us with the semantic clue to the topics covered in the document, and thus help us disambiguate the ambiguous keyphrases.

Given Wikipedia's broad coverage of human knowledge, the size of the Wikipedia inventory is extraordinarily large, with millions of keyphrases. Thus, recognizing matched keyphrases efficiently based on the Wikipedia inventory becomes a nontrivial task. We implement the keyphrase recognizer by using a prefix tree (Knuth, 1998) algorithm over the keyphrases of the same prefix word. This algorithm has a complexity of $O(n)$, where n is the length of the input document in number of words. The detailed description of the algorithm is provided in the appendix.

Two-Stage Disambiguator

We disambiguate an ambiguous keyphrase based on the semantic context to which it is associated. Given an ambiguous keyphrase k and its associated context $\mathcal{C}$, we compute the probability of keyphrase k referring to a candidate topic t, denoted by $P(t \mid k, \mathcal{C})$. The candidate topic with the highest probability is chosen to be the disambiguated sense of the keyphrase, shown in the following equation, where T_k denotes the set of candidate topics for keyphrase k:

$$t_o = \arg\max_{t \in T_k} P(t \mid k, \mathcal{C}) \tag{1}$$

To compute $P(t \mid k, \mathcal{C})$, we mimic a human being's disambiguation process by considering two factors: the prior probability of keyphrase k referring to a candidate topic t, also known as *commonness* (see Related Work), and the likelihood of context $\mathcal{C}$ related to the candidate topic. Assuming that the two factors are independent of each other and are independent for a given topic t, we have Equation 2 to compute $P(t \mid k, \mathcal{C})$. In this equation, $P(t|k)$ is the prior probability of topic t given keyphrase k (i.e., commonness), and $P(t \mid \mathcal{C})$ is the probability of the keyphrase referring to topic t given context $\mathcal{C}$:

[5]http://en.wikipedia.org/wiki/Wikipedia:TITLE
[6]http://en.wikipedia.org/wiki/Wikipedia:Linking

DOI: 10.1002/asi

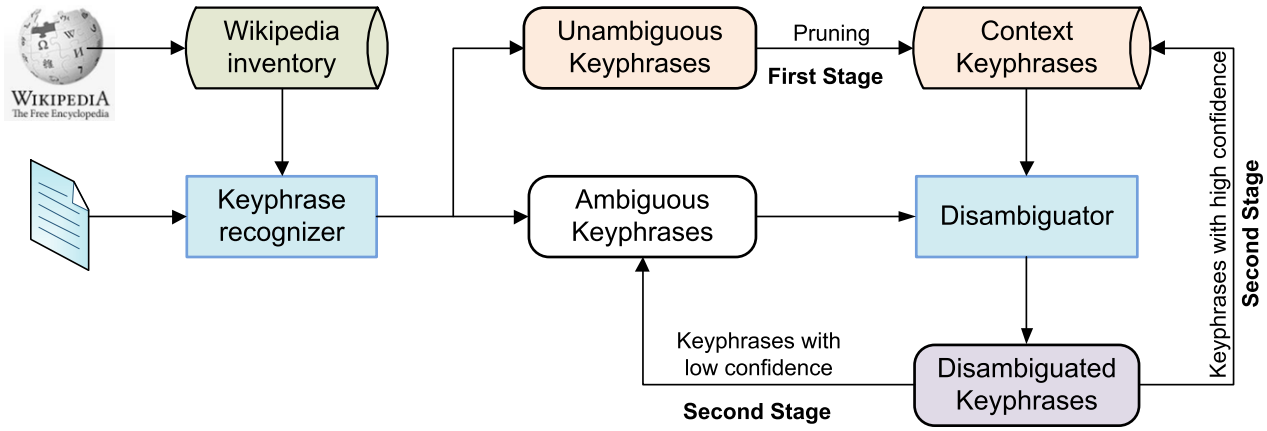

FIG. 2. Two-stage disambiguation process. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$
\begin{aligned}
P(t \mid k, \mathcal{C}) &= \frac{P(k, \mathcal{C} \mid t)P(t)}{P(k, \mathcal{C})} \\
&= \frac{P(k \mid t)P(\mathcal{C} \mid t)P(t)}{P(k, \mathcal{C})} \\
&= \frac{P(k, t)P(\mathcal{C}, t)}{P(t)P(k, \mathcal{C})} \\
&= \frac{P(k, t)P(\mathcal{C}, t)}{P(t)P(k)P(\mathcal{C})} \\
&= \frac{P(t \mid k)P(t \mid \mathcal{C})}{P(t)}
\end{aligned}
\tag{2}
$$

In Equation 2, $P(t)$ is the prior probability of topic t. We further assume a flat prior probability for all topics, that is, $P(t_i) = P(t_j)$. Thus, Equation 2 can then be updated as follows:

$$
P(t \mid k, \mathcal{C}) \propto P(t \mid k)P(t \mid \mathcal{C}) \tag{3}
$$

Because $P(t|k)$ is independent of the context and can be estimated from the Wikipedia data directly, the task of disambiguation is then reduced to estimating $P(t \mid \mathcal{C})$. As discussed in Related Work, the context is usually defined by the set of unambiguous keyphrases in the given document. However, a document may cover many diverse topics. Therefore, not all unambiguous keyphrases from the document are equally important in defining the context for a keyphrase to be disambiguated. Whereas the more related keyphrases help identify the correct sense of an ambiguous keyphrase, the less related ones may hurt the disambiguation accuracy and incur additional computation. The situation would be further exaggerated by the noise contained in Wikipedia. This calls for an appropriate construction of context $\mathcal{C}$ for an ambiguous keyphrase, aiming for both disambiguation effectiveness and efficiency. On the other hand, the discriminative information carried by the related unambiguous keyphrases may be limited. For example, a short text may contain a very small number of unambiguous keyphrases, or the unambiguous keyphrases cover only a subtopic of the text and are not useful for the ambiguous key-phrases about other subtopics. This calls for an approach to enlarging the context for disambiguation by adding additional information on top of the existing unambiguous keyphrases extracted from the document.

To meet the aforementioned two requirements, we propose a *two-stage disambiguator* to exploit the semantic clue provided by both the unambiguous keyphrases and the ambiguous keyphrases. Illustrated in Figure 2, in the first stage, the disambiguator disambiguates the ambiguous keyphrases based on the context formed by unambiguous keyphrases after pruning the less related unambiguous keyphrases. The small number of high-quality unambiguous keyphrases as context address the first requirement. Based on the confidence measure, each disambiguation is classified as disambiguation with high or low confidence. In the second stage, the keyphrases with high-confidence disambiguation are added into the context for the redisambiguation of the keyphrases with low confidence in the first stage by re-estimating $P(t \mid \mathcal{C})$. The use of additional context originated from the ambiguous keyphrases answers the second requirement. We detail the two-stage disambiguator in a subsequent section. Note that the second-stage disambiguation process can be easily repeated for more than one pass leading to a third-stage (or more) redisambiguation. However, we observe in our experiments that more iterations do not necessarily lead to improvement in the disambiguation accuracy.

First-stage disambiguation. In the first stage, we try to approximate the likelihood $P(t \mid \mathcal{C})$ by restricting context $\mathcal{C}$ to be a subset of all unambiguous keyphrases extracted from the input text. Given all unambiguous keyphrases extracted from the input text, not all of them are equally helpful for word sense disambiguation. For instance, the keyphrase *Mr.* is a keyphrase in the Wikipedia inventory. However, it is unlikely that the keyphrase would contribute positively to the estimation of $P(t \mid \mathcal{C})$ for any topic t. We therefore

DOI: 10.1002/asi

construct context $\mathcal{C}$ by applying keyphrase pruning and approximate $P(t \mid \mathcal{C})$ by using weighted relatedness to $\mathcal{C}$.

We use keyphraseness measure to quantify the importance of a keyphrase. For a given unambiguous keyphrase, keyphraseness is the a prior probability that a keyphrase is selected as a link, no matter where it appears in Wikipedia. For example, keyphrase *Conference halls* highlighted in Figure 1 has a very low keyphraseness value, because it is rarely linked by other Wikipedia articles or used as anchor text for wikilinks. Based on this measure, we select the top M_1 keyphrases with the highest keyphraseness values to form context $\mathcal{C}$. The keyphrases in $\mathcal{C}$ are known as *context keyphrases* and $|\mathcal{C}| \leq M_1$.

Recall that a document may cover many diverse topics, which is often reflected by its M_1 context phrases. That is, some context phrases from M_1 may not be strongly related to other context phrases. A context keyphrase is weighted by its relatedness to all other context keyphrases, shown in the following equation:

$$w(k, \mathcal{C}) = \frac{\sum_{k' \in \mathcal{C} \setminus k} r(k, k')}{|\mathcal{C}| - 1} \tag{4}$$

where $r(k, k')$ denotes the relatedness between two Wikipedia articles referred to by k and k', respectively. Because each keyphrase (or one of its candidate topics) refers to one Wikipedia article, the relatedness measure is therefore reduced to the problem of computing the relatedness between their associated Wikipedia articles. A few measures have been reported in the literature to measure the semantic relatedness between two Wikipedia articles, mainly based on wikilinks, such as Dice, Jaccard, and WLM measures, or cosine similarity based on the content of the articles (Ratinov et al., 2011). As a generic framework, TSDW can use any such measure. In the following discussion, we use $r(k, k')$ to denote the relatedness between two keyphrases k and k' (or candidate topic t) computed by any chosen semantic relatedness measure.

With a chosen relatedness measure, the weighted relatedness $r(t, \mathcal{C})$ between a candidate topic t to context $\mathcal{C}$ is computed using Equation 5. The likelihood $P(t \mid \mathcal{C})$ is approximated by using $r(t, \mathcal{C})$ with an exponential scale-up factor c as in Equation 6. Note that the scale-up factor c is specific to the relatedness measure under use, and its value can be learned by applying grid search over the range of [0, 10] (Li et al., 2011).

$$r(t, \mathcal{C}) = \frac{\sum_{k \in \mathcal{C}} w(k, \mathcal{C}) \times r(t, k)}{\sum_{k \in \mathcal{C}} w(k, \mathcal{C})} \tag{5}$$

$$P(t \mid \mathcal{C}) \approx r(t, \mathcal{C})^c \tag{6}$$

Given a keyphrase k to be disambiguated, Equation 3 can be updated as follows:

$$P(t \mid k, \mathcal{C}) \propto P(t \mid k) \times r(t, \mathcal{C})^c \tag{7}$$

Thus, the topic t_o is chosen as the disambiguated topic to k based on the following equation:

$$t_o = \arg\max_{t \in T_k} P(t \mid k) \times r(t, \mathcal{C})^c \tag{8}$$

Through the first-stage disambiguation, using Equation 8, the topic with the highest approximated probability is chosen as the disambiguated sense for an ambiguous keyphrase. We then predict the quality of each disambiguation decision by using a loss function defined in Equation 9.

$$\mathcal{L}(t_o, k, \mathcal{C}) = \sum_{t \in T_k \setminus t_o} P(t \mid k, \mathcal{C}) \propto \sum_{t \in T_k \setminus t_o} P(t \mid k) \times r(t, \mathcal{C})^c \tag{9}$$

We observe that the top two candidate topics of the highest probabilities often dominate the probability space. Thus, given topic t_2 being the second highest likelihood, we approximate the loss function by using $P(t_2|k, C)$. Although the loss function (Equation 9) is defined as a scalar of scale-free, specific to each ambiguous keyphrase, we define the confidence for each disambiguation decision by standardizing the loss function with respect to $P(t_o \mid k, \mathcal{C})$:

$$cfd(t_o, k, \mathcal{C}) = \frac{P(t_o \mid k, \mathcal{C}) - P(t_2 \mid k, \mathcal{C})}{P(t_o \mid k, \mathcal{C})} \tag{10}$$

Observe in Equation 10 that confidence measure $cfd(t_o, k, \mathcal{C})$ is the relative difference between the top two candidate topics and has a range of [0, 1], from the smallest to the largest level of confidence. It is reasonable because a very low confidence is indicative of a random guess among the two candidate topics with very close approximations of $P(t \mid k, \mathcal{C})$. Note that measuring decision confidence using the gap between the two best decisions is a strategy that has been applied in speech recognition tasks (Homma, Aikawa, & Sagayama, 1997; Willett, Worm, Neukirchen, & Rigoll, 1998).

In summary, the first-stage disambiguation involves two parameters: M_1 for the size of the context and c for the probability approximation. A smaller M_1 keeps most useful topics for disambiguation and improves the efficiency, with the risk for filtering away helpful topics as well. A larger M_1, in contrast, may bring in more useful topics, as well as noise, and certainly increases computation. 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).

Second-stage disambiguation. By applying the confidence measure, we classify the disambiguated from the first stage as either high-confidence or low-confidence keyphrases by using a predetermined threshold θ. We believe that the high-confidence keyphrases would provide helpful information to better disambiguate the keyphrases of low confidence. Thus, in the second stage, we update context $\mathcal{C}$ with the topics referred to by the high-confidence keyphrases. The ambiguous keyphrases of low confidence are disambiguated again based on the updated context. This is reasonable because the low-confidence disambiguations indicate that the context we

DOI: 10.1002/asi

used in the first stage may not provide adequate discriminative information for these difficult ambiguous keyphrases. Specifically, we add M_2 topics from the high-confident keyphrases as additional context keyphrases. The resulting new set of context keyphrases is denoted by $\mathcal{C}_2$.

The new M_2 topics used in the second stage are selected from the high-confident keyphrases by leveraging the semantic clue provided by the exiting context keyphrases used in the first stage. More specifically, we assign a score to each disambiguated keyphrase as follows:

$$\begin{aligned} score(t_o, k, \mathcal{C}) &= cfd(t_o, k, \mathcal{C}) \times P(t_o \mid \mathcal{C}) \\ &\approx cfd(t_o, k, \mathcal{C}) \times r(t_o, \mathcal{C})^c \end{aligned} \quad (11)$$

Observe that the disambiguated keyphrase (a) that has higher confidence and (b) is highly related to the existing context is assigned a higher score. Because each disambiguated keyphrase refers to a specific Wikipedia topic, we assume that picking topics with high relatedness to the existing context $\mathcal{C}$ would provide more discriminative information. For low-confident keyphrases, we redisambiguate them using the new context $\mathcal{C}_2$:

$$t_o = \arg\max_{t \in T_k} P(t \mid k) \times r(t, \mathcal{C}_2)^c \quad (12)$$

The parameter M_2 is similar to the parameter M_1 in the first stage to balance the efficiency and effectiveness. That is, a small M_2 may not provide enough discriminative information; a large M_2 brings in not only more useful information for disambiguation but also noise and extra computation. Thus, TSDW involves four parameters: M_1 for the context size in the first stage, c for the relatedness measure scaling, confidence threshold θ, and M_2 for the size of additional context keyphrases in the second stage.

The existing works (Li et al., 2011; Medelyan et al., 2008; Milne & Witten, 2008b; Ratinov et al., 2011) can be considered as specific cases of TSDW without the second-stage disambiguation (i.e., $M_2 = 0$). In the first stage of disambiguation, these works differ in the way to approximate $P(t \mid k, \mathcal{C})$. Medelyan et al. tried to approximate $P(t \mid \mathcal{C})$ by considering all unambiguous keyphrases equally. Milne and Witten improved the approximation by weighing all unambiguous keyphrases based on their relatedness to each other. The latter also tried to approximate $P(t \mid k, \mathcal{C})$ indirectly by using machine learning techniques. The approach proposed by Ratinov et al. is difficult to be compared with directly, because they tried to obtain a better approximation of $P(t \mid \mathcal{C})$ by combining the relatedness measures based on both local and global information in a supervised learning manner. Our previous approach (Li et al., 2011) constitutes the first-stage disambiguation only. Note that only the low-confident keyphrases are redisambiguated in the second stage with the updated context. In the next section, we empirically study the performance of TSDW and other state-of-the-art approaches in terms of both effectiveness and efficiency.

Experiments

In this section, we describe extensive experiments to evaluate the performance of TSDW. To demonstrate that TSDW is generic in accommodating different relatedness measures and Wikipedia in different languages, we evaluate TSDW using three relatedness measures on two versions of Wikipedia (English and Traditional Chinese). After detailing the TSDW setup and performance metric, we report the performance comparison between TSDW and existing methods. Last, we report a performance analysis of TSDW in different settings.

TSDW Setup and Performance Metric

Wikipedia inventory. We built two Wikipedia inventories from the English and Traditional Chinese Wikipedia dumps, respectively. Specifically, we used the English Wikipedia dump released on January 30, 2010.[7] There are 3,246,821 articles and 266,625,017 hyperlinks among them, excluding all redirection pages. The built Wikipedia keyphrase inventory consists of 6,168,269 unambiguous keyphrases and 526,081 ambiguous keyphrases. For the latter, each keyphrase refers to 4.22 candidate topics on average. The Traditional Chinese Wikipedia dump released on June 28, 2011,[8] is used to build the Chinese version of a keyphrase inventory. There are 355,245 articles and 24,720,728 hyperlinks among them, excluding all redirection pages. The built Wikipedia keyphrase inventory consists of 815,303 unambiguous keyphrases and 40,895 ambiguous keyphrases. Each ambiguous keyphrase refers to 3.0 candidate topics on average.

Relatedness measure. Three relatedness measures, namely, Dice (Dice, 1945), Jaccard (Jaccard, 1901), and WLM (Milne & Witten, 2008a) are investigated in TSDW for validating the general applicability of the framework. All three measures compute the relatedness between two Wikipedia articles by considering their incoming wikilinks:

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

$$r_{Jaccard}(a, b) = \frac{|A \cap B|}{|A \cup B|}$$

$$r_{Dice}(a, b) = \frac{2|A \cap B|}{|A| + |B|}$$

where a and b are two Wikipedia articles, A and B are the sets of Wikipedia articles that link to a and b, respectively, and W is the number of articles in Wikipedia. Although Dice and Jaccard are general similarity measures over sets, WLM

[7] http://dumps.wikimedia.org/enwiki/
[8] http://dumps.wikimedia.org/zhwiki/

DOI: 10.1002/asi

TABLE 1. Statistics on data sets.

Data set	No. of articles	No. of words	No. unambiguous	No. ambiguous	No. of candidates
$Trainning_{en}$	500	9,759	103.5	30.9	45.7
$Validation_{en}$	100	11,294	117.0	38.4	46.3
$Evaluation_{en}$	200	9,647	108.2	37.9	46.8
$Training_{zh}$	500	4,453	260.6	24.9	10.5
$Validation_{zh}$	100	3,837	229.1	24.1	12.5
$Evaluation_{zh}$	200	3,969	237.4	21.0	11.3

Note. en and *zh* refer to English and Traditional Chinese, respectively; articles is the number of Wikipedia articles; number of words is the average number of words per Wikipedia article; number of unambiguous refers to the average number of unambiguous keyphrases per Wikipedia article; number of ambiguous refers to the average number of ambiguous keyphrases per Wikipedia article; and number of candidates is the average number of candidate topics per ambiguous keyphrase.

is a Wikipedia-specific similarity measure based on *Normalized Google Distance* (Cilibrasi & Vitanyi, 2007). In contrast with Dice and Jaccard, WLM explicitly considers the issue that the two sets under consideration would have very unbalanced cardinalities. WLM has been widely adopted in the related works that exploit the semantic resources of Wikipedia (Han & Zhao, 2009; Huang, Milne, Frank, & Witten, 2012; Li et al., 2011; Milne & Witten, 2008b; Ratinov et al., 2011; Scaiella, Ferragina, Marino, & Ciaramita, 2012).

Performance metric. In our experiments, for each ambiguous keyphrase k to be disambiguated, exactly one candidate topic t is assigned to k by a disambiguation method to be evaluated. We report the *accuracy* of the assignments, that is, the ratio of the correct assignments for all ambiguous keyphrases involved in the evaluation. The correct assignments are determined by manual verification in our experiments. Note that because each ambiguous keyphrase cannot have more than one sense in a given context, the accuracy reported in this article is the same as precision or recall.

Efficiency performance is also reported for the experiments. All experiments are conducted on the same workstation with a 2.40 GHz Xeon quad-core CPU and 24 GB of RAM. The execution time by each method is the time taken for keyphrase recognition and keyphrase disambiguation, ignoring the time taken for data loading or classifier training.

Comparison With Other Methods

In this set of experiments, we compare our method with four state-of-the-art methods and two baseline methods for both effectiveness and efficiency. Specifically, we compare our method with the methods reported in Milne and Witten (2008b), Medelyan and colleagues (2008), and Ratinov and coworkers (2011) as well as our previous proposed approach (Li et al., 2011). The first builds machine learning classifiers to disambiguate the keyphrases; the second maximizes the balance between commonness and relatedness using equal weight; the third combines both the local word windows and link structures for disambiguation; and the last is equivalent to the first-stage disambiguation (i.e., $M_2 = 0$ in TSDW). For the first method, we build two classifiers C4.5 and Bagged C4.5 using Weka library (Hall et al., 2009). 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.

Note that Ratinov and coauthors (2011) published the implementation[9] of their system (called *Illinois Wikifier*) together with the four data sets used in their work. Because their system is implemented based on the Wikipedia dump of 2009, we compare Illinois Wikifier and TSDW on the four data sets used in their work separately. Given that Illinois Wikifier requires NER tagger and shallow parser, we do not compare it with TSDW on Traditional Chinese articles.

For comparison with the other methods, we prepare 2 data sets of 800 articles each, sampled from English and Traditional Chinese Wikipedia, respectively. In each data set, these 800 articles are randomly split into 3 nonoverlapping subsets of 500, 100, and 200 articles. The subset of 500 articles is used for classifier training. The trained classifiers are validated using the subset of 100 articles. For a fair comparison, all methods are evaluated on the subset of 200 articles of each data set. The statistics of the two data sets with their subsets are reported in Table 1.

We first report the evaluation results on the English data set. We use the following parameter settings for TSDW: $c = 1.5$, 1.5, and 6.0 for Dice, Jaccard, and WLM, respectively, and $\theta = 0.9$ and $M_2 = 5$, according to the findings reported in evaluation of TSDW. As for the size of the context in the first stage, we vary M_1 using three different settings: 5, 10, and 15. The disambiguation accuracy and execution time of the evaluated methods on the English evaluation set are reported in Table 2 (see columns 2 and 3). Note that, for *random sense*, the result is averaged over 10 runs. The number of ambiguous keyphrases that are processed by the second-stage disambiguation is reported in Table 3 (see column 2).

Overall, TSDW with WLM and $M_1 = 10$, $M_2 = 5$ achieves the best accuracy among all methods. Meanwhile, the

[9]http://cogcomp.cs.illinois.edu/page/download_view/Wikifier

DOI: 10.1002/asi

TABLE 2. Disambiguation accuracy and execution time on English and Traditional Chinese evaluation sets. The highest accuracy achieved for each similarity measure is highlighted in boldface.

	English		Traditional Chinese	
Method	Accuracy (%)	Time (seconds)	Accuracy (%)	Time (seconds)
Random sense	18.25	13	27.99	<1
Most common sense	78.21	41	92.56	<1
Medelyan and colleagues	85.96	5568	94.65	757
M&W with C4.5	85.60	5917	93.79	1167
M&W with bagged C4.5	85.14	5948	93.98	1164
TSDW (Dice, $M_1 = 5$, $M_2 = 0$)	92.39	78	95.96	64.0
TSDW (Dice, $M_1 = 5$, $M_2 = 5$)	**93.08**	104	96.00	64.2
TSDW (Dice, $M_1 = 10$, $M_2 = 0$)	92.68	155	**96.39**	73.0
TSDW (Dice, $M_1 = 10$, $M_2 = 5$)	92.79	183	96.24	73.3
TSDW (Dice, $M_1 = 15$, $M_2 = 0$)	92.35	226	96.29	79.0
TSDW (Dice, $M_1 = 15$, $M_2 = 5$)	92.58	258	96.31	82.4
TSDW (Jaccard, $M_1 = 5$, $M_2 = 0$)	92.25	78	95.91	62.0
TSDW (Jaccard, $M_1 = 5$, $M_2 = 5$)	**92.96**	104	95.93	64.7
TSDW (Jaccard, $M_1 = 10$, $M_2 = 0$)	92.47	156	**96.24**	72.0
TSDW (Jaccard, $M_1 = 10$, $M_2 = 5$)	92.67	183	**96.24**	74.0
TSDW (Jaccard, $M_1 = 15$, $M_2 = 0$)	92.08	225	96.22	79.0
TSDW (Jaccard, $M_1 = 15$, $M_2 = 5$)	92.52	259	96.08	82.6
TSDW (WLM, $M_1 = 5$, $M_2 = 0$)	93.74	78	96.69	64.0
TSDW (WLM, $M_1 = 5$, $M_2 = 5$)	94.15	102	97.05	64.6
TSDW (WLM, $M_1 = 10$, $M_2 = 0$)	94.14	167	97.19	72.0
TSDW (WLM, $M_1 = 10$, $M_2 = 5$)	**94.37**	180	97.34	72.4
TSDW (WLM, $M_1 = 15$, $M_2 = 0$)	94.14	226	**97.41**	79.0
TSDW (WLM, $M_1 = 15$, $M_2 = 5$)	94.35	256	97.36	81.5

M&W = Milne and Witten; TSDW = two-stage word sense disambiguation to Wikipedia; WLM = Wikipedia link-based measure.

TABLE 3. Number of ambiguous keyphrases processed by the second-stage disambiguation with different settings for TSDW on English and Traditional Chinese evaluation sets.

Setting	English	Traditional Chinese
TSDW (Dice, $M_1 = 5$)	1461	427
TSDW (Dice, $M_1 = 10$)	1485	400
TSDW (Dice, $M_1 = 15$)	1421	411
TSDW (Jaccard, $M_1 = 5$)	1459	420
TSDW (Jaccard, $M_1 = 10$)	1493	415
TSDW (Jaccard, $M_1 = 15$)	1424	423
TSDW (WLM, $M_1 = 5$)	923	314
TSDW (WLM, $M_1 = 10$)	871	292
TSDW (WLM, $M_1 = 15$)	857	290

TSDW = two-stage word sense disambiguation to Wikipedia; WLM = Wikipedia link-based measure.

methods with Dice and Jaccard yield competitive accuracies. The method of Medelyan and colleagues (2008) performs significantly better than Milne and Witten (2008b) with C4.5 and bagging C4.5 classifiers. Although classifier bagging improves the accuracy by 0.3% in (Milne & Witten, 2008b), it does not contribute positively to the accuracy in our experiments. All these methods, in contrast, significantly outperform the two baselines. Specifically, most common sense delivers an accuracy rate of 78.21%, and random guess has an accuracy rate of 18.25%. Compared with the approach without second-stage disambiguation (i.e., $M_2 = 0$), the two-stage disambiguation offers positive improvements on the accuracy for all three relatedness measures and M_1 values. Considering that the number of ambiguous keyphrases undergoing the second-stage disambiguation process is relatively small, the contribution is rather significant. As for efficiency, our results are significantly faster than Medelyan and colleagues and Milne and Witten (2008b). For instance, with $M_1 = 10$ and $M_2 = 5$, TSDW achieves at least 30 and 10 times faster than Medelyan and colleagues and Milne and Witten (2008b), respectively. Observe that the additional execution time incurred by the second-stage disambiguation is relatively small, because of the small number of ambiguous keyphrases that need to go through the second-stage disambiguation process. Note that the time taken by random sense and most common sense is mainly for keyphrase recognition because of the large size of the Wikipedia inventory. Based on the results, we highlight that the second-stage disambiguation improves both effectiveness and efficiency, compared with the approaches with first-stage disambiguation only. More specifically, by setting $M_1 = 5$, $M_2 = 5$ in TSDW, a higher accuracy and efficiency are achieved than by applying only first-stage disambiguation with $M_1 = 10$ (i.e., TSDW with $M_1 = 10$, $M_2 = 0$) for all three relatedness measures. Similar observation is made for the case of TSDW with $M_1 = 10$, $M_2 = 5$.

We next report the experimental results on the Traditional Chinese data set. For *TSDW*, we set $c = 1.4$, 1.3, and 4.0 for Dice, Jaccard, and WLM, respectively. The confidence

DOI: 10.1002/asi

TABLE 4. Statistics on the four data sets.

Data set	No. of articles	No. of words	No. unambiguous	No. ambiguous	No. of candidates
AQUAINT	50	1347	18.4	8.8	55.4
MSNBC	20	3262	40.0	10.3	81.6
ACE	57	2276	28.4	2.6	94.3
Wikipedia	40	3628	47.1	12.3	47.4

Note. Number of words is the average number of words per document; number unambiguous refers to the average number of unambiguous keyphrases per document; number ambiguous refers to the average number of ambiguous keyphrases per document; and number of candidates is the average number of candidate topics per ambiguous keyphrase.

threshold θ and M_2 are fixed at 0.9 and 5. Similarly, we vary M_1 to three different values: 5, 10, and 15.

The disambiguation accuracy and execution time of all methods on the Traditional Chinese data set are reported in Table 2 (see columns 4 and 5). The number of ambiguous keyphrases that are processed by the second-stage disambiguation is reported in Table 3 (see column 3). Observe that the Traditional Chinese data set is easier for word sense disambiguation compared with the English version. Random sense and most commmon sense offer accuracy rates of 27.99% and 92.56% respectively, a much better performance than for English articles, probably because each ambiguous keyphrase in the Chinese Wikipedia inventory has only 3.0 candidate topics on average, compared with 4.2 in the English version. A similar difference is also reflected in Table 1 (see column 6). All methods achieve better accuracies than the *most common sense* baseline. Similar to what we observed on the English data set, Medelyan and colleagues' method performs significantly better than Milne and Witten with C4.5 and bagging C4.5 classifiers. The classifier with bagging achieves marginally better accuracy than C4.5. Our previous approach achieves much better performance than the three existing approaches in terms of both accuracy and efficiency, for all three relatedness measures and M_1 values. Meanwhile, our previous approach offers 90% reduction, on average, in computation time compared with the three existing approaches. TSDW gives both marginally positive and negative effects across all settings. Five of the nine cases benefit slightly from the second-stage disambiguation; three of the nine cases show slight performance degradation. The additional computation time incurred by the second-stage disambiguation is almost negligible. This is confirmed by the relatively small number of ambiguous keyphrases handled by the second stage, mainly because the accuracy rate of the first-stage disambiguation is very high (about 96%). With $M_1 = 5$, we observe that TSDW achieves a slightly better performance than the corresponding method with first-stage disambiguation only for all three relatedness measures. This indicates that the updated context in the second-stage disambiguation indeed brings in more discriminative information. When $M_1 = 10$ or 15, degradations are observed for some settings. One possible reason is that Traditional Chinese Wikipedia is still immature and under development. Compared with English Wikipedia, Traditional Chinese Wikipedia is an order-of-magnitude smaller in the size. Thus, a larger context (i.e., $M_1 = 15, M_2 = 5$) may incur more noise than benefit. However, the experimental results (5 vs. 3) demonstrates that TSDW is still a promising approach across different languages. Considering Tables 2 and 3 together, we can see that the number of ambiguous keyphrases of low confidence is positively correlated with the accuracy for both English and Traditional Chinese articles. It partially justifies the correctness of the confidence measure we proposed in Equation 10 (further investigation of the affect by confidence threshold θ is conducted in evaluation of TSDW).

Comparison With Illinois Wikifier

In this section, we compare TSDW with Illinois Wikifier on the four data sets used in Ratinov and coworkers (2011), ranging from short newswire to Wikipedia paragraphs. The four data sets are summarized as follows:

- AQUAINT is a subset of the AQUAINT corpus of newswire text where the Wikipedia keyphrases are annotated to Wikipedia topics. This data set was also used in Milne and Witten (2008b).
- MSNBC is taken from MSNBC news, where only named entities after running NER are disambiguated to Wikipedia. This data set was used in Cucerzan (2007).
- ACE is a subset of ACE co-reference data set that was built by Ratinov and coworkers (2011). The annotations to Wikipedia topics were assigned by Amazon Mechanical Turk,[10] and the inconsistent annotations were manually corrected by the authors.
- Wikipedia is a sample of paragraphs from Wikipedia articles sampled by Ratinov and colleagues (2011). The annotations correspond to the hyperlinks in the Wikipedia text.

Because Illinois Wikifier was built using the English Wikipedia dump of 2009, for a fair comparison, we retain all the annotations that are both solvable to Illinois Wikifier and the Wikipedia inventory we built in this work. Solvable annotations refer to the annotations that appear in the inventory and the correct disambiguations are among the candidates indexed by the inventory (Ratinov et al., 2011). The unsolvable annotations in the data sets are removed. The details of the four data sets are reported in Table 4.

[10]https://www.mturk.com/

DOI: 10.1002/asi

TABLE 5. Disambiguation accuracy (%) and execution time (second) of TSDW and Illinois Wikifier on four evaluation sets: $M = 10$ for TSDW. The highest accuracy achieved for each evaluation set is highlighted in boldface.

	ACE		AQUAINT		MSNBC		Wikipedia	
Method	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
Illinois Wikifier	85.91	123.4	84.35	99.3	81.76	108.4	87.39	264.1
TSDW (WLM, $M_2 = 0$)	**93.92**	7.1	91.16	8.8	84.47	9.5	88.84	10.0
TSDW (WLM, $M_2 = 5$)	93.24	8.2	91.84	10.2	85.44	10.4	**90.47**	10.6
TSDW (Jaccard, $M_2 = 0$)	**93.92**	7.2	92.97	8.7	**85.92**	9.0	86.00	9.9
TSDW (Jaccard, $M_2 = 5$)	**93.92**	8.1	93.20	9.9	**85.92**	10.2	87.22	11.0
TSDW (Dice, $M_2 = 0$)	**93.92**	7.0	92.97	8.7	85.44	8.6	86.00	10.0
TSDW (Dice, $M_2 = 5$)	**93.92**	8.1	93.42	10.0	**85.92**	10.1	87.22	11.2

TSDW = two-stage word sense disambiguation to Wikipedia; WLM = Wikipedia link-based measure.

Table 5 reports the disambiguation accuracies and execution times on the four data sets. We fix $M_1 = 10$ and $M_2 = 5$ for TSDW. Observe that TSDW outperforms Illinois Wikifier in almost all cases. Although the second-stage disambiguation of TSDW does not contribute any positive improvement on ACE, it provides additional benefit for the other three data sets. The main reason is that there are few ambiguous keyphrases from the articles in the ACE data set; the data set has only 2.6 ambiguous keyphrases, on average, for each article. For this reason, almost no additional discriminative information can be explored in the second-stage disambiguation. Illinois Wikifier achieves comparable accuracy with our method on Wikipedia data set for most settings used in TSDW. However, using WLM as relatedness measure and two-stage disambiguation, our method achieves the best accuracy on the data set. In terms of efficiency, as can be observed in Table 5, TSDW is at least 10 times faster than the Illinois Wikifier for all four data sets. Note that Ratinov and colleagues (2011) used only the top 20 candidate topics of the highest likelihood in their system, whereas the average number of candidate topics per ambiguous keyphrase for the four data sets varies from 47 to 94 in TSDW (see Table 4). This again confirms that TSDW is superior to Illinois Wikifier in terms of efficiency. Compared with the performance reported by Ratinov and colleagues, we observed that the disambiguation accuracy degrades a bit for Illinois Wikifier in our experiments. One possible explanation is that the different sets of solvable annotations are used in the experiments. In particular, only a subset of solvable annotations used by Ratinov and colleagues is used for our evaluation. Similar performance deterioration is also observed for TSDW on the Wikipedia data set, compared with our previous experiments conducted earlier. Recall that the AQUAINT data set was also used in Milne and Witten's work (2008b). Although it is unfair to compare the effectiveness directly because of different experimental settings, an accuracy rate of 76% was reported in their original work (Milne & Witten, 2008b).

Evaluation of TSDW

To evaluate the disambiguation accuracy of TSDW and the impact of the parameter settings, we conduct another set of experiments on the training sets of 500 articles in English and Traditional Chinese, respectively. Recall that our proposed approach involves four parameters, M_1, M_2, c, and θ, and a relatedness measure. M_1 determines the number of unambiguous keyphrases involved in the first-stage disambiguation, M_2 determines the number of additional context keyphrases in the second stage, θ is the threshold for the high/low-confident keyphrases from the first-stage disambiguation, and c is specific to the relatedness measure.

First-stage disambiguation. The first-stage disambiguation involves two parameters: M_1 for the size of the context and c for the probability approximation. We apply grid search to analyze the impact of these two parameters on both data sets: English and Traditional Chinese data sets. We investigate the impact of M_1 value on the disambiguation accuracy by varying M_1 from 5 to 50 with a step of 5, and *All* that takes all unambiguous keyphrases into account, given c is fixed. Similarly, we learn a c value for each given relatedness measure by varying c from 0 to 10 with a step of 0.1, given M_1 is fixed. Observe that, when $c = 0$, our method degrades to the most common sense method. Three types of relatedness measures—Dice, Jaccard, and WLM—are evaluated in TSDW.

Figure 3 reports the disambiguation accuracy of the first-stage disambiguation by varying M_1 and c on three relatedness measures for English (Figure 3a) and Traditional Chinese data sets (Figure 3b). We made the following observations on the experimental results:

- Parameter c significantly affects the results for all relatedness measures. For a specific relatedness measure, different optimal c values are observed for the two data sets. When *Dice* is used, $c = 1.5/1.4$ gives the best accuracy across different M_1 values for English and Traditional Chinese data sets, respectively. Similarly, $c = 1.5/1.3$ gives the best accuracy for Jaccard. For WLM, the best accuracy is achieved when c is in the range of [5.0, 7.0]/[3.5, 5.0] for English and Traditional Chinese data sets, respectively.
- A larger M_1 does not necessarily lead to better accuracy for both data sets. In particular, accuracies dropped for all settings when $M_1 = All$, that is, taking all unambiguous keyphrases as the context. Specifically, $M_1 = 5$, 10, and 15 offer the best accuracies.

DOI: 10.1002/asi

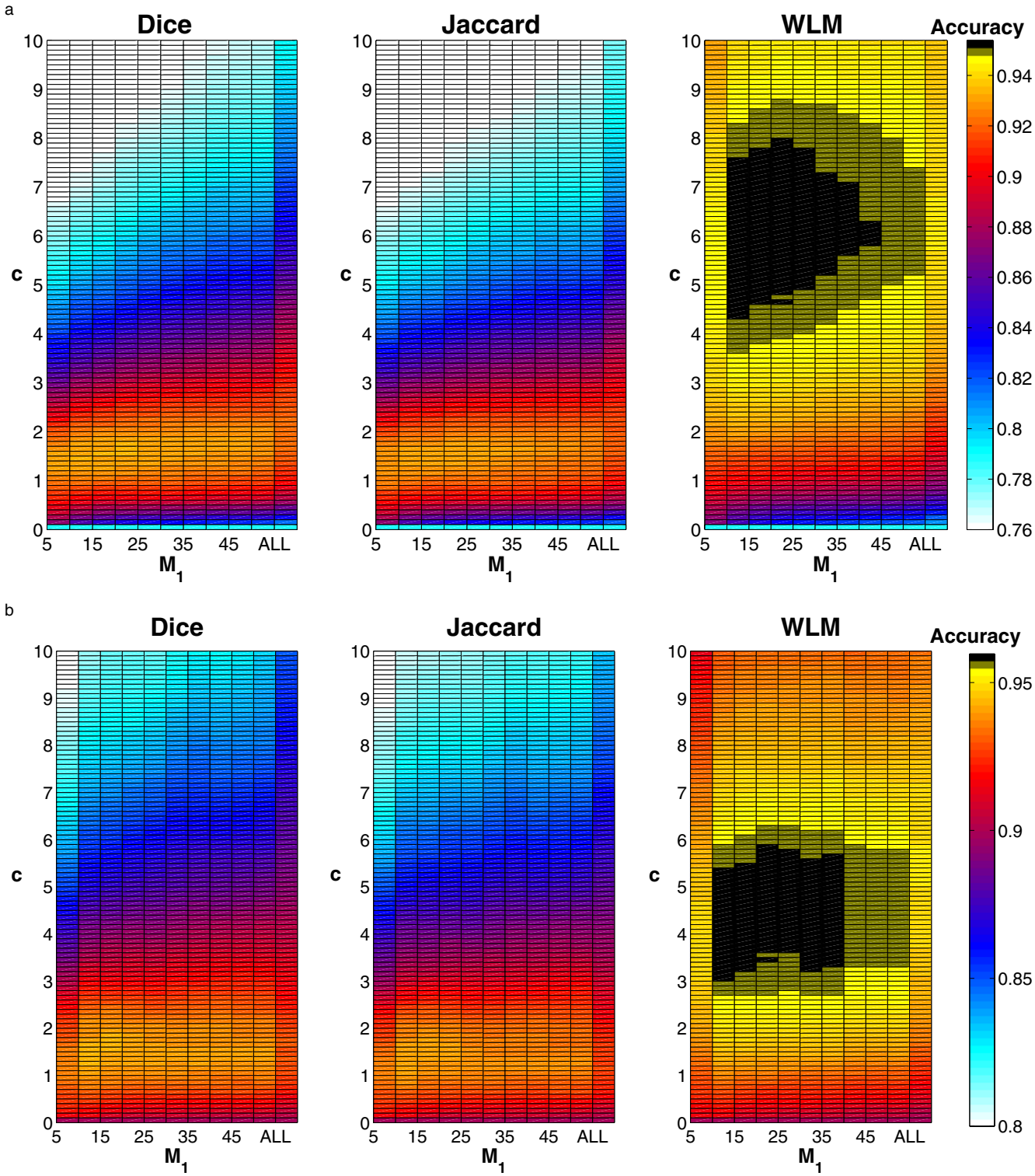

FIG. 3. Accuracy of varying M_1 and c with Dice, Jaccard, and Wikipedia link-based measure (WLM) for English (a) and Traditional Chinese (b) data sets. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Observe that Figures 3a and 3b demonstrate very similar patterns. Nevertheless, the comparison of the two sets of results reveals that parameter c is dependent on not only the relatedness measure but also the Wikipedia data set. Considering the size difference between the two Wikipedia data sets (English and Traditional Chinese), the characteristics of a specific relatedness measure based on the hyperlink structure would be largely affected. To better demonstrate the impact of c value on the three relatedness measures over the two languages, as a case study, we calculate the pairwise

DOI: 10.1002/asi

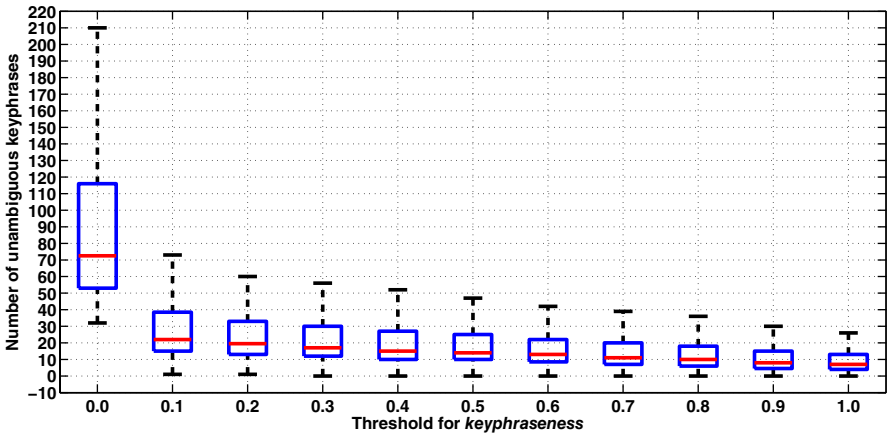

FIG. 4. Box plot for the number of unambiguous keyphrases per article with *keyphraseness* value above a threshold. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

relatedness measure between the Wikipedia topic *Great Wall of China* and all its outgoing neighbors for both English and Traditional Chinese, respectively, using Dice, Jaccard, and WLM. There are, respectively, 105 and 335 outgoing neighbors for English and Traditional Chinese articles about the topic Great Wall of China. Table 6 reports the mean, standard deviation, and coefficient of variation for each set of similarity values to empirically reflect the varying characteristics of each similarity measure over Wikipedia of the two languages. The relatedness values by Dice and Jaccard are widely scattered for both English and Traditional Chinese settings. WLM produces a narrow dispersion of relatedness values. This is consistent with the observations made in Figure 3 that a larger c obtains a better disambiguation accuracy with WLM. Moreover, we can observe that each similarity measure holds different value patterns for English and Traditional Chinese, in terms of the three statistics we studied in this article. Hence different optimal c values for a specific relatedness measure over the two Wikipedia data sets are reasonable. This also illustrates the robustness of TSDW in adapting to different settings.

In the first-stage disambiguation, TSDW filters away noisy contextual information by retaining only top M_1 unambiguous keyphrases with the highest *keyphraseness* values. An alternative option is to apply a predefined threshold value for the *keyphraseness*, so that all unambiguous keyphrases with *keyphraseness* value larger than the threshold are considered as the context. Such a threshold can be applied globally to all articles of interest. However, given such a threshold, an article may have too many or too few unambiguous keyphrases left as context. As observed in Figure 3, more unambiguous keyphrases did not lead to significantly better disambiguation accuracy on either English or Traditional Chinese data set. Figure 4 plots the box plot[11] of the number of unambiguous keyphrases per article with *keyphraseness* values larger than a specific threshold for the 500 articles of English training set. We make two observations: (a) 50% of the articles contain more than 10 unambiguous keyphrases with *keyphraseness* larger than 0.8, and (b) 10% of the articles contain just one unambiguous keyphrase whose keyphraseness is larger than 0.1. That is, a high threshold on keyphraseness leads to a empty context for these articles.

We also conducted experiments for the first-stage disambiguation on the English training set by applying a specific keyphraseness threshold. If an article has no unambiguous keyphrase left as context after applying the threshold, most common sense is used for disambiguation. Table 7 lists the accuracies obtained by using different threshold values, along with the corresponding computation time. Given the marginal change of the accuracy, two observations were made: (a) a low threshold leads to more computation and relatively low accuracy, and (b) a high threshold achieves a shorter computation time with relatively low accuracy. These two observations are consistent with the results shown in Figure 4. By applying $M_1 = 15$, an accuracy rate of 95.32% is achieved with execution time of 482.8 seconds. Compared with results listed in Table 7, retaining top M_1 unambiguous keyphrases as the context achieves comparable accuracy with shorter computation time.

The first-stage disambiguation of TSDW predicts a confidence for each disambiguation decision. In this study, we investigate the correctness of the confidence measure used in Equation 10 by using the three relatedness measures and three M_1 values: 5, 10, 15. Parameter c for Dice, Jaccard, and WLM is fixed at 1.5, 1.5, 6.0 / 1.4, 1.3, 4.0 for English and Traditional Chinese, respectively. Let the accuracy at confidence *cfd* be the accuracy of disambiguation decisions with confidence less than or equal to *cfd*. Figure 5 plots the accuracies at varying *cfd* on the English (Figure 5a) and Traditional Chinese (Figure 5b) training data sets. Observe that the accuracy increases monotonically with the confidence. An accuracy rate of about 50% is achieved when the confidence is 0.5 for all settings. The accuracy of

[11]Box plot is a convenient way of graphically depicting groups of numerical data through their five-number summaries: 10th, 25th, 50th, 75th, and 90th percentiles.

DOI: 10.1002/asi

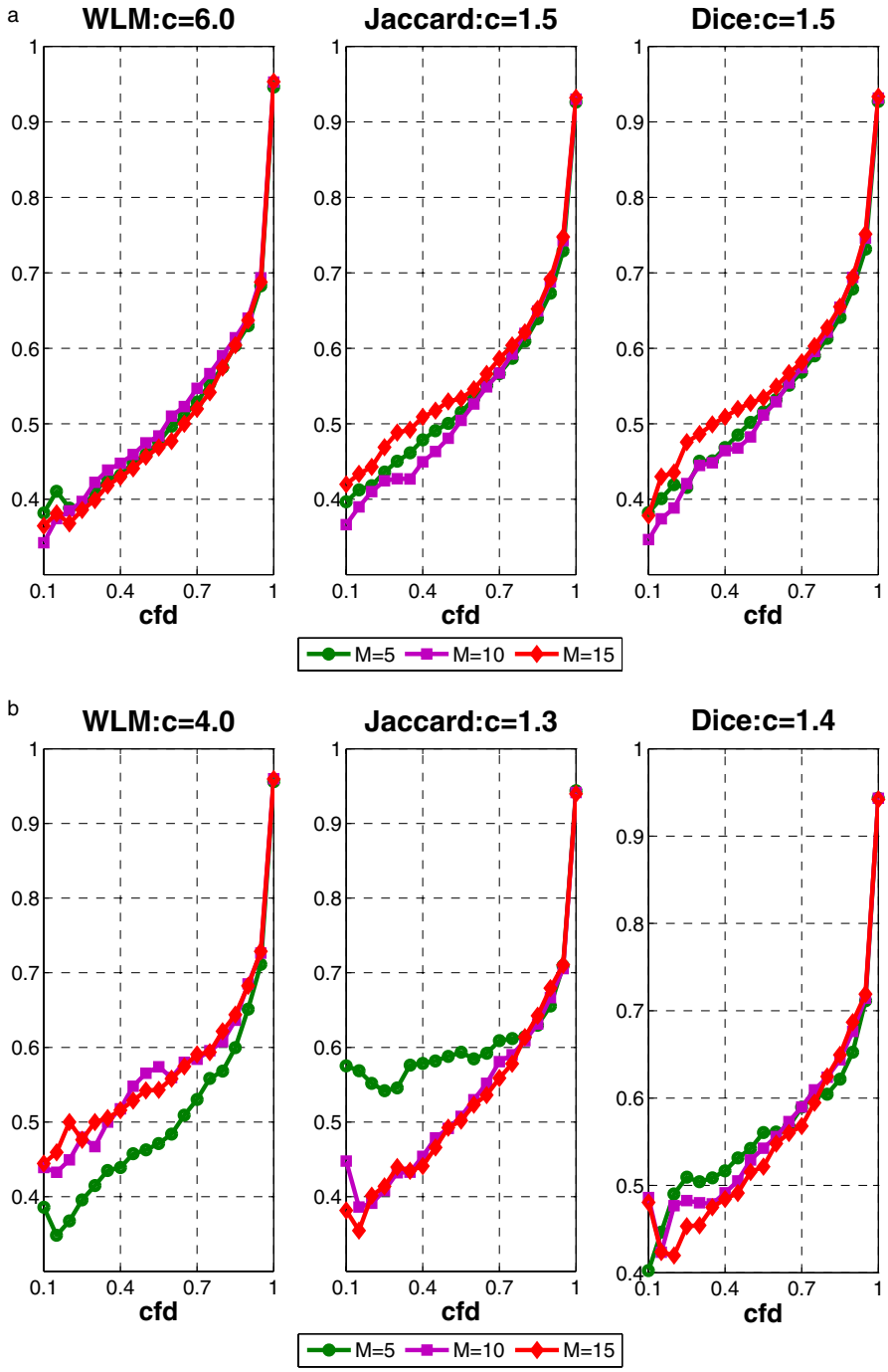

FIG. 5. Accuracy (%) at varying θ on the English (a) and Traditional Chinese (b) training sets. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

disambiguation decisions with confidence larger than or equal to 0.9 is more than 98% for all cases. Accordingly, we set $\theta = 0.9$ to distinguish the low/high-confidence disambiguation decisions. The experimental results show that the confidence measure is a reasonable indicator to reflect the quality of disambiguation decisions.

Second-stage disambiguation. The second-stage disambiguation of TSDW involves two parameters: confidence threshold θ and the number of new context keyphrases M_2. According to experimental results in the previous section, we fix θ to be 0.9. The impact of M_2 is studied by varying its value from 0 to 20 with a step of 5. The experiments are conducted using $M_1 = 5$, 10, 15 and the best c on the three relatedness measures found earlier. Figure 6 illustrates the effectiveness of the second-stage disambiguation on different M_2 values on English (Figure 6a) and Traditional Chinese (Figure 6b) training data sets. Note that $M_2 = 0$ is

DOI: 10.1002/asi

TABLE 6. Relatedness distribution using Dice, Jaccard, and WLM.

	Mean		SD		CV	
Measure	En	Zh	En	Zh	En	Zh
Dice	0.025	0.078	0.028	0.096	1.120	1.231
Jaccard	0.013	0.044	0.015	0.070	1.154	1.591
WLM	0.435	0.527	0.245	0.166	0.563	0.315

Note. En and Zh refer to the case study by using English and Traditional Chinese article, respectively, about the topic *Great Wall of China*. CV = coefficient of variation; SD = standard deviation; WLM = Wikipedia link-based measure.

TABLE 7. Disambiguation accuracy and execution time by applying varying keyphraseness thresholds on the English training set.

Threshold	Accuracy	Time	Threshold	Accuracy	Time
0.1	95.27	1183.7	0.6	95.37	740.9
0.2	95.31	1019.6	0.7	95.36	672.0
0.3	95.34	935.4	0.8	95.23	622.8
0.4	95.34	878.4	0.9	94.94	521.2
0.5	95.40	824.5	1.0	94.89	456.6

TABLE 8. Accuracies (%) and the number of the high-confidence disambiguation decisions (HiCfd) at different stages.

Stage	Accuracy	HiCfd
1	94.63	13,708
2	95.34	399
3	95.36	55
4	95.37	8

equivalent to the output of using first-stage disambiguation only. From Figure 6a, we observe that the second-stage disambiguation further improves the disambiguation accuracy with varying M_2 values for English articles. The largest improvement is observed when $M_2 = 5$ for all the three relatedness measures. This indicates that more additional context keyphrases bring in more noise than discriminative information. For articles in Traditional Chinese, this side effect of bringing in more noisy context keyphrases becomes more apparent. The marginal improvement by the second-stage disambiguation is obtained when M_1 is 5 for all the three relatedness measures. We believe that this is specific to the nature of the Traditional Chinese Wikipedia we studied (i.e., incomplete and relatively small in size). Overall, based on the experimental results, we see that $M_2 = 5$ is a reasonable value for the size of additional context in the second stage. For both the English and Traditional Chinese articles, WLM achieves the best performance over Dice and Jaccard. This has been confirmed by the extensive experiments conducted in earlier sections. Given the size of ambiguous keyphrases undergoing the second-stage disambiguation is relatively small, the positive improvements we observed in this study are significant.

Multistage disambiguation. As demonstrated in Figure 2, TSDW naturally supports multiple redisambiguations, such as a third-stage disambiguation. The stop criterion for further disambiguation stage can be defined based on the amount of additional information gained through the disambiguation decisions of the current stage, that is, the number of high-confidence disambiguation decisions of the current stage. Given the high accuracy of the first-stage disambiguation, it is expected that the additional information is very limited after the second-stage disambiguation. We conducted experiments on the English training set with up to fourth-stage disambiguation by using WLM relatedness measure. We fixed θ, c, and M_1 to 0.9, 6.0, and 5, respectively. The number of additional context keyphrases added at the next stage is fixed to be 5, that is, $M_i = 5$ for $i = 2, 3, 4$. Table 8 reports the accuracies along with the number of the high-confidence disambiguation decisions obtained after each stage. Observe that, after the second stage, only 399 high-confidence disambiguation decisions are obtained for the 500 articles. This leads to little improvement at further stages. Similar observations hold for different settings (i.e., relatedness measure, parameter values, and language). Thus, we restrict TSDW to have two-stage disambiguation for better efficiency and negligible loss in effectiveness.

Error analysis. We analyzed the disambiguation errors made by TSDW manually. We found that most disambiguation errors happen when the correct topic is ranked as the second best candidate topic, particularly when both the top two best candidate topics are closely related to the context of the article. For example, *joey* cannot be disambiguated correctly to the topic *Joey* (1985 film), but to a wrong topic *Joey* (TV series) in Wikipedia article *Don Porter*.[12] The actor *Don Porter* appeared in the film *Joey* (1985 film).[13] Because both topics belong to the same concept *show business*, they have very similar context-relatedness values. Thus, topic *Joey* (TV series) is wrongly selected because it has a much higher prior probability than topic *Joey* (1985 film) (0.462 vs. 0.016). We observe that these kinds of mistakes are often related to low-confident disambiguation decisions. In this situation, we believe that the use of other relatedness measures, such as cosine similarity based on bag-of-word model, might help us out. Another source of disambiguation errors is that an ambiguous keyphrase happens to have a false candidate topic that is very related to the main topic of the article, although its true topic is related only to the surrounding semantic context. For example, the ambiguous keyphrase *underground* appears in the Wikipedia article *The Vinyl Underground*,[14] which is about a comic book series. *underground* refers to the Wikipedia topic *London Underground* in the section of "Characters" of the article. However, because *underground* has a candidate topic *Underground comix*, which is about self-published comic

[12]http://en.wikipedia.org/wiki/Don_Porter

[13]http://en.wikipedia.org/wiki/Joey_(1985_film)

[14]http://en.wikipedia.org/wiki/The_Vinyl_Underground

DOI: 10.1002/asi

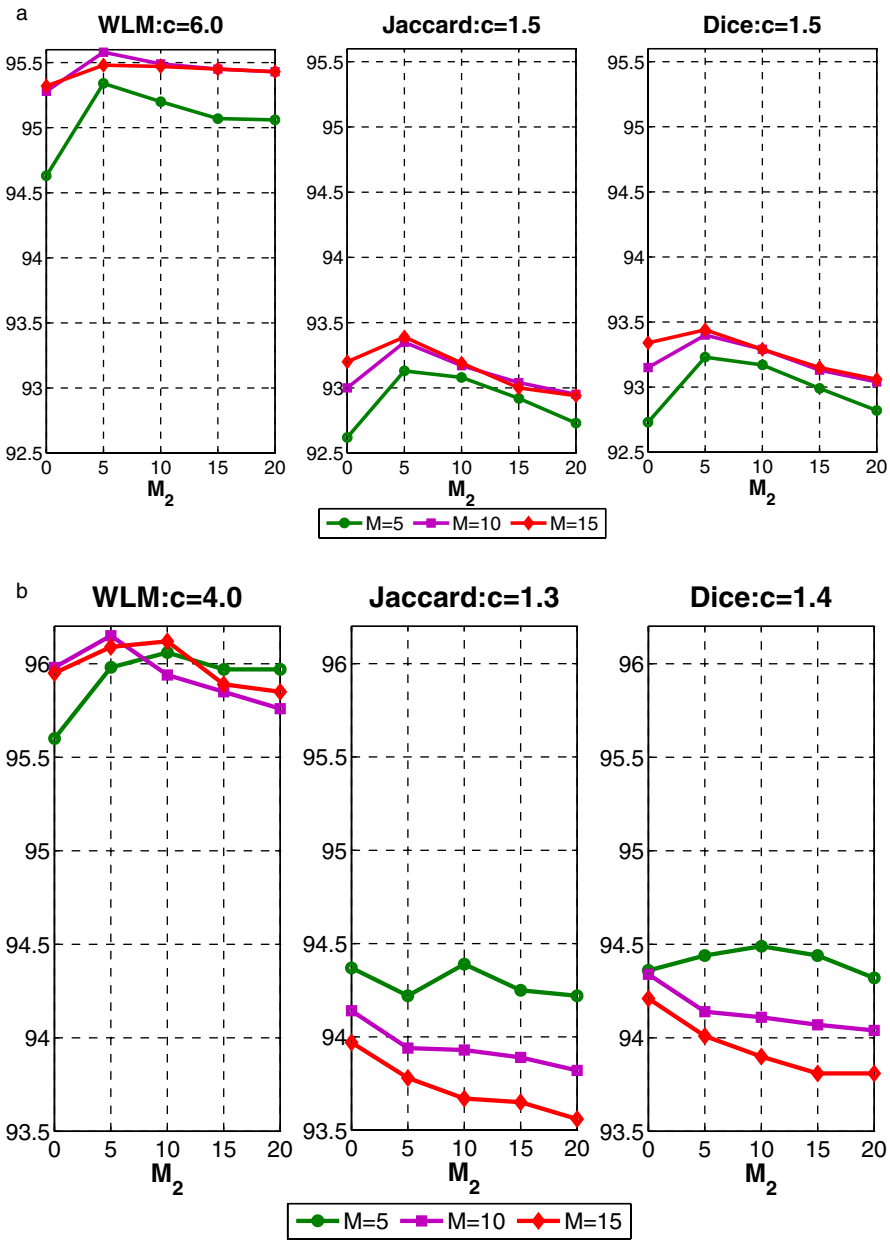

FIG. 6. Accuracy (%) of varying M_2 on the English (a) and Traditional Chinese (b) sets of articles. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

books, TSDW mistakenly disambiguates the keyphrase to topic *Underground comix*. In this case, it is better to take the local context information into account to help disambiguation.

Conclusion and Future Work

Word sense D2W is a key component in many applications in the areas of natural language processing, information retrieval, and others. In this article, we propose an innovative TSDW. In contrast with existing works, TSDW leverages the semantic clue from both unambiguous and ambiguous keyphrases in a given document. The context of the first stage is defined by pruning unimportant and noisy unambiguous keyphrase, leading to a highly efficient and effective disambiguation process for all ambiguous keyphrases. With the confidence measure, the high-quality knowledge provided by the ambiguous keyphrases is recruited as additional contextual information in the second-stage disambiguation. Because the second-stage disambiguation focuses on a small size of ambiguous keyphrases of low confidence, better accuracy is obtained from the additional discriminative knowledge with little extra computation. Extensive experiments are conducted to study the performance of TSDW using data sets in two languages, English and Traditional Chinese, to validate

DOI: 10.1002/asi

its generalizability. Experimental results show that TSDW generalizes well to different languages and measures, and achieves better disambiguation accuracy with lower computation than state-of-the-art approaches. Despite its promising performance, there is still room for improvement. As discussed in error analysis, a specific relatedness measure may have its limitations in distinguishing the correct topic in some cases. An alternative is to apply a relatedness measure of a different aspect, or a combination of multiple relatedness measures in the second-stage disambiguation. In addition, the local context of the ambiguous keyphrase may also be incorporated into the TSDW framework for better disambiguation accuracy.

References

Agirre, E., & de Lacalle, O.L. (2009). Supervised domain adaption for wsd. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09) (pp. 42–50). Stroudsburg, PA: Association for Computational Linguistics.

Agirre, E., & Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09) (pp. 33–41). Stroudsburg, PA: Association for Computational Linguistics.

Bunescu, R., & Pasca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In Proceedings of 11st Conference of the European Chapter of the Association for Computational Linguistics (EACL '06) (pp. 9–16). Stroudsburg, PA: Association for Computer Linguistics.

Cho, H., Chen, M., & Chung, S. (2010). Testing an integrative theoretical model of knowledge-sharing behavior in the context of wikipedia. Journal of the American Society for Information Science and Technology, 61(6), 1198–1212.

Chrupala, G., & Klakow, D. (2010). A named entity labeler for German: Exploiting Wikipedia and distributional clusters. In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC '10), Valletta, Malta. European Language Resources Association.

Cilibrasi, R.L., & Vitanyi, P.M.B. (2007, March). The Google similarity distance. IEEE Transactions on Knowledge and Data Engineering, 19(3), 370–383.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on Wikipedia data. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '07), Prague, Czech Republic (pp. 708–716). Association for Computational Linguistics.

Dice, L.R. (1945, July). Measures of the amount of ecologic association between species. Ecology, 26(3), 297–302.

Gabrilovich, E., & Markovitch, S. (2006). Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI '06) (pp. 1301–1306). Palo Alto, CA: AAAI Press.

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI '07) (pp. 1606–1611). San Francisco, CA: Morgan Kaufmann Publishers Inc.

Giles, J. (2005, Dec 14). Internet encyclopaedias go head to head. Nature, 438(7070), 900–901.

Grineva, M., Grinev, M., & Lizorkin, D. (2009). Extracting key terms from noisy and multitheme documents. In Proceedings of the 18th International Conference on World Wide Web (WWW '09) (pp. 661–670). New York, NY: ACM.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I.H. (2009). The weka data mining software: An update. SIGKDD Explorations Newsletter, 11(1), 10–18.

Han, X., & Zhao, J. (2009). Named entity disambiguation by leveraging Wikipedia semantic knowledge. In Proceeding of the 18th ACM Conference on Information and Knowledge Management (CIKM '09) (pp. 215–224). New York, NY: ACM.

Hearst, M.A. (1992). Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th Conference on Computational Linguistics (COLING '92) (pp. 539–545). Morristown, NJ: Association for Computational Linguistics.

Homma, S., Aikawa, K., & Sagayama, S. (1997). Improved estimation of supervision in unsupervised speaker adaptation. In Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97). Washington, DC: IEEE Computer Society.

Hu, X., Zhang, X., Lu, C., Park, E.K., & Zhou, X. (2009). Exploiting Wikipedia as external knowledge for document clustering. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09) (pp. 389–396). New York, NY: ACM.

Huang, L., Milne, D., Frank, E., & Witten, I. H. (2012). Learning a concept-based document similarity measure. Journal of the American Society for Information Science and Technology, 63(8), 1593–1608.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. Bulletin del la Société Vaudoise des Sciences Naturelles, 37, 547–579.

Kassner, L., Nastase, V., & Strube, M. (2008). Acquiring a taxonomy from the German Wikipedia. In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC '08), Marrakech, Morocco. European Language Resources Association.

Knuth, D.E. (1998). The art of computer programming, Vol. 3: Sorting and searching (2nd ed.). Redwood City, CA: Addison Wesley Longman Publishing Co.

Komachi, M., Kudo, T., Shimbo, M., & Matsumoto, Y. (2008). Graph-based analysis of semantic drift in espressolike bootstrapping algorithms. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08) (pp. 1011–1020). Stroudsburg, PA: Association for Computational Linguistics.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC '86) (pp. 24–26). New York, NY: ACM.

Li, C., Sun, A., & Datta, A. (2011). A generalized method for word sense disambiguation based on Wikipedia. In Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR '11) (pp. 653–664). Berlin/Heidelberg, Germany: Springer-Verlag.

Li, Y., Huang, K., Tsuchiya, S., Ren, F., & Zhong, Y. (2008). Exploring words with semantic correlations from Chinese Wikipedia. In Intelligent Information Processing IV, 5th IFIP International Conference on Intelligent Information Processing (pp. 103–108). New York, NY: Springer.

Malo, P., Sinha, A., Wallenius, J., & Korhonen, P. (2011). Concept-based document classification using Wikipedia and value function. Journal of the American Society for Information Science and Technology, 62(12), 2496–2511.

Medelyan, O., Witten, I.H., & Milne, D. (2008). Topic indexing with Wikipedia. In Proceedings of the First AAAI Workshop on Wikipedia and Artificial Intelligence (pp. 19–24). Palo Alto, CA: AAAI Press.

Mihalcea, R. (2005). Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT '05) (pp. 411–418). Stroudsburg, PA: Association for Computational Linguistics.

Mihalcea, R., & Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. In Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM '07) (pp. 233–242). New York: ACM.

DOI: 10.1002/asi

Milne, D., & Witten, I.H. (2008a). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI'08). Palo Alto, CA: AAAI Press.

Milne, D., & Witten, I.H. (2008b). Learning to link with Wikipedia. In Proceeding of the 17th ACM Conference on Information and Knowledge Management (CIKM '08) (pp. 509–518). New York, NY: ACM.

Nigam, K., McCallum, A.K., Thrun, S., & Mitchell, T. (2000, May). Text classification from labeled and unlabeled documents using EM. Machine Learning, 39(2-3), 103–134.

Ratinov, L.-A., Roth, D., Downey, D., & Anderson, M. (2011). Local and global algorithms for disambiguation to Wikipedia. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL '11) (pp. 1375–1384). Stroudsburg, PA: Association for Computational Linguistics.

Scaiella, U., Ferragina, P., Marino, A., & Ciaramita, M. (2012). Topical clustering of search results. In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM '12) (pp. 223–232). New York, NY: ACM.

Shirakawa, M., Nakayama, K., Aramaki, E., Hara, T., & Nishio, S. (2010). Relation extraction between related concepts by combining Wikipedia and web information for Japanese language. In Proceedings of 6th Asia Information Retrieval Societies Conference (AIRS '10) (pp. 310–319). New York: Springer.

Strube, M., & Ponzetto, S.P. (2006). Wikirelate! computing semantic relatedness using Wikipedia. In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI '06) (pp. 1419–1424). Palo Alto, CA: AAAI Press.

Stvilia, B., Twidale, M.B., Smith, L.C., & Gasser, L. (2008). Information quality work organization in Wikipedia. Journal of the American Society for Information Science and Technology, 59(6), 983–1001.

Su, J., Shirab, J.S., & Matwin, S. (2011, June). Large scale text classification using semi-supervised multinomial naive bayes. In L. Getoor & T. Scheffer (Eds.), Proceedings of the 28th International Conference on Machine Learning (ICML '11) (pp. 97–104). New York, NY: ACM.

Tamagawa, S., Sakurai, S., Tejima, T., Morita, T., Izumi, N., & Yamaguchi, T. (2010). Learning a large scale of ontology from Japanese Wikipedia. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'10) (pp. 279–286). Washington, DC: IEEE Computer Society.

Turdakov, D., & Velikhov, P. (2008). Semantic relatedness metric for Wikipedia concepts based on link analysis and its application to word sense disambiguation. In Proceedings of the SYRCODIS 2008 Colloquium on Databases and Information Systems, St. Petersburg, Russia (Vol. 355, pp. 35–40).

Wang, P., & Domeniconi, C. (2008). Building semantic kernels for text classification using Wikipedia. In Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08) (pp. 713–721). New York, NY: ACM.

Willett, D., Worm, A., Neukirchen, C., & Rigoll, G. (1998). Confidence measures for hmm-based speech recognition. In Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98) (pp. 3241–3244). Sydney, Australia: International Speech Communication Association.

Yeh, E., Ramage, D., Manning, C.D., Agirre, E., & Soroa, A. (2009). Wikiwalk: Random walks on Wikipedia for semantic relatedness. In Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (pp. 41–49). Stroudsburg, PA: Association for Computational Linguistics.

Yoshida, M., Ikeda, M., Ono, S., Sato, I., & Nakagawa, H. (2010). Person name disambiguation by bootstrapping. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10) (pp. 10–17). New York, NY: ACM.

Zesch, T., Gurevych, I., & Mühlhäuser, M. (2007). Comparing Wikipedia and German wordnet by evaluating semantic relatedness on multiple datasets. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '07) (pp. 205–208). Stroudsburg, PA: Association for Computational Linguistics.

DOI: 10.1002/asi

Appendix

Keyphrase Recognition

Let $k = w_1 w_2 \ldots w_m$ be a keyphrase of length m, and $K = \{k\}$ be all keyphrases in the Wikipedia inventory. We group the keyphrases based on their prefix words, so that each group K_w is a set of keyphrases with the same prefix word: $K_w = \{k | k = w_1 w_2 \ldots w_m, w_1 = w, m \geq 1\}$. Then, any group K_w can be accessed in constant time by looking up a hashtable with the key being the prefix word w. For each group K_w, we build a prefix tree for all keyphrases within this group. Specifically, given a keyphrase in K_w, we start creating a path from the root, where the node at level i denotes the word at position $i + 1$ of the keyphrase. The root of the tree is the prefix word w at level 0. Each node has a boolean mark indicating whether it is the last word of some keyphrases, that is, the path from the root to the node constitutes a keyphrase in K_w. In this way, we build an index for the Wikipedia inventory as a forest, where each prefix tree of the forest corresponds to a keyphrase group K_w with the same prefix word w. We call such a prefix tree a *keyphrase tree*.

ALGORITHM A1. Keyphrase recognition.

```
input :
        A text: t = w1 w2 ... wn;
        A hash table: f(w, tree) indexes the forest of Wikipedia inventory;
output:
        A set of matched key phrases 𝕊;
1  𝕊 = {}; m = 1; // m is either 1 or the length of last recognized key phrase
2  for i = 1; i <= n; i+ = m do
3      w = w_i;
4      t_w = f.get(w); // look up the prefix tree associated with word w
5      s = '';
6      pre = '';
7      if t_w ! = null then
8          node = t_w.root();
9          pre.append(w)
10         if node.lastword() then
11             s = pre.string();
12         for j = i + 1; j <= n; j + + do
13             w = w_j;
14             node = node.child(w);
15             if node ! = null then
16                 pre.append(node.word());
17                 if node.lastword() then
18                     s = pre.string(); // store the longest key phrase matched so far
19             else
20                 break; // no further word can be matched
21         if s ! = '' then
22             𝕊.add(s); m = s.length(); // skip all words of the matched key phrase
23         else
24             m = 1;
25     else
26         m = 1;
27 return 𝕊;
```

Figure A1 illustrates an example keyphrase tree based on the prefix word *Java*. In this figure, the tree contains nine keyphrases: *Java*, *Java virtual machine*, *Java virtual machine heap*, *Java virtual machine tools interface*, *Java sdk*, *Java speech api*, *Java speech api markup language*, *Java speech markup language*, and *Java swing*. Algorithm A1 outlines the keyphrase recognition algorithm using the Wikipedia inventory. Given an input text, we process the recognition word after word. If the word at position i of the input text is w_i, the corresponding keyphrase tree is looked up

DOI: 10.1002/asi

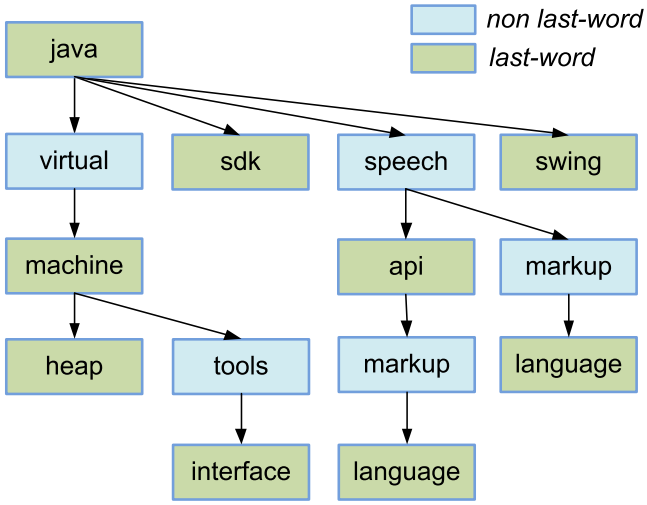

FIG. A1. Example of keyphrase tree of initial word *Java*. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

via a hash table for key w_i (line 4). If no such tree exists, we skip the current word and proceed to the next word (lines 7, 26). When such a keyphrase tree exists, we dive into the tree and identify the matched child node with the next word w_{i+1} (lines 13–14). The child node search operation can be realized efficiently by using a hash table or a binary tree data structure. If a child node matches w_{i+1}, the search process continues to match the next word w_{i+2}. During the search process, if a node is marked as the last word, we store the corresponding keyphrase in a variable s (lines 17–18). Then variable s contains the longest keyphrase identified so far (along some path of the keyphrase tree). The recognition process continues until we reach the leaf node of the path or fail to match the next word with any child node (lines 13–24). After the search process terminates, the keyphrase contained in s is returned as the recognized longest keyphrase (line 22). If a keyphrase with length m is recognized, the recognition process continues with the word at position $i + m$ (line 22). The algorithm has a complexity of $O(n)$, where n is the length of the input text in number of words.

DOI: 10.1002/asi