Disambiguating Words Senses with the Aid of Wikipedia

Abdullah Bawakid[1], Mourad Oussalah[2], Naveed Afzal[1], Seong-O Shim[1], Syed Ahsan[1]
[1]Faculty of Computing and Information Technology – North Jeddah, King Abdulaziz University, Jeddah, Saudi Arabia
[2]School of Engineering, Department of Electronic, Electrical and Computer Engineering, University of Birmingham, Birmingham, United Kingdom
[1]{ abawakid } @kau.edu.sa

Abstract: In this paper, a novel framework for extracting and using features from Wikipedia for the task of Word Sense Disambiguation is presented. We highlight how the features are extracted, re-organized and applied for building what we call term-concepts table. We utilize the internal structure within Wikipedia such as its categories structure and inter links while building the proposed framework. We describe an evaluation we ran on the built framework to test its effectiveness in the application of Disambiguating Word Senses. We also report the obtained results and compare them with those of other competing systems.
[Bawakid A, Oussalah M, Afzal N, Shim S, Ahsan S. **Disambiguating Word Senses with the Aid of Wikipedia**. *Life Sci J* 2013;10(2): 1414-1426] (ISSN:1097-8135). http://www.lifesciencesite.com.

Keywords: Disambiguating Word Senses, Wikipedia, Features Extraction, WSD, Links Analysis, Strong Links

1. Introduction

With the increasing expansion of the web, the effort applied in the field of Information Extraction (IE) to draw structured information from semi-structured or unstructured documents continued to expand. Word Sense Disambiguation (WSD) is among the most heavily investigated areas in the IE domain. The need for WSD is highlighted when inspecting text documents that often include references to different entity types. These entities can sometimes be difficult to trace by information processing systems. For illustration, the word *Fall* may refer to multiple meanings such as the autumn season, the academic semester, or the downward movement due to earth gravity. The right meaning for such a word is usually determined by the context it exists in. In previous research, it was suggested that the right sense for a polysemous word is chosen by the surrounding words accompanying it (Mihalcea 2007a; Patwardhan et al. 2007). In the work we describe in this paper, we follow a similar hypothesis by relying on the surrounding words to help decide the correct sense of a target term.

The problem of Word Sense Disambiguation (WSD) has been extensively researched in the NLP community (Lesk 1986; Patwardhan et al. 2007; Turdakov and Velikhov 2008). In general, it is assumed for WSD that all entity mentions would have a match in a reference dictionary. The most common dictionaries used for matching words senses are WordNet and Wikipedia. With the framework we present in this paper, we build our dictionary by relying on Wikipedia.

In this paper, we devise a novel Wikipedia-based semantic relatedness measure employing both the content and also the structure of Wikipedia for WSD. The work we present here mainly differs from many of the previously developed systems in the literature in that it uses unsupervised methods selecting and also ranking the best suitable senses for a target word. After constructing the system, we present an evaluation on the system output and compare the performance of different runs of the system we implemented.

This paper is organized as follows: In section 2 we give an overview on related work. Section 3, we briefly describe the Wikipedia-based framework we constructed and how we extracted features from Wikipedia for usage in in our system. In section 4, we show the evaluation performed on different runs of our system and compare their performance. In the

last section, we summarize the findings of this paper and conclude it.

2. Related Work and Discussion

A large variety of WSD methodologies were proposed in the literature. Some methods are knowledge-based employing external knowledge sources and dictionary definitions. Lesk (Lesk 1986), for instance, used words dictionary definitions and compared them with the surroundings of the target term to decide what the right sense is. Navigli et al (Navigli and Velardi 2005) employed WordNet and some other lexical resources to build structural sense specifications for each term in a given context. They selected the best candidate by applying a set of rules they predefined in their system. Reference (Patwardhan et al. 2007) employed a relatedness measure that is based on WordNet to compare the context terms of an ambiguous word and all the possible senses for that word. The sense which obtains the highest relatedness score is the chosen right sense.

Other methodologies implemented in other systems are mainly data-driven. They rely on statistical probabilities precomputed from a corpus which is sense-annotated. For example, Gliozzo et al (Gliozzo et al. 2005) exploited supervised kernel methodologies for modeling sense distinctions. In (Wang and Martinez 2006), related disambiguated terms were used for building examples obtained from the web. The examples were afterwards processed for replacing non ambiguous terms with other ambiguous ones. In essence, this provides example contexts for the varying senses of the ambiguous terms. (Diab 2004) employed the idea different translations of an ambiguous word can be generated in different languages. Hence, collections of parallel text were used in their methodologies for annotating the different senses of ambiguous words.

The method proposed in this paper has several advantages when compared with the above-mentioned ones. First, when using a dictionary or a knowledge database, this will in effect cause the system to be limited by the coverage and accuracy of the database or dictionary they rely on. In our work, we chose Wikipedia which is the largest known encyclopedia. Its size is increasing along with its breadth, coverage and accuracy. Second, the method proposed here is monolingual. Hence, the need for parallel data is not necessary. We use the extracted and constructed features from the chosen Wikipedia dump in the built system for as many application processes as needed. When performing the evaluation, we note that the obtained accuracy with our methodologies is comparable, or sometimes exceeding, methodologies similar to the systems mentioned above.

As for systems relying on the usage of Wikipedia, some systems investigated the usage of its links and categories. Mihalca (Mihalcea 2007a) employed a supervised method using internal links for constructing a corpus with annotated senses for the application of WSD. After constructing the corpus, they linked the senses to the definitions existing in WordNet. In (Mihalcea and Csomai 2007), a method was proposed to extract feature vectors for ambiguous words from the links existing in Wikipedia. We employed a similar method for constructing an evaluation corpus of our system.

In (Fogarolli 2009), a method was investigated using the mutual links of Wikipedia. In (Turdakov and Velikhov 2008), several link types were investigated and compared. In our work, we extend these two methodologies through including even more important link types and employing the category structure of Wikipedia to help with fine-tuning the chosen links. We also use a term-concepts table which gives for any given word a more comprehensive and larger list of related articles. The effect of this is to allow for the consideration of senses which are not mentioned explicitly in the disambiguation pages of Wikipedia. In the evaluation results that were run, we illustrate that even with the breadth caused from introducing the test-concept table to the words, the system still obtains relatively good evaluation results.

3. Wikipedia-based Framework

Due to the structure and openness of Wikipedia, it is not possible to use it in its raw condition without having it preprocessed first. In the system we build for extracting semantics from Wikipedia, each Wikipedia article is

viewed as a single concept that is labeled with its title. We build a term-concepts vector that is constructed by using the inner content of the articles along with their titles. In addition, we process the categories and their network within Wikipedia and the various links to help find a measure for computing the relatedness between any two concepts. In the following subsections, we briefly describe the main processes we apply to the downloaded Wikipedia dump for building the needed vectors and extracting the important features.

3.1. Preprocessing Wikipedia

The framework we describe here used the English dump of Wikipedia that was created in 16/03/2010. A series of processes were applied to it for preparing it for the following stages before it is analyzed. These processes include removing unimportant tags in each article such the non-English characters and the edit history of the article. We also discarded too short articles which were less than 100 words in length or those having less number of links than five. In addition, categories that were too generic such as "Years" and "Centuries" were removed. The articles belonging to these generic categories were removed, too. After applying the mentioned operations, we had 1,504,748 articles and 126,709 categories.

3.2. Features Extraction from Wikipedia

After preprocessing the adopted dump of Wikipedia, we extract its main features by analyzing the inner text of each article, the article title, links and categories of Wikipedia. Stop-words are also removed. Then, the remaining term are used for representing the different concepts of Wikipedia by giving each word a weight. For this, we employ the Term Frequency-Inverse Document Frequency (TFIDF) measure (Manning and Schuetze 1999).

3.2.1. Boosted Term-Concepts Table

In essence, we link all the terms within Wikipedia to all concepts by creating a vector for each term. The vector is formed by the l2-normalized TFIDF score of the term within each Wikipedia article. These scores give indication to how much each term contributes to the concepts it is linked to. We build the Term-Concepts table by simply ranking all the concepts linked to a term based on the computed word score in a decreasing order.

After creating an initial term-concepts table, we take advantage of the redirect links existing within Wikipedia by examining the words of the redirect links and the title of the page they link to. In many cases, redirect links contain a word or multiple words that have some degree of relevancy to the pages they point to and yet the words may not exist in the article title or even its inner content. To address this issue, we apply a two-level boosting process to consider the presence of a word or more in a redirect link by changing the initial weight of the concept the link points to in attempt to reflect this redirect link.

The boosting algorithm we implement is shown in Figure 1 in which *W* is the set of words that we desire the most related concepts to. c_t is the concept title and c_s is its score. *C* represents the set (c_t , c_s) for the concepts generated after applying the Term-Concept table stage to *W*. *allC* represents all of the Concepts within Wikipedia while *allR* is for all the redirect links. After we apply the boosting process to the term *Unhappy*, we obtain the results shown in Figure 2.

```
Boost_Concepts( W, C, FirstLevelBoost, SecondLevelBoost, allC, allR){
        HS ← Max( c_s ∈ C) /* Highest Score in the C set */
        FL ← FindFirstLevelConcepts(W, allC, allR)
        FLB ← ϕ
        ∀ (c_t , c_s) ∈ FL, FLB ← FLB ∪ (c_t , (HS * FirstLevelBoost) )
        SL ← FindSecondLevelConcepts(W,C, SecondLevelBoost)
        Res ← C
        Res ← Res ∪FLB ∪SL /* Overwriting repetitions with those in SL and FLB */
        return Res
}

FindFirstLevelConcepts(W, allC, allR) {
  Res ← ϕ
  Loop over (c_t , c_s) ∈ (allC ∪ allR) {
    /* all elements of W are contained within c_t */
    If ((W ≡ c_t) and (len(W)= len(c_t)) Then Res ← Res ∪ (c_t , c_s)
  }
  return Res
}

FindSecondLevelConcepts(W,C) {
  Res ← ϕ
  Loop over (c_t , c_s) ∈ C {
    If ( (∀ w ∈ W) ∃ c' where (c' ≡ w) and (c' ∈ c_t) and (len(W) <> len(c_t)) ) Then
      c_s = c_s * ( ((SecondLevelBoost-1)^(|w|/| c_t |))+1)
      Res ← Res ∪ (c_t , c_s)
    EndIf
  }
  return Res
}
```

Figure 1: The Pseudo code of the boosting process

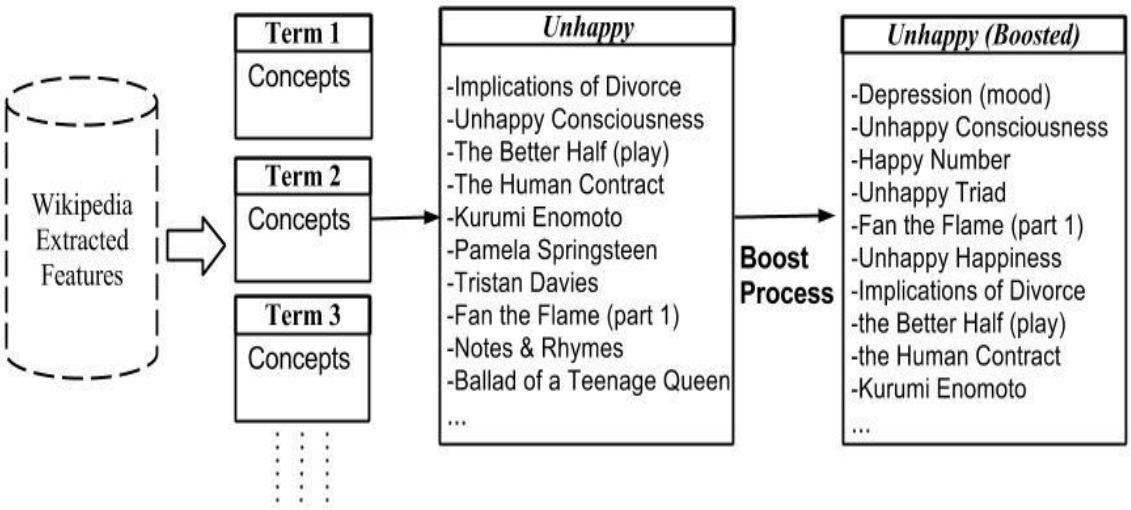

Figure 2: Using redirect links for boosting the term-concepts table

3.2.2. Wikipedia Internal Links and Categories

In this section, we employ the internal hyperlinks within the Wikipedia articles text linking to other Wikipedia articles and also the categories structure within Wikipedia for extracting other useful semantic features. Since not all the links in an article are of the same importance level, we apply a customized filtering module that focuses mainly on links of high quality and attempts to discard those of less significance. This is accomplished by applying a classification scheme on the internal links within Wikipedia articles to segment them into several levels reflecting their importance. The hierarchies of the categories existing in Wikipedia were also employed. In **Error! Reference source not found.**, we show the link types utilized in our system. The links in the figure are sorted in a decreasing order based on the scores they carry. In general, there are three main link categories: First is the mutual links in which the two articles are linking to each other. Second is a single link among two articles where both articles share a parent category (or a grandparent category). Third is the "See Also" links which are manually appended by Wikipedia contributors to many articles. The weights that were given to the varying link types are illustrated Table 1. The scores given to w_9 and w_{10} are for "See Also" links and Inverse See Also links, respectively.

The chosen weights for the different links are meant to show their significance level. To evaluate the effectiveness of the given weights, we employ the extracted features in the task of WSD. We illustrate in the evaluation how the weight of each link type influences the performance of the system.

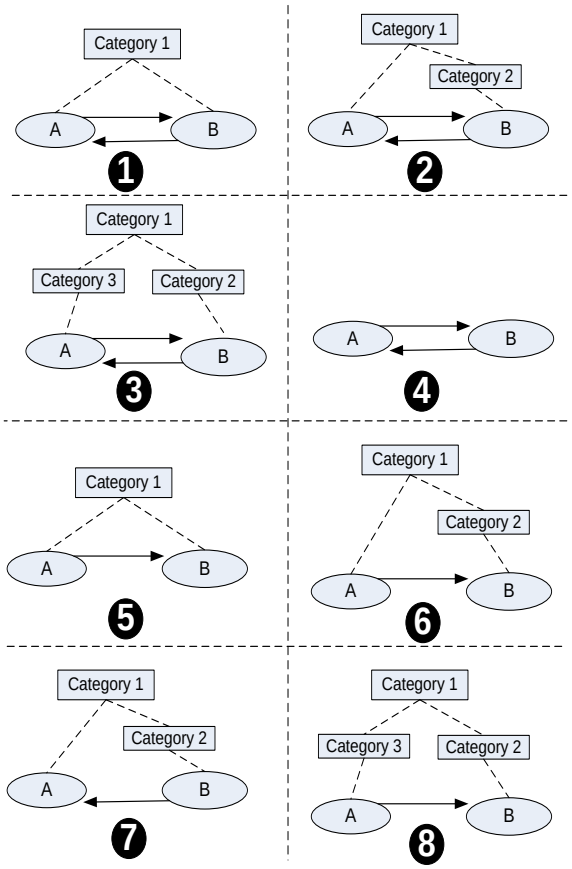

Figure 3: The defined link types sorted in a descending order based on their scores

Table 1: The assigned weights for the chosen links types

Link Type	Weight Assigned	Link Type	Weight Assigned
1	3	6	1.5
2	2.75	7	1.5
3	2.5	8	1.25
4	2.25	9	3.75
5	1.75	10	3.25

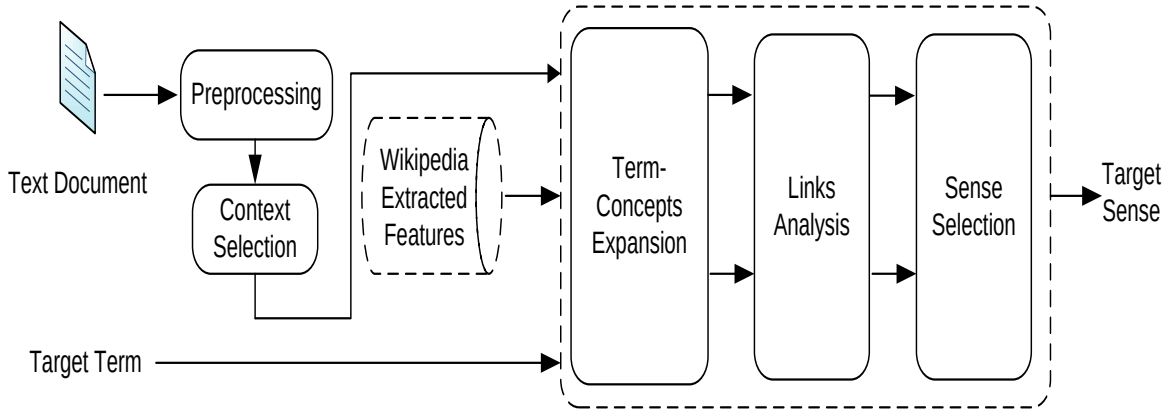

Figure 4: The main processes of the WSD

4. Disambiguating Word Senses

The main goal is to recognize the right sense for a specific word in the context it is found in. We employ the explicit features found in the context and the features we extracted from Wikipedia. In the method we implemented, we applied the term-concepts table to the target term and then assigned a score to each concept based on the strong links analysis. Therefore, we generate a sorted list of concepts with the top being the chosen right sense for the target term. The following subsections briefly describe the main modules of WSD which we show in Figure 4.

4.1. Preprocessing and Specifying the Context

The text document having the target word is first parsed and tokenized. Its stop words are also discarded. Then the target term is marked in the source document and a predetermined number *2n* of its surrounding words are extracted, with *n* words being before and *n* words after the target word. We call the *2n* extracted words the Context Terms and abbreviate them with CT. Thus, elements of CT which are essentially the individual context terms can be labeled with $ct_{i=1...|CT|}$.

4.2. Term-Concepts Expansion

After specifying CT in the previous stage, the term-concepts table gets applied onto CT and the target word to come up with a concept list C_i for each word resulting in a total of |CT| + 1 concepts lists. Each concept list can be represented with the following formula:

$$C_i = \{c_{ij}\}_{j=\{1...V\},i=\{1...|CT|\}} \qquad 1$$

Where *i* is the C_i concept list number, *j* is for specifying individual concepts within the concepts list and *V* is the total number of c_{ij} concepts in the list. As for the target word concepts list *TW*, we represent it with:

$$TW = \{g_k\}_{k=\{1...M\}} \qquad 2$$

Each concept in *TW* is tagged with g_k where *k* is the number of the concept and *M* is the total number of concepts in *TW*.

4.3. Links Analysis and Sense Selection

In this subsection, we illustrate how the devised links can be used to quantify the relatedness score between any two concepts and how this was employed in the task of WSD. We explain different methodologies employing different aspects of the previously extracted features and illustrate their performance in WSD. For quantifying the relatedness between any two articles, say article A and article B, we

assign scores to the hyperlinks present in both articles. We also form two sets of articles S_A and S_B in which the first set has the most relevant articles to the article A and the second set is used for saving the most relevant articles to B. We compute the similarity between the sets S_A and S_B by relying on the cosine distance measure.

4.3.1. Terms Vectors Intersection

With this method we create a vector T that is produced by aggregating both CT and the target term. The new vector can be represented with:

$$T = \{w_y\}_{y=\{1.....|CT|+1\}} \quad 3$$

Assume that the possible senses for the target term are represented with S_{ye} in which y is a unique number used for identifying the target term in the list T while e is the possible sense number. The generic assumption made in the WSD task is that for each possible sense there is a corresponding article that can be matched to that sense in Wikipedia. Therefore, the term-concepts table can be utilized to form the top R representative terms with the highest weights for each sense. The list of the best representative terms for each sense can be represented with:

$$S_{ye} = \{a_{yef}\}_{f=\{1...D\}} \quad 4$$

where a is a representative term in the list S_{ye} and f is an identifying unique number for each term in the list. For each possible concept (or sense) e, we compare its representative words in S_{ye} with T. The chosen sense (or concept) is the one with the largest number of overlaps between S_{ye} and T.

4.3.2. Unweighted Strong Links

In this method, *CT* is also inspected first and T is formed. When creating a list of the most related articles for each possible concept (or sense) of the ambiguous term, we rely on the strong links analysis method but without taking into account the scores of the links. The formed list of the most relevant concepts for each possible sense can be presented with:

$$P_{ye} = \{a_{yef}\}_{f=\{1...Q\}} \quad 5$$

where Q represents the number of concepts depicted from the strong links method. As with the previous methodology, the most related articles for each concept (or sense) are compared with T. The best representative sense is the one having the biggest number of overlaps with T.

4.3.3. Weighted Strong Links

For every concept C_i, an analysis is applied on all the hyperlinks existing within its corresponding article and is compared and evaluated against all *TW* concepts. The comparison is performed by evaluating the concepts list of each article and comparing it against the concepts list of the other. This can be represented with the following:

$$eTW = \{G_k\}_{k=\{1...M\}} \quad 6$$

Where *eTW* is the expanded list formed for *TW*. This formed list contains a list of related articles we call G_k for each possible meaning g_k. The G_k list is formed as:

$$G_k = \{(gc_w, f(g_k, gc_w))\}_{w=\{1...V\}, f(g_k, gc_w)>0} \quad 7$$

where gc_w is a related article to g_k, and $f(g_k,gc_w)$ is basically a function that quantifies the relatedness between the g_k and gc_w by utilizing the strong links method. V represents how many concepts are available after expanding G_k. The same actions are applied to C_i concepts which are grouped together into a single set with the scores of repeated concepts being added together. Thus, we have the following after expanding C_i:

$$eC_i = \{(rc_{ijw}, f(c_{ij}, rc_{ijw}))\}_{j=\{1...V\}, i=\{1...|CT|\}, w=\{1...Q\}, f(c_{ij}, rc_{ijw})>0} \quad 8$$

Where rc_{ijw} is the concept related to c_{ij}, and Q represents the number of related concepts. The score generated by the function $f(c_{ii}, rc_{ijw})$ can be rewritten as tw_{ijw}. We aggregate all of eC_i into a single list called *eC* which also takes into account the repeated concepts as done previously by summing their weights. This results in:

$$eC = \{(rc_v, tw_v)\}_{v=\{1...|D|\}} \quad 9$$

$$tw_v = \sum_{w=1}^{Q} (tw_{ijw}) \text{ , } \textbf{where } \boldsymbol{rw_v = rc_w} \qquad 10$$

Where each rc_v is a unique single concept in eC_i, D is the total number of *unique* concepts in the list eCj, and tw_v is the score given (after the aggregation) to rc_v.

After utilizing the context terms to come up with the eC list, and deriving a list G_k for each sense of the target term, the distance between eC and each G_k is generated with the cosine distance measure. We tag the most representative meaning in the list by selecting the concept numbered k in:

$$\max_k (dist(G_k, eC)) \qquad 11$$

4.4. Evaluation

Selecting a dataset for WSD evaluation is greatly affected by the variations of the system to be tested. For instance, SemEval and Senseval-1/2/3 test collections are built with the aid of WordNet. This makes them difficult to use with our developed system since senses defined in WordNet have to be linked with Wikipedia concepts. This has been known to be a challenging task (Mihalcea 2007b) and needs its own evaluation. Therefore, we chose to come up with our own benchmark which is very similar to the what was selected in (Mihalcea and Csomai 2007; Turdakov and Velikhov 2008; Fogarolli 2009). We used the manually-created links in Wikipedia to build our dataset. Hyperlinks in Wikipedia take the form: [[PartA | PartB]] where PartA is the title of the page pointed to while PartB is the text shown to the article reader. We built a dataset of 1,000 Wikipedia hyperlinks along with the paragraphs containing them. In our performed evaluations, 20 words were selected as the context terms. This selection was based on the experiment applied on the third method we adopted. The results and findings of this experiment are illustrated in Figure 5.

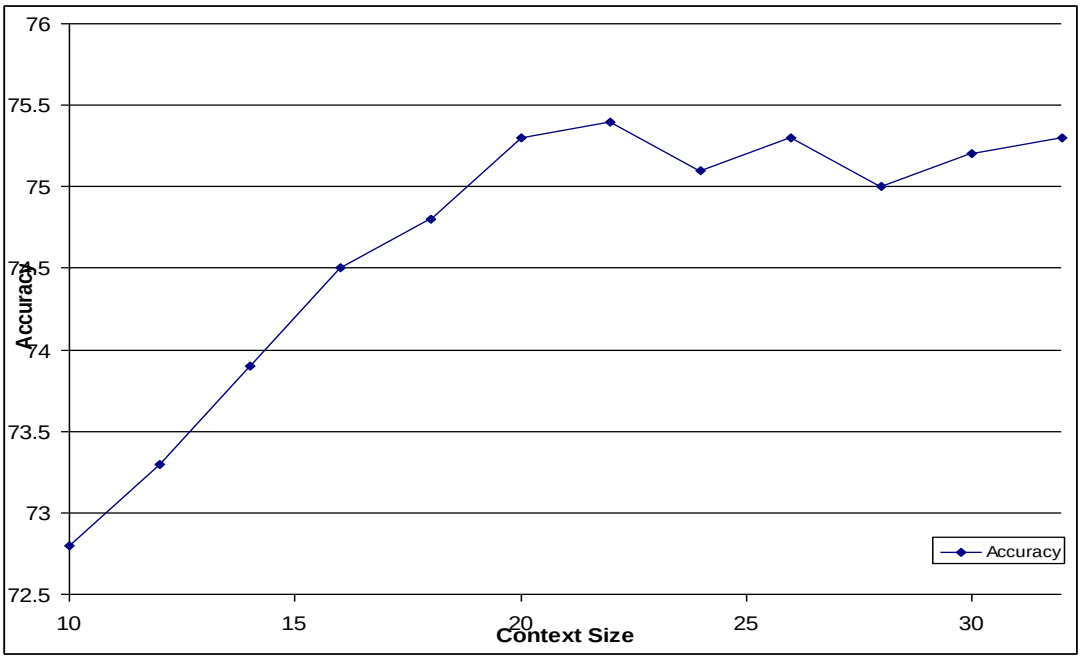

Figure 5: Effects of Varying the Context Size on the Accuracy in WSD after applying the Simple Links Analysis

We also conducted another experiment to select the most optimal weights for the selected strong links. In the experiment, the score of each link type is changed once at a time while keeping the scores for the rest of the links types fixed. The results obtained and their effects on WSD are illustrated in Figure 6 for the weights w_1 to w_5 and in Figure 7 for w_6 to w_{10}. From the reported results, it can be noted that the "See Also" and "Inverse See Also" links give the strongest effect on the system. Overall, the accuracy of the system negatively decreased to 65.91% and 66.41% respectively when we assigned zero to their weights. We found the weakest link type to be single links pointing from an article to another in where both articles share a single grandparent category. The accuracy of the system with this link type decreased to 75.31% when w_8 was set to zero. Changing w_8 to any value above 1.25 had a negative performance effect on the system, too. It was also noted that the effects of the link types number 6 and 7 are so similar. Figure 7 shows that their chart curves almost overlap. Setting the weights to any value above 2.5 for the mutual links of types (1 – 4) leaded to the best overall performance of the system. On the other hand, the performance of the system changed negatively when selecting weights of larger values than 1.75 for all single link types.

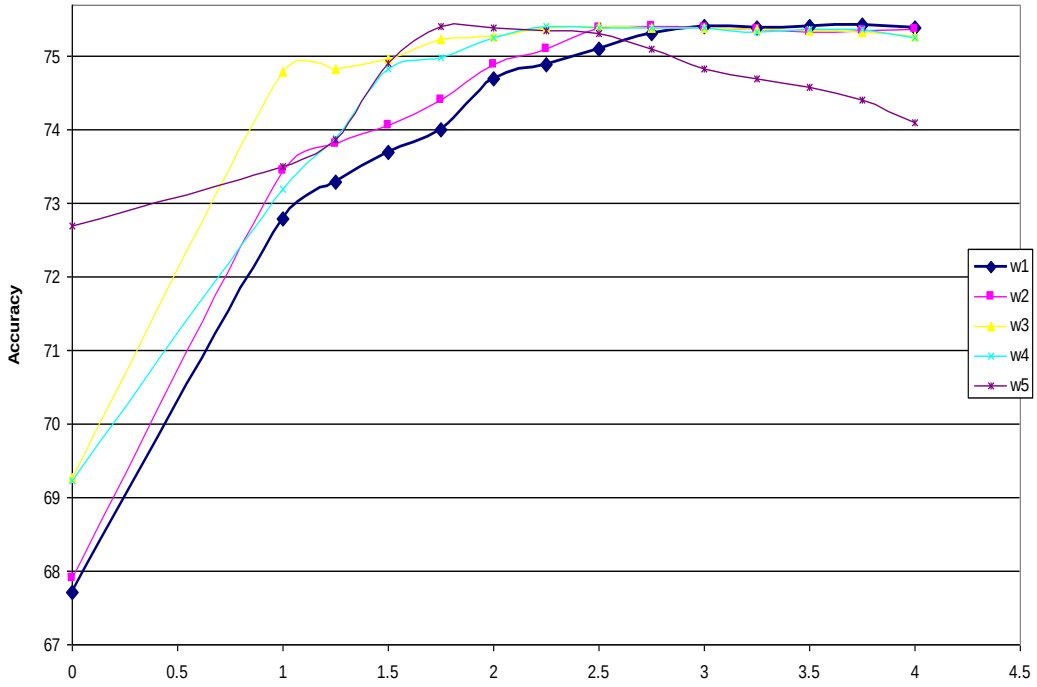

Figure 6: Effects of Varying Strong Links Weights for w_1 to w_5

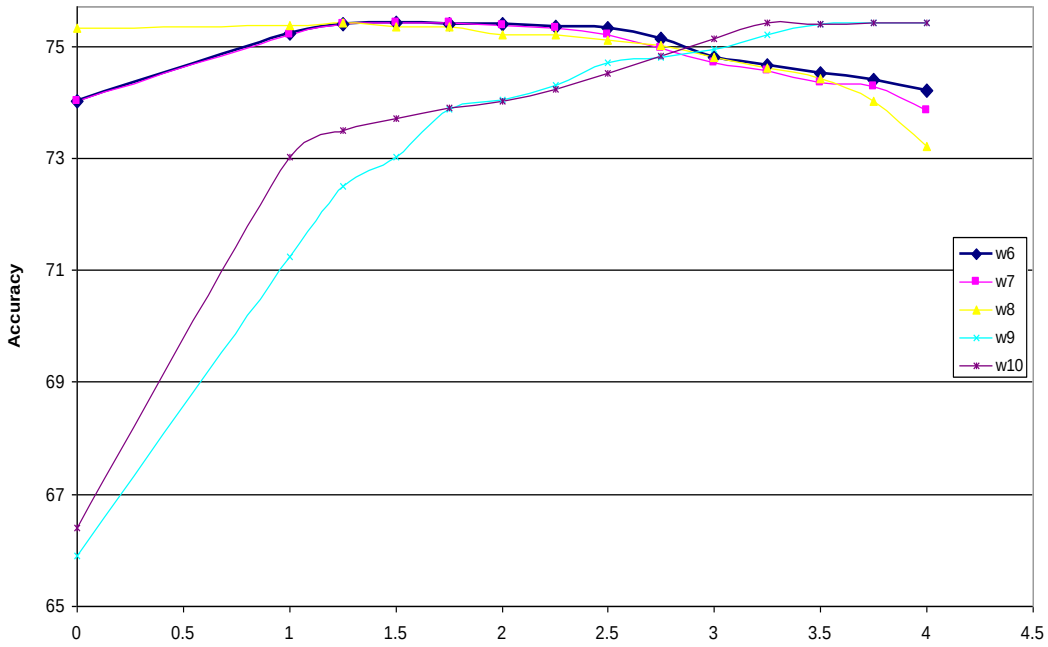

Figure 7: Effects of Varying Strong Links Weights for w_6 to w_{10}

Several experiments were executed on the built dataset to test the explained methodologies in this paper. In particular, we experimented with (1) the Term vectors Intersection method, (2) the Unweighted strong links method, and (3) the Weighted strong links method. The weighted strong links methodology gave the best performance by producing accuracy of 75.41%. The term vectors intersection methodology produced an accuracy of 69.17%. The results produced for all methods are shown in Table 2. We also measured the chances of having the right sense being among the top-2 and 3 senses of those in the sorted senses list generated with our system. The accuracy of our best performing methodology changed positively to 91.82% when considering the top-3 senses.

Table 2: The Accuracies Obtained from All Implemented Methods

	Top	**Top-2**	**Top-3**
Term Vectors Intersection	69.17	75.8	82.71
Unweighted Strong Links	71.84	84.08	87.29
Weighted Strong Links	**75.41**	**87.19**	**91.82**

6. Conclusion

In this paper, we have presented several novel algorithms exploiting both the content and structure of Wikipedia for quantifying the relatedness between any two terms, phrases or sentences. We also explained how to apply the extracted features from Wikipedia, namely the term-concepts table and the strong links method, to the task of disambiguating word senses. We utilized the manually created hyperlinks in Wikipedia as the basis for building the dataset of our evaluation method. Furthermore, we

presented the effects of varying the strong links weights on the overall performance of the system and explained the reasoning behind selecting the weights. The results we had from evaluating the Wikipedia-extracted features in the task of disambiguating word senses illustrate that the strong links method lead to better performance than the term-concepts table. We intend to investigate the implemented methods furthermore in other tasks and applications in our future work.

Acknowledgement
This work was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (968-010-D1433). The authors, therefore, acknowledge with thanks DSR technical and financial support.

Corresponding Author:
Dr. Abdullah Bawakid
Faculty of Computing and Information Technology- North Jeddah
King Abdulaziz University, Jeddah, Saudi Arabia
E-mail: abawakid@kau.edu.sa

References

1. Diab M. Relieving the data acquisition bottleneck in word sense disambiguation. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. Barcelona, Spain: Association for Computational Linguistics; 2004. p. 303
2. Fogarolli A. Word sense disambiguation based on wikipedia link structure. International Conference on Semantic Computing. IEEE; 2009. p. 77–82.
3. Gliozzo A, Giuliano C, Strapparava C. Domain kernels for word sense disambiguation. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Ann Arbor, Michigan: Association for Computational Linguistics; 2005. p. 403–10.
4. Lesk M. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. Proceedings of the 5th Annual International Conference on Systems Documentation. Toronto, Ontario, Canada: ACM; 1986. p. 26, 24.
5. Manning CD, Schuetze H. Foundations of Statistical Natural Language Processing. 1st ed. Cambridge, US: The MIT Press; 1999.
6. Mihalcea R. Using Wikipedia for Automatic Word Sense Disambiguation. Proceedings of the Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics. Rochester, New York; 2007a.
7. Mihalcea R. Using Wikipedia for Automatic Word Sense Disambiguation. Proceedings of the Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics. Rochester, New York; 2007b.
8. Mihalcea R, Csomai A. Wikify!: linking documents to encyclopedic knowledge. CIKM '07: Proceedings of the 16th ACM Conference on Information and Knowledge Management. Lisbon, Portugal: ACM; 2007. p. 242, 233.
9. Navigli R, Velardi P. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. IEEE Trans Pattern Anal Machine Intell. 2005 Jul;27(7):1075–86.
10. Patwardhan S, Banerjee S, Pedersen T. UMND1: unsupervised word sense disambiguation using contextual semantic relatedness. Proceedings of the 4th International Workshop on Semantic Evaluations [Internet]. Prague, Czech Republic: Association for Computational Linguistics; 2007. p. 390–3.

11. Turdakov D, Velikhov P. Semantic relatedness metric for Wikipedia concepts based on link analysis and its application to word sense disambiguation. Colloquium on Databases and Information Systems (SYRCoDIS). CEUR-WS; 2008.

12. Wang X, Martinez D. Word sense disambiguation using automatically translated sense examples. Proceedings of the International Workshop on Cross-Language Knowledge Induction. Trento, Italy: Association for Computational Linguistics; 2006. p. 45–52.

29/7/2013