A Random Graph Walk based Approach to Computing Semantic Relatedness Using Knowledge from Wikipedia

Ziqi Zhang[a], Anna Lisa Gentile[b], Lei Xia[c], José Iria[d], Sam Chapman[e]

[a] Department of Computer Science, University of Sheffield, UK
[b] Department of Computer Science, University of Bari, Italy
[c] Archaeology Data Service, University of York, UK[1]
[d] IBM Research, Zurich[1], [e] K-Now, UK

E-mail: z.zhang@dcs.shef.ac.uk, al.gentile@di.uniba.it, lx535@york.ac.uk, jir@zurich.ibm.com,
sam.chapman@k-now.co.uk

Abstract

Determining semantic relatedness between words or concepts is a fundamental process to many Natural Language Processing applications. Approaches for this task typically make use of knowledge resources such as WordNet and Wikipedia. However, these approaches only make use of limited number of features extracted from these resources, without investigating the usefulness of combining various different features and their importance in the task of semantic relatedness. In this paper, we propose a random walk model based approach to measuring semantic relatedness between words or concepts, which seamlessly integrates various features extracted from Wikipedia to compute semantic relatedness. We empirically study the usefulness of these features in the task, and prove that by combining multiple features that are weighed according to their importance, our system obtains competitive results, and outperforms other systems on some datasets.

1. Introduction

Measuring semantic relatedness between words or phrases is a fundamental process to many Natural Language Processing (NLP) applications, such as Word Sense Disambiguation (WSD) (Patwardhan et al., 2003), spelling correction (Budanitsky and Hirst, 2006), and information retrieval (Finkelstein et al., 2002). Determining relatedness between words or concepts is not an easy task, since it involves reasoning and understanding of context and background knowledge. This has long been recognized as a major challenge in artificial intelligence. Most studies on semantic relatedness make use of WordNet (Fellbaum, 1998) as the background knowledge resource. However, it has been argued that manually building and maintaining resources like WordNet is time-consuming and expensive (Bollegala et al., 2007; Zesch et al., 2008). Furthermore, WordNet suffers from lack of coverage for named entities and specialized concepts that can be crucial to the understanding of concepts. (Strube and Ponzetto, 2006; Bollegala et al., 2007)

Recent research has focused on using Wikipedia[2], which is the largest knowledge resource that is built and maintained in a collaborative manner, in the task of computing semantic relatedness. Although these methods have achieved better results than WordNet-based approaches, the major limitation is that most of these simply adapt WordNet-based methods to a different knowledge resource, the Wikipedia (Strube and Ponzetto, 2006; Zesch et al. 2008a); and that they make use of only one or two types of information content and structural elements (features) in Wikipedia without investigating the usefulness of other features, or the importance of these features in the task of semantic relatedness. However, we argue that using diverse and rich sets of features that are weighed according to their importance produces better representation of an object, which has proved useful in many other NLP tasks such as Named Entity Recognition (Guo et al., 2005), document classification (Benjamin et al., 2008) and information retrieval (Jin et al., 2005). Therefore, in this paper we propose a method for computing semantic relatedness by integrating various features extracted from Wikipedia through a random graph walk model (Lovász, 1993), and study the effects of different features and their importance for the semantic relatedness task.

Our contributions are two-fold. First, we propose a new way of computing relatedness of words or concepts that naturally combines various features using a random walk model. Compared to state-of-the-art systems we achieve competitive results. To test the applicability of the method in complex NLP problems, we apply it to a Named Entity Disambiguation (NED) experiment and achieve better results than one of the best system reported in (Cucerzan, 2007). Second, we present in-depth analysis of the importance of various features and effects of combining them in the task of computing semantic relatedness. This analysis provides useful reference to future development of such systems.

The remainder of this paper is organized as follows: in Section 2 we review related work. Section 3 describes our methodology. Section 4 presents our experiments and findings on feature tuning. Section 5 presents discussion, and finally we conclude the paper.

2. Related Work

Semantic relatedness between words or concepts measures how much two words or concepts are related by encompassing all kinds of relations between them, such as hypernymy, hyponymy, meronymy, antonymy and functional relations. Semantic relatedness is more general

[1] This author carried out this work while being a member of the Department of Computer science, University of Sheffield

[2] http://www.wikipedia.org/

than semantic similarity, which quantifies relatedness by only using hypernymy and hyponymy relations (Strube and Ponzetto, 2006). Computational applications typically require relatedness rather than similarity (Budanistsky and Hirst, 2006).

Approaches to measuring semantic relatedness between words or concepts typically make use of structural elements and information content extracted from lexical resources. Among these many exploit WordNet. Leacock and Chodorow (1998) compute similarity of two words using the path length by following the hypernymy and hyponymy relations in WordNet. The path length is then normalized by the depth of the taxonomy in which the words are found. Resnik (1995) argues that the similarity between two concepts is "the extent to which they share information in common". Specifically using WordNet, this is determined by the relative position of the lowest concept in the hierarchy that subsumes both concepts. Banerjee and Pedersen (2003) measure relatedness of concepts by text overlap of their WordNet glosses. Hughes and Ramage (2007) apply a random graph walk model to compute semantic similarity of two words by exploiting the relations between the words extracted from WordNet. Despite the popularity of WordNet, it has been criticized for its lack of coverage for named entities and specialized concepts which are crucial to domain-specific problems. (Strube & Ponzetto, 2006; Bollegala et al., 2007). Additionally, maintaining the resource up-to-date can be difficult since word senses evolve over time (Bollegala et al., 2007; Zesch et al., 2008a).

Recent years have witnessed the flourishing of collaborative knowledge resources (Zesch et al., 2008b) as the result of the exponential growth of the web and the popularity of Web 2.0 technologies. One of these is the Wiktionary, a multi-lingual free content dictionary covering over 270 languages and over 5 million entries inter-linked with semantic relations posing a strong competition against WordNet. In the domain of semantic relatedness, many have piloted the studies of using Wiktionary as an alternative to WordNet. Zesch et al. (2008a) port several WordNet-based approaches to Wiktionary by replacing WordNet features (e.g., path-length, content vector) with corresponding equivalent or similar contents or structural elements in Wiktionary. On seven test datasets, their method achieved better results than similar approaches that use WordNet. Weale et al. (2009) apply a page rank based algorithm to compute relatedness of words using Wiktionary, and tested their method in a synonym detection task. Müller and Gurevych (2009) use knowledge in Wiktionary to improve information retrieval systems. Essentially, their approach is based on aggregating the semantic relatedness between of each query and document term pair. Texts of Wiktionary entries of words are used to build representational concept vectors. When compared against a standard Lucene[3] search, their methods achieved better results on many datasets. However, being a similar word-based knowledge resource to WordNet, Wiktionary does not overcome the limitation that it has little or no coverage of specialised concepts or named entities, which may hinder its application to domain-specific NLP tasks.

In contrast, contents in Wikipedia are mostly organised based on named entities (Kazama & Torisawa, 2007). For this reason, it has attracted more attention from researchers of semantic relatedness. Strube and Ponzetto (2006) adapt several WordNet-based approaches including that of Leacok and Chodorow (1998), Wu and Palmer (1994), Resnik (1995), and Banerjee and Pedersen (2003) to make use of Wikipedia. For each approach, they identify the feature in Wikipedia that is equivalent to that used within WordNet, and demonstrate that the same approaches applied to Wikipedia achieve better results than using WordNet on two of the experimental datasets they used. Similarly, Zesch et al. (2008a) use the category structure as feature in a path-length based technique; they then use words in an article and words from the first paragraph of an article as features in gloss-based techniques. The main limitation of these methods is that they only make use of one or two types of features, and the choice of features are limited by the methods adapted from WordNet-based approaches. In contrast, we believe that other information content and structural elements in Wikipedia can be also useful for the semantic relatedness task; and that combining various features in an integrated model and studying their importance in the semantic relatedness task is crucial for improving performance.

Turdakov and Velikhov (2008), in their work on applying semantic relatedness in the WSD context, analyze different types of links in Wikipedia and use those features with different weights. However their method heavily relies on heuristics, and they do not evaluate the system in the semantic relatedness task. Gabrilovich and Markovitch (2007) represent each Wikipedia concept using a weighed vector of words that occur in the corresponding article, and then build an inverted index that map each word into a list of concepts in which it appears. Thus to compute relatedness between two texts, a weighted vector of Wikipedia concepts is built for each text by aggregating the concept vectors of each word retrieved from the index. The vectors are then inputted to the cosine metric to derive a similarity score. Hassan and Mihalcea (2009) improve on Gabrilovich and Markovitch's approach by introducing several modifications, which include replacing the cosine similarity function with a different metric, taking into account the length of articles, and then placing more importance on category-type concepts. Essentially these methods treat content in a Wikipedia article as bag-of-words, and do not make use of the structural elements in Wikipedia. In addition, their approaches require pre-computing the inverted index of the entire Wikipedia knowledge resource. Given the size and growth of Wikipedia[4], producing and maintaining an

[3] http://lucene.apache.org/java/docs/

[4] http://en.wikipedia.org/wiki/Wikipedia:Modelling_Wikipedia%27s_growth

up-to-date index of such kind can be computationally expensive.

3. Methodology

We propose a novel method for computing semantic relatedness between words or phrases to address the aforementioned limitations. Most importantly, we study various features extracted from Wikipedia and their importance in the semantic relatedness task. Our approach consists of three steps. First, given a pair of words or phrases, which we will refer to as *surfaces*, we retrieve the relative Wikipedia pages describing the *concepts* represented by the *surfaces*. Next, we extract various information content and structural elements from these pages as features to represent the *concept*. Finally, we transform these *concepts*, their features and relations into a graph representation, and apply a random graph walk model to combine the effects of features to derive a relatedness score. In the following, we describe each step in details.

3.1 Page retrieval

We query Wikipedia using a *surface* to retrieve relevant pages. If a *surface* matches an entry in Wikipedia, a Wikipedia page will be returned. If the *surface* has one sense defined in Wikipedia, or the most often used common sense, a single page describing the *concept* that matches the *surface* is returned. In this case, we refer to this page as the *sense page* for the *concept*. Alternatively a disambiguation page may be returned if the *surface* has several senses defined in Wikipedia. Such a page lists different senses as links to other pages. In Strube and Ponzetto (2006), the authors adopt a heuristic to disambiguate senses by selecting the ones (and the corresponding *sense page*) whose surface texts share a same label marked by brackets. For example, for the pair of words "king" and "rook" both of which return disambiguation pages, they choose the senses for which the link share a common label, as "king(chess)" and "rook(chess)". When such information is not available, they choose the first sense on the disambiguation page. Contrary to their method, we do not apply any heuristics for disambiguation. Instead, we follow every link and keep all *sense pages*, and expect our feature extraction and the semantic relatedness algorithm to naturally select the most related senses for the input pair of *surfaces*. Returning to the previous example, suppose we have only two senses for the *surface* "king" listed on the disambiguation page, which refer to "King, the title for the head of state" and "king (chess), a chess piece"; likewise two senses for the *surface* "rook" are listed on disambiguation page as "rook, a chess piece" and "rook, a family of birds". We represent all four *concepts* using features extracted from their sense pages, and expect our algorithm to produce maximum relatedness score for the *concept* pair "king (chess), a chess piece" and "rook, a chess piece" rather than other combinations, and we use this maximum score as the final relatedness score for the input *surface* pair "king" and "rook". The way to achieve this will be detailed in section 3.3.

3.2 Feature extraction using Wikipedia

Once we have identified relevant *concepts* and their *sense pages* for the input *surfaces*, we use the *sense page* retrieved from Wikipedia for each *concept* to build its feature space. We identify the following features that are potentially useful:

1. Words from the titles of a page (*title_words*) – including words in the title of a *sense page* plus words from all redirecting links that group several *surfaces* of a single *concept* to this page.
2. Words from first section (*first_sec_words*), as opposed to top *n* most frequently used words in the entire page (*frequent_words_n*). We test each feature separately.
3. Words from categories (*cat_words*) assigned to a page – each page in Wikipedia is assigned several category labels. These labels are organized in a taxonomy-like structure. We retrieve the category labels by performing a depth limited search of 2, and split these labels to words.
4. Words from outgoing links that are contained in a list structure on a page (*list_link_words*) – the intuition is that links found in list structures may be more important than other links on the same page, as suggested by Turdakov and Velikhov (2008). We take the "target" of a link rather than the "surface" of a link. E.g., in the page about "King (chess)", the first sentence "In chess, the King is the most important piece" contains a link "Chess_piece" with surface form "piece"; in such case, we take the target "Chess_piece" other than "piece".
5. Words from all other outgoing links excluding links extracted by feature 4. (*other_link_words*).
6. Words from the describing nouns (*desc_noun_words*) - Kazama and Torisawa (2007) noted that, in the first sentence of a page, the head noun of the noun phrase just after *be* is most likely the hypernym of the concept described by the page. We extend their idea by keeping the entire noun phrase and also including the noun phrases connected by a correlative conjunction word; e.g., we keep the nouns marked in italic in the sentence "Sheffield is a *city* and *metropolitan borough* in South Yorkshire, England", and refer to them as "describing nouns".

Thus, for each concept, we extract the above features from its sense page, and transform the text features into a graph conforming to the random walk model, which is explained in the following section.

3.3 Random walk graph model

A random walk is a formalization of the intuitive idea of taking successive steps in a graph, each in a random direction (Lovász, 1993). Intuitively, the "harder" it is to arrive at a given node starting from another, the less related the two nodes are. The advantage of a random-walk model lies at its strength of seamlessly combining different features to arrive at one single measure of relatedness between two entities. (Iria et al., 2007)

We follow a similar approach to that of Hughes and Ramage (2007). Specifically, we build an undirected weighted typed graph that encompasses all concepts identified in the page retrieval step and their extracted features as described in the previous section. The graph is a 5-tuple $G = (V, E, t, l, w)$, where V is the set of nodes representing the concepts and their features; $E : V \times V$ is the set of edges that connect concepts and their features, representing an undirected path from concepts to their features, and vice versa; $t : V \rightarrow T$ is the node type function ($T = \{t_1, . . . , t_{|T|}\}$ is a set of types, e.g., concepts, or different types of features), $l : E \rightarrow L$ is the edge label function ($L = \{l_1, . . . , l_{|L|}\}$ is a set of labels that define relations between concepts and their features), and $w : L \rightarrow R$ is the label weighing function that assigns a weight to an edge. Thus, given the features described in the previous section, we structure our problem domain with the types and labels presented in Figure 1. Concepts sharing same features will be connected via the edges that connect features and concepts.

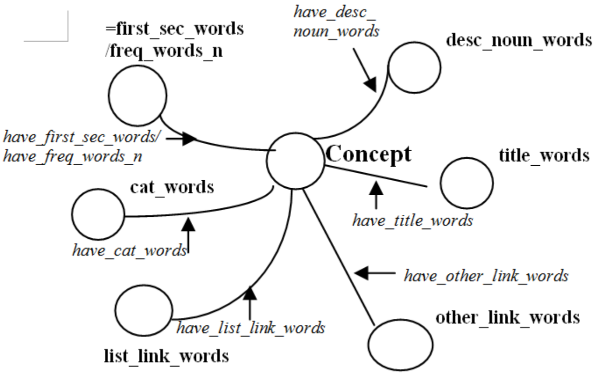

Figure 1. Graph representation model of concepts, features, and their relations. Circles indicate nodes (V) representing concepts and features; bold texts indicate types (T) of nodes; solid lines connecting nodes indicate edges (E), representing relations between concepts and features; italic texts indicate types (L) of edges.

We define weights for each edge type, which, informally, determine the relevance of each feature to establish the relatedness between any two concepts. Let $L_{t_d} = \{l(x, y): (x, y) \in E \wedge t(x) = t_d\}$ be the set of possible labels for edges leaving nodes of type t_d. We require that the weights form a probability distribution over L_{t_d}, i.e.

$$\sum_{l \in L_{t_d}} w(l) = 1$$

We build an adjacency matrix of locally appropriate similarity between nodes as

$$W_{ij} = \left\{ \begin{array}{l} \sum_{l_k \in L} \frac{w(l_k)}{|(i,\cdot) \in E : l(i,\cdot) = l_k|}, (i,j) \in E \\ 0, otherwise \end{array} \right\}$$

Where W_{ij} is the i^{th}-line and j^{th}-column entry of W, indexed by V. The above equation distributes uniformly the weight of edges of the same type leaving a given node. To tune the weight of different features – which we will refer to as a *weight model* (*wm*), we use a simulated annealing optimization method as described in Nie et al. (2005). Details on this are presented in Section 4.2.

We associate the state of a Markov chain to every node of the graph, that is, to each node i we associate the one-step probability $P^{(0)}(j|i)$ of a random walker traversing to an adjacent node j. These probabilities are expressed by the row stochastic matrix $D^{-1}W$, where D is the diagonal degree matrix given by $D_{ii} = \Sigma_k W_{ik}$. The "reinforced" relatedness between two nodes in the graph is given by the t-step transition probability $P^{(t)}(j|i)$, computed by a matrix power, i.e., $P^{(t)}(j|i) = [(D^{-1}W)^t]_{ij}$. In our work, we have set $t = 2$ in order to prevent smoothing out the walk.

This transition results in a sparse, non-symmetric matrix filled with probabilities of reaching node i from j after t steps. To transform probability to relatedness, we use the observation that the probability of walking from i to j then coming back to i is always the same as starting from j, reaching i and then coming back to j. Thus we define a transformation function as:

$$Rel(i,|j) = Rel(j|i) = \frac{P^{(t)}(j|i) + P^{(t)}(i|j)}{2}$$

and we normalize the score to range $\{0, 1\}$ by using the maximum $Rel(i|j)$ as denominator to every value of $Rel(i|j)$ using:

$$\mathrm{Rel}(i \mid j) = \frac{\mathrm{Rel}(i \mid j)}{Max_{\mathrm{Rel}(i|j)}}$$

4. Experiment

The purpose of our experiments is twofold: to evaluate the effectiveness of the proposed method and to study the importance of different features extracted from Wikipedia and their contribution to the accuracy of the method. We firstly describe the datasets for the experiment and the baseline systems we are comparing with in Section 4.1. Next, we present our experiments for feature analysis, and then compare our system performance against baseline systems and state-of-the-art systems in Section 4.2. Finally we evaluate the usefulness of our method in a named entity disambiguation task.

4.1 Dataset and baseline systems

The evaluation of semantic relatedness measures in all previous literature makes use of human judgments as gold standard. The datasets are usually organised as sets of pair of words, such as "dog" with "tiger", and "cat" with "pet", for which domain experts are required to score their semantic relatedness within a scale of minimum and maximum values. System produced scores are then compared against human-rated values using a correlation coefficient function. We make use of three such datasets that have been used extensively. These include the Rubenstein and Goodenough (1965) dataset consisting of 65 pairs of words (*RG-65*), the Miller and Charles (1991) dataset (*MC-30*) that contains a subset of 30 pairs from the *RG-65* dataset, and the *WordSimilarity-353* Test Collection (*WordSim353*, Finkelstein et al, 2002). The *WordSim353* contains 353 pairs of words; however it contains two subsets that are annotated by different annotators, each consisting of 153 and 200 pairs respectively. Therefore we treat this dataset as three different sets, *Fin-153* containing the 153 pair of words, and *Fin-200* containing the other 200 pairs.

We select five WordNet based approaches as our baseline systems. These are widely accepted and frequently used state-of-the-art methods, including Leacock and Chodorow (*lch*) (1998), Lin (*lin*) (1998), Wu and Palmer (*wup*) (1994), Jiang and Conrath (*jc*) (1997), and Resnik (*res*) (1995). For consistency with previous literature, we use the Spearman rank-order correlation coefficient to evaluate our system and baseline systems against the gold standard datasets. To obtain baseline results we use the Java WordNet::Similarity[5] package, which has already implemented these methods. The Wikipedia dump we used for the experiments is a version obtained on 15th Aug, 2009. The Java Wikipedia Library (JWPL)[6] is used for accessing the contents and structural elements in Wikipedia.

4.2 Feature tuning and analysis

In this section, we study the importance of different features extracted from Wikipedia for measuring semantic relatedness by using the simulated annealing technique as mentioned in Section 3.3. The algorithm explores the search space of possible combinations of feature weights and iteratively reduces the difference in accuracy between the domain experts' annotations and that of our system on the training set. The algorithm allows us to run our method on one dataset in an iterative manner. In each iteration, the algorithm generates a random *weight model (wm)* for the feature set and scores our system results obtained with that model against the gold standard.

We select the *Fin-200* dataset for feature tuning because it is the largest dataset. We run our system with initial feature set as proposed in Section 3.2 on this dataset for 100 iterations, starting from a uniform distribution of feature weights for all six types of features described in Section 3.2 to find the optimum *wm*. We fix the number of iterations to 100 in all the following simulated annealing experiments. In Table 1, we show the best and worst performance figures and the corresponding *wm* obtained in those iterations.

The results give us an insight of the effects of different features for computing semantic relatedness. As shown in Table 1, the features *title_words, cat_words* and links on the page *(list_link_words* and *other_link_words)* are more important since the increase in their weights (*wm1 and wm2*) leads to improvements in system accuracy; on the other hand, the increase in the weight for feature *desc_noun_words* (*wm3 and wm4*) has the opposite effect, suggesting it a less useful feature.

Feature	*wm1*	*wm2*	*wm3*	*wm4*
title_words	0.17	0.24	0.02	0.03
first_sec_words	0.15	0.1	0.28	0.24
cat_words	0.15	0.14	0.02	0.01
desc_noun_words	< 0.01	< 0.01	0.28	0.24
list_link_words	0.24	0.21	0.39	0.47
other_link_words	0.28	0.3	< 0.01	< 0.01
Correlation	0.39	0.387	0.167	0.165

Table 1. The best and the worst results obtained with initial feature sets and corresponding *wm* from feature tuning experiments (selected from 100 iterations of runs on *Fin-200* dataset)

Based on these observations, we modify our features in order to improve the system performance. To do this, we group the features that bear similar semantics into a single type of feature, since these can be considered as similar features. In particular, we merge the least useful feature *desc_noun_words* and *cat_words* into *cat_words_merged,* because as described in section 3.2, the description noun usually bears the hypernymy relation to the concept described by the Wikipedia article; the category labels have similar nature. Likewise, we merge *list_link_words* and *other_link_words* into *link_words_ merged* as they are equally important features and are both outgoing links. We re-run simulated annealing for another 100 iterations on the same dataset using the revised feature set, and present results in Table 2.

Feature	*wm1*	*wm2*	*wm3*	*wm4*
title_words	0.2	0.22	0.26	0.4
first_sec_words	0.05	0.05	<0.01	< 0.01
cat_words_merged	0.15	0.17	0.56	0.52
link_words_merged	0.6	0.56	0.17	0.07
Correlation	0.422	0.418	0.287	0.28

Table 2. The best and the worst results obtained with *refined* feature sets and corresponding weight models from feature tuning experiments (selected from 100 iterations of runs on the *Fin-200* dataset).

As the results suggest, system performance improves after feature grouping. Also, the *wm's* are generally consistent with the previous findings with the initial feature set; that is, links on the page are more important features than others, and thus tend to receive higher weights; *title_words* and *cat_words_merged* come as the next. Encouraged by these results, we take one further step in feature grouping in order to verify whether this always improves performance. For instance, we test further grouping *title_words* with *cat_words_merged,* and re-run simulated annealing. However, system performance is reduced (with best result of 0.406), suggesting it is useful to keep feature set diverse rather than over-grouping features. Due to space limit, we do not present these results.

Another observation based on figures in Table 2 is that *first_sec_words* receives few weight, indicating the ineffectiveness of this feature. This is possibly due to the sparse feature space created by the varying sizes of first sections of Wikipedia pages. In some cases of short articles, the first section consists of only one sentence. To improve this, we replaced it with the most frequent *n*

[5] http://www.cogs.susx.ac.uk/users/drh21/

[6] http://www.ukp.tu-darmstadt.de/software/jwpl/

words extracted from a page (*frequent_words_n)*, and run simulated annealing again varying the number *n*. Results are shown in Table 3: this modification further improves the system performance, and the *wm* derived with different *n* tends to be consistent.

Feature	*n=25*	*n=50*	*n=75*	*n=100*
title_words	0.16	0.17	0.16	0.19
freq_words_n	0.28	0.30	0.39	0.32
cat_words_merged	0.11	0.12	0.1	0.11
link_words_merged	0.45	0.41	0.35	0.38
Correlation	0.382	0.443	0.454	0.41

Table 3. Results obtained by replacing the feature *first_sec_words* with *n* most frequent words, and varying the value *n* and weight models. Figures are obtained on the Fin-200 dataset after 100 iterations of runs

As a result of feature tuning and analysis, we obtain the best feature set and an optimum feature *weight model*, as summarized in Table 4. Next, we apply these feature sets with the *wm* on other datasets chosen for our experiments, and compare our results against the baseline and the state-of-the-art systems in Table 5 and 6.

title_ words	*freq_ words_75*	*cat_ words_merged*	*link_ words_merged*
0.16	0.28	0.11	0.45

Table 4. Final feature set and weight model

	Fin-153	*Fin-200*	*RG-65*	*MC-30*
Our system	**0.71**	**0.46**	0.76	0.71
lch	0.32	0.24	0.79	0.75
lin	0.39	0.25	0.78	0.76
wup	0.37	0.24	0.78	0.77
jc	0.38	0.23	0.78	0.81
res	0.37	0.25	0.78	0.76

Table 5. Comparison of our proposed system against baseline WordNet-based systems.

	Fin-153	*Fin-200*	*RG-65*	*MC-30*
Our system	**0.71**	0.46	**0.76**	**0.71**
Zesch et al. (2008a) ESA	0.62	0.31	n/a	n/a
Zesch et al. (2008a) Wiki	0.7	**0.5**	**0.76**	0.68
Strube & Ponzetto (2006)	0.55	n/a	0.69	0.67

Table 6. Comparison of our system with state-of-the-art systems that use Wikipedia.

4.2 Named Entity Disambiguation

In order to evaluate the usefulness of our method, we apply it to a Named Entity disambiguation (NED) task, the purpose of which is mapping mentions of entities in a text with the object they are referencing. Our method of NED employs the output of the semantic relatedness computation. Given a set of names (*surfaces*) extracted from a document, we hypothesize that the entity or meaning (*concept*) the name refers to is collectively defined by other entities. For example, "Apple" is likely to mean the American company if it occurs together with "Macintosh", "Microsoft" in the same document; but it is likely to mean "fruit" if it occurs with "pear", "passion fruit". Therefore, given $S = \{s_1, \ldots, s_n\}$ the set of surfaces in a document, we apply our method to extract the set of all their possible senses (*concepts*) from Wikipedia, denoted by $C = \{c_{1_k}, \ldots, c_{m_k}\}$ (with $k=1 \ldots |S|$), and compute inter-concept semantic relatedness to form a matrix of concept relatedness *R*, where $R(i|j)$ indicates the strength of relatedness between concept c_{i_k} and concept $c_{j_{k'}}$ (where $k \neq k'$, that is c_{i_k} and $c_{j_{k'}}$ have different *surfaces*). The NED algorithm is then defined as a function $f\colon S \rightarrow C$, which given a set of *surfaces S* returns the list of disambiguated *concepts*, using *R*. The function is designed to ensure that only one single *concept* is chosen for each *surface*; and the choice of *concept* is collectively determined by its relatedness scores with all other *concepts* in *R*. We defined and tested three different functions *f*, details of which can be found in Gentile et al., (2009).

Briefly, let $cand_{k_i}$ be the list of candidate winner *concepts* for each *surface*, with *i* being the candidate *concept* for *surface* s_k $(k = 1 \ldots |S|)$. We define the *highest method* ($f_{highest}$), with which for each *surface* s_k that has more than one candidate winner, i.e., *i concepts*, we simply pick the *concept* that has the highest value in the matrix *R* from those *i concepts*. We then define the *combination method* (f_{comb}), which calculates for each *concept* c_{i_k} the sum of relatedness with all different *concepts* $c_{j_{k'}}$ from different *surfaces* (such as $j \neq i$, $k \neq k'$). Given $V = \{v_1, \ldots, v_{|C|}\}$ the vector of such values, the function returns for each surface s_k the concept c_{i_k} that have the max v_i. Finally we define the *propagation method* (f_{prop}) that works as follows: taking as seed the highest similarity value in the matrix *R* we fix the two *concepts i* and *j* giving that value: for their *surface* s_k and $s_{k'}$ we delete rows and columns in the matrix *R* coming from other *concepts* for the same *surfaces* (all c_{t_k} and $c_{t_{k'}}$ with $t \neq i$ and $t \neq j$). This step is repeated recursively, each turn the next highest value in *R* is selected and corresponding rows deleted until only one *concept* row remains in *R* for each *surface*.

The corpus used for evaluating NED is the benchmarking dataset available in Cucerzan (2007). It is created based on 20 news stories, for each of which a list of named entities are extracted. Most names are ambiguous, since they point to multiple entries of entities defined in Wikipedia. The purpose of NED is to select the accurate Wikipedia entries for each name depending on the context collectively defined by other named entities in each news story. The number of entities in each story can vary from 10 to 50.

For each news story, we take the extracted names as input to our method proposed in section 3, and compute pair-wise semantic relatedness between their underlying concepts. The result forms the semantic relatedness matrix *R*, which is then submitted to the three NED functions proposed before. The results of the NED experiment are shown in Table 7.

f_{comb}	**91.5%**
f_{prop}	**68.7%**
$f_{highest}$	**82.2%**
Cucerzan (2007) baseline	51.7%
Cucerzan (2007) best	91.4%

Table 7. Comparison of our NED system against state-of-the-art

5. Discussion

Firstly, comparing all systems' performance using figures in Table 5 and 6 we notice that there is no single system that always achieves the best results on all datasets, indicating measuring semantic relatedness between words or phrases is not an easy task. Compared to baseline systems in Table 5, our system gains significant improvement on the *Fin-153* and the *Fin-200* datasets.

Secondly, compared to other state-of-the-art systems, our method achieves higher accuracy than Strube and Ponzetto (2006), and the re-implementation (Zesch et al. ESA) of Gabrilovich & Markovitch (2007)'s ESA method by Zesch et al. (2008a) and re-testing using a more recent version of Wikipedia knowledge base[7]. Compared to the method that also uses Wikipedia by Zesch et al. (2008a), our system produces better results on two datasets. As aforementioned, these systems only make use of one or two types of un-weighed features extracted from Wikipedia. The performance improvement by our system suggests that using multiple Wikipedia features that are weighed according to their importance for the semantic relatedness task can achieve better results.

The effectiveness of the method is then further verified by its application to an NED task. Compared to the baseline of Cucerzan (2007), all three disambiguation functions produced significant improvement, possibly indicating the good quality of the semantic relatedness matrix generated by our method. The best function is f_{comb}, which further improved over the best system of Cucerzan (2007).

Our findings from feature tuning suggest grouping features extracted from Wikipedia by similar semantics achieve better results than treating them separately. And different features contribute differently to semantic relatedness, as proved through feature tuning. The most indicative features for this specific task are words from outgoing links on a page; followed by the most frequent words on the page, words from titles and redirection links, and words from category labels. Additionally, most frequent words on a page are more useful features than words extracted from the first section.

In addition, for the feature *cat_words* derived from category labels, we also tested with search depth of 1 and 3, both produced worse results; which suggests when the search depth is too low the feature space becomes too sparse and less useful, however raising the depth may introduce noise. Also, we carried out experiments using the exact phrases instead of treating them as words, but results are also worse. Indeed, using words as features may be more useful since many information and structural elements extracted from Wikipedia can be too fine-grained. For example, the page for "tiger" has categories "Mammals of Asia", "Mammals of "Indonesia"; and the page for "jaguar" has categories "Mammals of North America" and "Mammals of South America". The category labels are clearly too restrictive, and the most indicative information in this case is in the word "Mammals". Although using word-based features can introduce noise, our experiments have shown that, in general, it leads to better accuracy. Due to the space limitation we do not present details of these experiments.

6. Conclusion

In this paper, we introduced a novel approach to measuring semantic relatedness between words or concepts using a random walk model. We investigate various features extracted from Wikipedia for a pair of words or concepts and their importance in measuring semantic relatedness. Our experiments show that, measuring semantic relatedness is a difficult task; as a result, no single system reported in the literature always outperforms the others on all testing datasets. In general, by integrating various weighted features extracted from Wikipedia through a random walk model, we can obtain better results than WordNet-based approaches, and other Wikipedia-based approaches that only make use of limited un-weighted features. Also, we empirically derived the best features and analyzed their importance for the semantic relatedness task. The conclusions from the analysis can be useful references to future research in computing semantic relatedness.

However, the effects of using a random walk model in combining diverse features are unclear. In the future, we will study and compare effects of other similarity functions with diverse feature sets in semantic relatedness task, and compare against other distributional models such as Latent Semantic Analysis (LSA) to investigate the possibility of improving performance with other similarity functions. Also, we will research the possibilities of integrating different lexical resources in a coherent methodology. In addition, we will look into adapting our methodology to computing semantic relatedness between longer text fragments, such as sentences and snippets, which is another major challenge in the studies of semantic relatedness.

[7] The reason for using this re-implementation and re-testing is that original work by Gabrilovich and Markovitch (2007) treats the WordSimilarity353 dataset as a single testing set, thus results are not directly comparable to ours

7. Acknowledgements

This work is funded by the Archaeotools project that is carried out by Archaeology Data Service, University of York, UK and the Organisation, Information and Knowledge Group (OAK) of the Department of Computer Science, University of Sheffield, UK.

8. References

Banerjee, S., Pedersen, T. (2003). Extended gloss overlap as a measure of semantic relatedness. *In Proceedings of IJCAI-03,* pp. 805-810

Benjamin, C., Woon, W., Wong, K. (2008). Customized Term Weighting Scheme for Document Classification. In *Proceedings of the International Conference on Computer and Communication Engineering*

Budanitsky, A., Hirst, G. (2006). Evaluating WordNet-based measures of lexical semantic relatedness. In *Computational Linguistics*, 32(1), pp. 13-47

Bollegala, D., Matsuo, Y., Ishizuka, M. (2007). An integrated approach to measuring semantic similarity between words using information available on the web. In *Proceedings of NAACL HLT 2007*, pp. 340-347

Cucerzan, S. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *EMNLP'07*

Fellbaum, C. (1998). WordNet an electronic database. Cambridge, MA: MIT press.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: the concept revisited. In *ACM Transactions on Information Systems*, 20 (1), pp. 116 – 131

Gabrilovich, E., Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of IJCAI'07*, pp. 1606-1611

Gentile, A., Zhang, Z., Xia, L., Iria, J. (2009). Graph-based semantic relatedness for named entity disambiguation. *In S3T*

Guo, H., Jiang, J., Hu, G., Zhang, T. (2005). Chinese named entity recognition based on multilevel linguistic features. In *Lecture notes in computer science*, pp. 90-99

Hughes, T., Ramage, D. (2007). Lexical semantic relatedness with random graph walks. In *Proceedings of EMNLP-CONLL'07*

Jiang, J., Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International conference on research in Computational Linguistics*, pp. 19-33

Jin, R., Chai, J., Si, L. (2005). Learn to weight terms in information retrieval using category information. In *Proceedings of the 22nd international conference on Machine learning*, pp.353-360

Iria, J., Xia, L., Zhang, Z. (2007). WIT: Web People Search Disambiguation using RandomWalks. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007),* pp. 480–483

Kazama, J., Torisawa, K. (2007). Exploiting Wikipedia as external knowledge for named entity recognition. *In Proceedings of EMNLP-07*.

Leacock, C., Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. In C. Fellbaum (Ed.), *WordNet. An Electronic Lexical Database*, Chp. 11, pp. 265-283.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th COLING*, pp. 768-774

Lovász, L. (1993). RandomWalks on Graphs: A Survey. In *Combinatorics, Paul Erdos is Eighty*, 2, pp. 1-46.

Miller, G., Charles, W. (1991). Contextual correlates of semantic similarity. In *Language and Cognitive Processes,* 6(1): 1-28

Müller, C., Gurevych, I. (2009). Using Wikipedia and Wiktionary in Domain-Specific Information Retrieval, in *Evaluating Systems for Multilingual and Multimodal Information Access*.

Nie, Z., Zhang, Y., Wen, J., Ma, W. (2005). Object-level ranking: bringing order to web objects. In *Proceedings of WWW'05*

Patwardhan, S., Banerjee, S., and Pedersen, T. (2003). Using measures of semantic relatedness for Word Sense Disambiguation. In: *Proceedings of International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-03)* pp. 241-257.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI-95*, pp. 448-453

Rubenstein, H., Goodenough, J. (1965). Contextual correlates of synonymy. In *Communications of the ACM,* 8(10):627-633

Strube, M., Ponzetto, S. (2006). WikiRelate! Computing semantic relatedness using Wikipedia. In *AAAI'06*

Turdakov, D., Velikhov, P. (2008) Semantic Relatedness Metric for Wikipedia Concepts Based on Link Analysis and its Application to Word Sense Disambiguation. *SYRCoDIS*

Weale, T., Brew, C., and Fosler-Lussier, E. (2009). Using the Wiktionary Graph Structure for Synonym Detection, *Proceedings of the Workshop on the People's Web Meets NLP, ACL-IJCNLP*

Wu, Z., Palmer, M. (1994). Verb semantics and lexical selection. In *Proceedings of ACL94*. pp. 133-138

Zesch, T., Müller, C., Gurevych, I. (2008a). Using Wiktionary for computing semantic relatedness. In Proceedings of AAAI'08

Zesch, T., Müller, C., Gurevych, I. (2008b). Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. *In Proceedings of LREC'08*, 2008