A Random Walk Framework to Compute Textual Semantic Similarity: a Unified Model for Three Benchmark Tasks

Majid Yazdani
Idiap Research Institute / EPFL
Martigny / Lausanne, Switzerland
Email: majid.yazdani@idiap.ch

Andrei Popescu-Belis
Idiap Research Institute
Martigny, Switzerland
Email: andrei.popescu-belis@idiap.ch

***Abstract*—A network of concepts is built from Wikipedia documents using a random walk approach to compute distances between documents. Three algorithms for distance computation are considered: hitting/commute time, personalized page rank, and truncated visiting probability. In parallel, four types of weighted links in the document network are considered: actual hyperlinks, lexical similarity, common category membership, and common template use. The resulting network is used to solve three benchmark semantic tasks – word similarity, paraphrase detection between sentences, and document similarity – by mapping pairs of data to the network, and then computing a distance between these representations. The model reaches state-of-the-art performance on each task, showing that the constructed network is a general, valuable resource for semantic similarity judgments.**

I. Introduction

Many natural language processing applications must estimate the semantic similarity of pairs of text fragments provided as input, e.g. information retrieval, summarization, or textual entailment. A simple lexical overlap measure cannot be successful when text similarity is not based on identical words, and in general when words are not independent. We propose here a method to use structured and unstructured knowledge extracted from the English version of Wikipedia to compute semantic similarity. A document network is built from Wikipedia, capturing various forms of user-contributed knowledge, by considering that every article in Wikipedia corresponds to a concept node in a graph. Relations between concepts are derived from: hyperlinks between articles, lexical content of articles, templates and infoboxes that are invocated, and membership in a category. To estimate the semantic similarity of two text fragments (single words, phrases, sentences, or entire documents), they are first mapped to the vertices of the network built from Wikipedia, then the distance between sets of vertices is computed.

For example, let us consider two vertices in our network corresponding to the English Wikipedia documents on "Natural language processing" (NLP) and "Machine learning" (ML), with the following notations for relations between documents A and B: $A \xrightarrow{links} B$ indicates that the text of A contains a hyperlink to B; $A \xrightarrow{cat.} B$ indicates that A and B belong to the same Wikipedia category and $A \xrightarrow{cont.} B$ indicates that they have similar lexical contents (as defined in Section III). It is possible to find various paths between the NLP and ML documents by using a sequence of one or more relations from Wikipedia. Some of the simplest paths are shown below, but as their length increases, there are of course many more: ML $\xrightarrow{links}$ NLP;
ML $\xrightarrow{links}$ Artificial Intelligence $\xrightarrow{links}$ NLP;
NLP $\xrightarrow{links}$ Data Mining $\xrightarrow{links}$ ML;
ML $\xrightarrow{cat.}$ Mallet (software project) $\xrightarrow{links}$ NLP;
ML $\xrightarrow{cont.}$ Algorithm $\xrightarrow{links}$ NLP;
NLP $\xrightarrow{cat.}$ Grammar Induction $\xrightarrow{links}$ ML.

In this work, we use the random walk framework to compute various properties of the paths between concept nodes. We overview related work in Section II, and describe in Section III the English Wikipedia resource used to build the network, in particular the type of knowledge used to infer paths between vertices. Similarity measures are computed in Section V, and are applied in Section VI to three well-known problems of semantic computing: word similarity (VI-A), document similarity (VI-B), and paraphrase detection (VI-C). The results on each data set are close to the best published ones, showing that this resource provides a unified and robust answer to several semantic tasks.

II. Related Work

Several approaches enhance the overlap-based lexical similarity distance. A taxonomy of concepts and relations can be constructed manually or automatically onto which fragments to be compared can be mapped, such as Wordnet [1] and Cyc This makes use of coherent concepts that humans can understand and reason about, but the knowledge representation granularity is limited by the taxonomy. Building and maintaining the knowledge bases requires a lot of effort from experts. Moreover, the bases cover only a small fraction of the vocabulary of a language and usually include few proper names, conversational words, or technical terms. Another approach makes use of unsupervised methods to construct a semantic representation of documents by analyzing mainly co-occurrence relations between words in a corpus. Latent Semantic Analysis(LSA) [2] and probabilistic LSA [3] are

unsupervised methods that construct a low-dimensional feature representation or "concept space", in which words are no longer supposed to be independent. These methods offer a larger vocabulary coverage, but the resulting "concepts" are difficult for humans to interpret. Mihalcea et al. [4] compare several knowledge-based and corpus-based methods. In addition, word similarity and word specificity are used to define one general text semantic similarity measure. However, the measure is only suitable for similarity between two short text fragments because it needs to compare all word pairs.

Explicit Semantic Analysis (ESA) [5], instead of mapping a text to a node (or a small group of nodes) in a taxonomy, maps the text to the entire collection of available concepts, by computing the degree of affinity of each concept to the input text. ESA uses Wikipedia articles as a collection of concepts and maps texts to this collection of concepts by use of terms/documents affinity matrix. Similarity is measured in the new concept space, with the implicit (but questionable) assumption that concepts are orthogonal. However, ESA does not use link structure and other structured knowledge from Wikipedia, although these contain valuable information about relatedness between articles.

Milne and Witten [6] attempt to enrich documents (e.g. news stories or educational) with links to explanatory Wikipedia articles, thus bringing structured knowledge to any unstructured text fragment, using a bag of words representation. Their method requires heavy computation, as it performs disambiguation for all n-grams and computes relatedness of all senses to the context articles.

The closest antecedent of this study is the work of Yeh et al. [7], who also start from a graph of documents and hyperlinks computed from Wikipedia. Then, a personalized page rank [8] is computed for each text fragment, with the teleport vector being the one resulting from the ESA algorithm. To compute semantic similarity between two texts, Yeh et al. then simply compare their personalized page rank vectors. By comparison, we consider in the present work, in addition to hyperlinks, the effect of word co-occurrence between article contents. We also propose a different method to capture knowledge from category membership and template calls, which makes random walk computation faster and easier to handle. We also use two other measures, in addition to personalized page rank, capturing different properties of the network and giving improved final results.

III. Wikipedia as a Network of Concepts

We consider the Wikipedia encyclopedia as a network of concepts. Every article in Wikipedia is considered as one concept in the network – after a filtering procedure explained at the end of this section. Based on different relations between articles that we can infer from Wikipedia, we consider the following link types in the construction of the network.

Hyperlinks between articles. We assume that if page A links to page B in Wikipedia, this shows that article B helps to understand article A, so B is related to article A. To capture this relation, we simply use the hyperlink structure between articles in Wikipedia to our network.

Content links. Another important potential relation between articles is based on their word co-occurrence. If two articles are using the same words, then there is a similarity relation between them. To capture content similarity we compute the cosine similarity between articles body vectors. We link every article to its k most similar articles (here, $k = 10$), with a weight according to their similarity score. By using this approach we can benefit from word co-occurrence knowledge between millions of articles in Wikipedia.

Category links. Every article in Wikipedia is at least assigned to one category, and often to more than one. There is a hierarchy between categories, constituting a directed acyclic graph. To capture relations between articles according to their categories, given an article, all other articles are sorted based on their category membership similarities and the top k are chosen to connect to the article according to their similarity scores (here, $k = 10$). In other words, we build a k nearest neighbor graph between articles based on category membership similarity. To compute category membership similarity between articles, we consider every article as a document that consists of its direct categories and one level ancestors upper ones (limitation to one level is based on experiments). Similarity between two articles according to category membership can be computed by cosine similarity in the space of all categories.

Templates and Infoboxes. Each template or infobox is defined in its own article, and then referenced in calls from other articles. Each template call includes a set of named parameters to the template. Calling a same template from two articles shows a certain relation between them that is different from category membership. Again, there is a hierarchy structure between templates. To capture relations between articles based on templates we proceed as with categories. Every article is connected to the k most similar articles according to the template call similarity scores (here, $k = 10$).

Overall, the concept network is built from Wikipedia by using the Wex [9] data set. We drop all Wikipedia articles that belong to the following name spaces: Talk, File, Image, Template, Category, Portal, and List – because these articles do not describe concepts. In addition to that, we drop all articles with less than 100 words, as their content is not reliable for the distance computation described below. Yeh et al. set this limit to 2,000 non-stop words, but we found that we might lose important concepts with this limit. After filtering, we are left with **1,264,611 articles** and **35,214,537 hyperlinks** between them. For every hyperlink we extracted the corresponding anchor text and mentioned it as another title for the linked article. Similarly, we added the title of each redirect page to the title of the article it redirects to. There are thus three types of titles for an article: original title, anchor texts pointing to it, and redirects.

IV. Mapping Text Fragments to Network Concepts

To compute similarity between two texts, first we map them to the network of concepts, and then we compute similarity between the sets of vertices resulting from the mapping. To perform mapping of a given text, we compute lexical similarity of the text and of the Wikipedia articles corresponding to the vertices in the network. The text is mapped to vertices of the k most similar articles according to the corresponding lexical similarity score. To have a more accurate lexical similarity measure, we distinguish words in the title from words in the body of articles and give more importance to the former. After each text fragment is mapped to the network, methods to compute similarity between concepts in the network (see next section) are applied to solve semantic similarity tasks (see SectionVI).

V. Semantic Similarity using Random Walk

We define now a general framework to compute similarity between two texts by using a network of concepts and the documents associated to them. The framework is independent of a specific network, though we will keep the Wikipedia-based network in mind. Let $S = \{s_i\}$ be set of concept vertices with size n in the network. The relation between concepts s_i and s_j is represented by a combination of L different types of directed links. Link types between concepts are represented by L matrices $A_l, 1 \leq l \leq L$, where A_l shows the structure of links of type l between all concepts. In each matrix, $A_l(i,j)$ is the weight of the link (of type l) between s_i and s_j.

The transition matrix C_l that shows the probability of transition between concepts based on link type l can be computed from the A_l matrix: $C_l(i,j) = \frac{A_l(i,j)}{\sum_{r=1}^{n} A_l(i,r)}$.

Let the weight w_l show the importance of link type l in the Markov process related to the random walk over this network (actual weight values are disscussed in Section VI). If we note the transition matrix of this walk by C, then we have: $C = \sum_{l=1}^{L} w_l \times C_l$. $C_{i,j}$ is the transition probability between concepts s_i and s_j.

In the following subsections, we introduce various algorithms to measure similarity between two texts by measuring different properties of the concept network. We suppose throughout this section that the probability vector r resulting from mapping a text to the vertices of the network is given. The vector r indicates the probability of concepts in the network given the text, therefore the sum of its elements is one.

A. Hitting Time and Commute Time

The first method to compute the distance between two vertices in the graph uses the hitting time from node s_i to s_j, noted H_{ij}. This is the average number of steps a random walker that starts from s_i should take to visit node s_j for the first time. If $p(s_i \rightarrow^t s_j)$ is the probability to visit s_j for the first time after exactly t steps starting from s_i, we can define H_{ij} as: $H_{ij} = \sum_{t=1}^{\infty} t \times p(s_i \rightarrow^t s_j)$

This defines hitting time as the expected path length of the first visit of s_j starting from s_i. It is possible to truncate the computation after T steps, i.e. to consider paths of length at most T, e.g. using the approach proposed in [10]. Because hitting time is not symmetric and we are looking here for a similarity measure, we define commute time as: $C_{ij} = H_{ij} + H_{ji}$.

Hitting time and commute time are defined between two vertices. Let us consider now again r_1 and r_2, the resulting vectors from mapping texts to the network. First, let us measure hitting time from r_1 to a vertex s_j. We assume there is a virtual node corresponding to r_1 that is connected to concept s_i according to $(r_1)_i$ and we do not consider penalty for traveling links between the virtual node and concepts in the network. According to this assumption, hitting time can be computed as: $H_{r_1 j} = \sum_i (r_1)_i H_{ij}$. To compute the hitting time from a single vertex s_i to the vector r_2 we assume there is a virtual vertex V_{r_2} representing r_2. Concept s_i is connected to it according to $(r_2)_i$, Therefore we introduce matrix C' by adding a new row to matrix C with all elements zero to show V_{r_2} and update other rows as follows:

$$C'_{ij} = C_{ij}(1 - (r_2)_i) \text{ for } i, j \neq V_{r_2}$$
$$C'_{iV_{r_2}} = (r_2)_i \text{ for all } i,$$
$$C'_{V_{r_2} j} = 0 \text{ for all } j$$

Now we compute $H_{s_i V_{r_2}}$ to measure $H_{s_i r_2}$. Based on the above equations we are able to compute Hitting time between vectors r_1 and r_2 resulting from mapping the texts to the network vertices: $H_{r_1 r_2} = \sum_i r_{1_i} H_{s_i V_{r_2}}$. Finally, to approximate similarity between two texts, we compute $C_{r_2 r_1}$, the average distance between the texts given the network.

B. Personalized Page Rank

If the average path length connecting two concepts is long, the distance based on hitting time is large regardless of the "connection strength" between them. The personalized page rank (PPR) distance [8] considers "connection strength" between concepts with a penalty for long paths. Let $p(s_i|r, N)$ be the probability of concept s_i given r and N, where r is the normalized vector resulted from the mapping function and N is the network. To compute this probability, we imagine a random walk process, the input of which is the vector r and the output of which is a concept in the network as follows:

Step 0: Choose the initial state of the transition process with probability $P(S_0 = s_i|r) = r_i$ where r_i is i^{th} element of the vector r showing the probability of concept s_i in r.

Step t: Given that we have chosen state S_{t-1}, then with probability $1 - \alpha$ return the concept corresponding to S_{t-1} and reset the process to step 0. Otherwise, with probability α, choose next concept according to transition matrix C.

The probability of returning s_i given r, $p(s_i|r, N)$, is the sum over all probabilities of returning s_i given r with different lengths of random walk:

$$p(s_i|r, N) = \sum_{t=0}^{\infty} p(s_i|r, N)^t = \sum_{t=0}^{\infty} (1 - \alpha)\alpha^t (rC^t)_i$$

In the above equation, $(rC^t)_i$ is ith element of the vector rC^t and $p(s_i|r, N)^t$ shows the probability of returning s_i after t

steps. The parameter α conveys the notion of "distance": a smaller α makes the process tend to return concepts with closer distance to r. If vectors r_1 and r_2 are the results of mapping two text fragments to the network, then after running the above process for each of them, to approximate text semantic similarity between them, we compare the two resulting probability distributions for r_1 and r_2 (using cosine similarity in the experiments below).

C. Visiting Probability and a Truncated Approximation

In the computation of PPR, self-loops (that is, paths that start from a concept and end to it) boost the probability of the concept. If some pages have this type of loops after using PPR, then they have high probability although they might not be very close to the teleport vector. We introduce Visiting Probability (VP) to address this issue. Given the initial probability vector r of the network vertices, and a vertex s_j in the network, we compute the probability of visiting s_j for the first time starting from r in the network according to the following process:

Step 0: Choose initial state with probability $P(S_0 = s_i|r) = r_i$.

Step t: Given that we have chosen state S_{t-1}, if $S_{t-1} = s_j$ then return success and finish the process. Otherwise, with probability of α choose next concept according to transition matrix C^t and with probability $1 - \alpha$ return fail.

We introduce C' as being equal to the transition matrix C, except that in row j, $C'_{jk} = 0$ for all k. This indicates the fact that when the random walker visits j for the first time, it can not exit from it and its probability mass drops to zero in the next step. We defined this new transition matrix to fit with the definition of VP as the probability of *first* visit of s_j.

We define $p^t(success)$ as the probability of success in the above process at step t. We can then compute $p^t(success)$ as follows: $p^t(success) = \alpha^t \times (rC'^t)_j$. The probability of success in the above process is the sum over all probabilities of success with different lengths:

$$p(success) = \sum_{t=1}^{\infty} p^t(success) = \sum_{t=1}^{\infty} \alpha^t \times (rC'^t)_j$$

The probability of success in the above process shows the probability of visiting s_j for the first time when we consider a random walk with a penalty for long paths. This probability converges if s_j is reachable from all vertices in the graph.

Truncation in this case can be done by looking at the probability of not returning success and failure in first t steps, which is: $\sum_{i \neq j}^{n} \alpha^t (rC'^t)_i$, that is, the probability mass at time t at all vertices except s_j. If $p_t(success)$ denotes the probability of success considering path of length at most t, then we have: $p(success) - p_t(success) \leq \sum_{i \neq j}^{n} \alpha^t (rC'^t)_i$. The term on the right is decreasing over time because α^t and $\sum_{i \neq j}^{n} (rC'^t)_i$ are both decreasing over time. Truncate the computation, we can return $p_t(success)$ as an approximation for $p(success)$ with at most $\sum_{i \neq j}^{n} \alpha^t (rD'^t)_i$ error.

A more flexible method is to truncate different paths at different times, more specifically truncate paths with lower probabilities in earlier steps and let paths with higher probabilities continue more steps. If $\alpha^t (rC'^t)_i$ denotes the probability of being at s_i in time step t, then if it is small enough we can neglect it and set it to zero. The maximum error caused by this truncation is $\alpha^t (rC'^t)_i$. This means we no longer follow paths that are at s_i in time step t. We used this truncated visiting probability in our experiments below, showing competitive results while allowing a very fast computation time – an essential requirement when using the entire Wikipedia.

Finally, to compute the visiting probability of vector r_2 from vector r_1, we assume a virtual vertex representing r_2, V_{r_2}, in the graph. We connect all concepts s_i to V_{r_2} according to r_i. Therefore we introduce matrix C' by adding a new row to matrix C with all elements zero to show V_{r_2} and update other rows as follows:

$$C'_{ij} = C_{ij}(1 - r_{2_i}) \text{ for } i, j \neq V_{r_2}$$
$$C'_{iV_{r_2}} = r_{2_i} \text{ for all } i,$$
$$C'_{V_{r_2}j} = 0 \text{ for all } j$$

To compute similarity between two texts, we average between visiting probability of r_1 given r_2 and visiting probability of r_2 given r_1. A larger probability denotes more similarity between vectors.

VI. Experiments with Three Benchmark Tasks

In this section, we discuss the results of the random walk methods described above, which approximate text similarity on different data sets. First, we examine each link type separately and measure the effectiveness of random walk. Then we measure walk effectiveness on the combinations of link types.

A. Word Similarity

We used Word Similarity-353 test collection [11] to measure the effectiveness of our models for word similarity. In this collection, there are 353 pairs of words, and the average human similarity score is given for each pair. We measured Spearman Rank correlation between results of different walks and human judgments based on exact mapping of each word to its closest Wikipedia article. Results of walks on every link type separately are given in Table I.

TABLE I
SPEARMAN CORRELATION BETWEEN RESULT OF RANDOM WALKS AND HUMAN JUDGMENTS ON WSIM353

Link type	Commute time	Personalized page rank	Visiting probability
Hyperlink graph	.654	.664	.684
Content graph	.495	.595	.573
Category graph	.103	.490	.371
Template call graph	.010	.302	.250

In this experiment, personalized page rank (PPR) and hitting time are truncated after 5 steps, and every path with probability less than 10^{-5} is truncated when computing visiting probability (VP). In the experiments, the results of PPR and VP were always higher than those of hitting time. It is interesting to find out how much the results of a random walk on different

link types are correlated. In Table II we give the Spearman correlation between the scores obtained with VP on different link types. The correlation shows how much the resulting scores based on different link types differ.

We also examined random walk on some different combination of links. We chose the combination weights experimentally by considering the correlation between link types (shown in Table II) and results on each individual link type, with results given in Table III. The results show that random walk results are improved by combining links in comparison with random walks on individual link types. For example, when equally combining hyperlinks, content links and category links, results are improved in comparison with each link type individually.

TABLE II
SPEARMAN CORRELATION BETWEEN WSIM353 RESULTS OF VISITING PROBABILITY ON DIFFERENT LINK TYPES

Link types	**Content**	**Category**	**Template**
Hyperlinks	.7164	.392	.378
Content	–	.374	.517
Category	–	–	.281

TABLE III
CORRELATION BETWEEN WALK RESULTS ON COMBINATIONS OF LINKS AND HUMAN JUDGMENT FOR W353 (PPR: PERSONALIZED PAGE RANK; VP: VISITING PROBABILITY)

w_1 w_2 w_3 w_4	**PPR**	**VP**
0.25 0.25 0.25 0.25	.686	.696
0.4 0.4 0.1 0.1	.688	.703
0.3 0.3 0.3 0.1	**.706**	**.707**
(0.7 0.1 0.1 0.1)2 - (0.2 0.6 0.1 0.1)3	.691	.705
(0.2 0.6 0.1 0.1)2 - (0.7 0.1 0.1 0.1)3	.682	.690
(0.5 0.1 0.4 0)2 - (0.3 0.6 0.1 0)3	**.709**	**.711**

In addition, we examined two-stage random walks, in which at the first stage the network is built by one set of weights and in the second stage with a different set of weights. The hypothesis here is that there might be a set of links that are more useful to be explored first and some other links that are more useful to be explored later, as discussed by Collins-Thompson and Callan [12]. For example, in Table III, the row corresponding to (0.5 0.1 0.4 0)2 - (0.3 0.6 0.1 0)3 shows that the random walker first explored mainly hyperlinks and categories for two steps and then for three steps the random walker mainly followed article content lexical similarities (this two-stage random walk gave the best result in our experiments). Our observation shows that multi-stage random walk gives different results than one stage random walk and can achieve better results in some cases.

Gabrilovich and Markovitch [5] provide the results of other methods on the same data set. The best result belongs to Explicit Semantic Analysis (ESA) with 0.74 Spearman correlation (results of other methods are given in [5]). We implemented ESA based on our set of concepts derived from Wikipedia and our result was a 0.52 correlation. This difference can be due to different versions of Wikipedia in our experiments, but more probably to a different way of filtering pages, which might lead to a substantially different set of final concepts. Therefore, direct comparison between our score and the ESA score reported in [5] might not be accurate, and a closer analysis is required. One of the best reported scores on this data set (apart from ESA) belongs to LSA, with 0.56 Spearman correlation, which is lower than our best results.

B. Document Similarity

The document similarity data set gathered by Lee et al [13] was also used as a task. The set contains 50 documents with average human similarity scores for each pair of documents. Table IV shows the Pearson correlation of the result of random walks with human judgments. In this experiment, we mapped each document to the 1,000 closest concepts in the network. We used the same random walk parameters which we had used for the word similarity task.

TABLE IV
PEARSON CORRELATION BETWEEN RANDOM WALK RESULTS AND HUMAN JUDGMENTS ON DOCUMENT SIMILARITY

Link type	**PPR**	**VP**
Hyperlink graph	.578	.667
Content similarity graph	.629	.624
Category graph	.679	.648
Template call graph	.477	.652
0.25 0.25 0.25 0.25	.611	.671
0.4 0.4 0.1 0.1	.614	.673
0.3 0.3 0.3 0.1	.609	.670
(0.7 0.1 0.1 0.1)2 - (0.2 0.6 0.1 0.1)3	.668	**.680**
(0.2 0.6 0.1 0.1)2 - (0.7 0.1 0.1 0.1)3	.644	.657
(0.5 0.1 0.4 0)2 - (0.3 0.6 0.1 0)3	**.681**	.676

Again, we examined the correlation between different link types but due to the space constraints we cannot provide the complete results here. Category and content links showed higher correlation in comparison to other link types. We show however results of some different combinations of links and some two-stage random walks in Table IV. It is interesting to see that the best two-stage random walk combination is similar to the one for word similarity, i.e. walking first mainly on hyperlinks and then on content links. The best reported result in Lee et al [13] belongs to LSA with 0.6 correlation, which is slightly lower than most of our results.

C. Paraphrase Detection

The Microsoft Paraphrase Corpus was used for a third series of experiments. We used only the test set, which contains 1,725 pairs of statements. We applied our methods it on each pair of statements, and sorted pairs by their semantic similarity scores. We compute mean average precision (MAP) for sorted list of pairs. A high MAP value means that scores of paraphrase pairs are higher in comparison with no-paraphrase pairs. We give the result of MAP scores for different random walks in Table V. All the parameters are the same as in the previous subsection.

To compare our results with the state-of-the-art method given in Mihalcea et al. [4] for this task, we give precision, recall and F-measure in Table VI for the PPR method on the concept network built using hyperlinks and then content

TABLE V
MEAN AVERAGE PRECISION ON THE MICROSOFT PARAPHRASE CORPUS

Link type	PPR	VP
Hyperlink graph	.818	**.793**
Content similarity graph	.789	.751
Category graph	.802	.762
Template call graph	.809	.768
0.25 0.25 0.25 0.25	.811	.778
0.40 0.15 0.20 0.25	**.835**	.793
(0.7 0.1 0.1 0.1)2 - (0.2 0.6 0.1 0.1)3	.815	.788
(0.2 0.6 0.1 0.1)2 - (0.7 0.1 0.1 0.1)3	.806	.774

similarity. In this case we assume a pair of statements is a paraphrase if the returned score is greater than 0.5. Our results are comparable to those obtained by Mihalcea et al. [4]: the "combined" line in Table VI is the average of different knowledge based and corpus based scores [4]. In comparison with the semantic similarity from [4], our approach scales better, because the former method must compute word similarity between all pairs of words in two text fragments. Another interesting observation is that PPR on hyperlinks graph has higher recall, while on the content similarity graph it has higher precision. One possible explanation is that the content similarity graph shows more accurate relations based on exact words between articles, while the hyperlinks graph expands relations between articles and helps the recall score.

TABLE VI
PRECISION, RECALL AND F-MEASURE OF DIFFERENT METHODS ON MICROSOFT PARAPHRASE CORPUS

Method	Precision	Recall	F
Lexical matching	71.6	79.5	75.3
LSA	69.7	95.2	80.5
Combined	69.6	97.7	81.3
Personalized page rank on hyperlinks	68.7	96.5	80.3
Personalized page rank on Content similarity graph	70.3	89.7	78.8

VII. CONCLUSION AND FUTURE WORK

In this work we proposed a general framework for text semantic similarity based on knowledge extracted from Wikipedia. We tested our approach on three different benchmark data sets, and found that results were competitive with state-of-the-art results on each set, obtained through methods particular to each task. We built a unique graph from Wikipedia articles and extracted four different link structures between concepts. Results of random walks on different link structures are different, and but combining them gives better results on each task. We also examined three different methods to estimate distances based on random walks: hitting time, visiting probability, and personalized page rank. The similarity measures derived from them can be different as they measure different properties of paths between concepts. One advantage of our approach is that the update of data set and the graph is easy without additional cost of recomputation. Also, using k-nearest neighbors graphs to capture different knowledge makes the computation possible for a large corpus such as the English Wikipedia. Finding the optimum weights to combine different link types is still a challenging issue that should be investigated more in future using an appropriate training procedure. Another future task is to design a better way to filter articles and links in Wikipedia. Different filtering methods lead to different final set of concepts and different final graph that have strong effects on the results.

ACKNOWLEDGMENT

This work is supported by the Swiss National Science Foundation through the NCCR on Interactive Multimodal Information Management (IM2), `http://www.im2.ch`. The authors are grateful to Michael D. Lee for access to the textual similarity dataset.

REFERENCES

[1] C. Fellbaum *et al.*, *WordNet: An electronic lexical database*. Cambridge, MA: The MIT Press, 1998.

[2] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[3] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of SIGIR 1999 (22nd Annual International ACM Conference on Research and Development in Information Retrieval)*, Berkeley, CA, 1999, pp. 50–57.

[4] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proceedings of AAAI 2006 (21st National Conference on Artificial Intelligence)*, Boston, MA, 2006, pp. 775–782.

[5] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *Proceedings of IJCAI 2007 (20th International Joint Conference on Artificial Intelligence)*, Hyderabad, India, 2007, pp. 6–12.

[6] D. Milne and I. H. Witten, "Learning to link with Wikipedia," in *Proceedings of CIKM 2008 (17th ACM Conference on Information and Knowledge Management)*, Napa Valley, CA, 2008, pp. 509–518.

[7] E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa, "Wikiwalk: random walks on wikipedia for semantic relatedness," in *Proceedings of TextGraphs-4 (2009 Workshop on Graph-based Methods for Natural Language Processing)*, Singapore, 2009, pp. 41–49.

[8] T. H. Haveliwala, "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, pp. 784–796, 2003.

[9] Metaweb Technologies, "Freebase wikipedia extraction (WEX)," http://download.freebase.com/wex/, 2010.

[10] P. Sarkar and A. Moore, "A tractable approach to finding closest truncated-commute-time neighbors in large graphs," in *Proceedings of UAI 2007 (23rd Conference on Uncertainty in Artificial Intelligence)*, Vancouver, BC, 2007.

[11] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: The concept revisited," *ACM Transactions on Information Systems*, vol. 20, no. 1, pp. 116–131, 2002.

[12] K. Collins-Thompson and J. Callan, "Query expansion using random walk models," in *Proceedings of CIKM 2005 (14th ACM Conference on Information and Knowledge Management)*, Bremen, Germany, 2005, pp. 704–711.

[13] M. Lee, B. Pincombe, and M. Welsh, "An empirical evaluation of models of text document similarity," in *Proceedings of CogSci 2005 (27th Annual Conference of the Cognitive Science Society)*, Stresa, Italy, 2005, pp. 1254–1259.