SemCiR
A citation recommendation system based on a novel semantic distance measure

Fattane Zarrinkalam and Mohsen Kahani
Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Received December 2011
Revised July 2012
August 2012
August 2012
Accepted September 2012

Abstract
Purpose – The purpose of this paper is to propose a novel citation recommendation system that inputs a text and recommends publications that should be cited by it. Its goal is to help researchers in finding related works. Further, this paper seeks to explore the effect of using relational features in addition to textual features on the quality of recommended citations.

Design/methodology/approach – In order to propose a novel citation recommendation system, first a new relational similarity measure is proposed for calculating the relatedness of two publications. Then, a recommendation algorithm is presented that uses both relational and textual features to compute the semantic distances of publications of a bibliographic dataset from the input text.

Findings – The evaluation of the proposed system shows that combining relational features with textual features leads to better recommendations, in comparison with relying only on the textual features. It also demonstrates that citation context plays an important role among textual features. In addition, it is concluded that different relational features have different contributions to the proposed similarity measure.

Originality/value – A new citation recommendation system is proposed which uses a novel semantic distance measure. This measure is based on textual similarities and a new relational similarity concept. The other contribution of this paper is that it sheds more light on the importance of citation context in citation recommendation, by providing more evidences through analysis of the results. In addition, a genetic algorithm is developed for assigning weights to the relational features in the similarity measure.

Keywords Citation recommendation, Citation analysis, Semantic distance, Bibliographic dataset, Semantics, Information management, Bibliographies

Paper type Research paper

1. Introduction
A major task every researcher is involved in is to acquire appropriate knowledge about the current state of his research area by finding the most related works. Traditional search approaches, e.g. Googling or using digital libraries like CiteSeer, generally have the characteristic that they are very sensitive to the input keywords, and they focus on finding documents that are textually similar to these keywords. Consequently, they have limitations in finding the papers related to the researcher's subject of interest (Henzinger *et al.*, 2003).

Another approach for addressing this task is to use a citation recommendation system. The input of such system is a text, and it recommends a short list of publications that should be cited by that text. Since citations of a text are related to it, a citation recommendation system actually focuses on finding the related publications and not just textually similar ones (Strohman *et al.*, 2007). Currently, there are a

Program: electronic library and information systems
Vol. 47 No. 1, 2013
pp. 92-112
© Emerald Group Publishing Limited
0033-0337
DOI 10.1108/00330331311296320

number of works dedicated to citation recommendation systems (Bethard and Jurafsky, 2010; He *et al.*, 2011; Tang and Zhang, 2009).

This paper is formed around the idea that relying only on the textual features of publications (e.g. its title and abstract) is not enough for finding the related works. It is helpful to utilize the relational features, i.e. features that indicate the relations of publications to each other, as well. For instance, it is possible that two publications have low textual similarity, but since they are published in the same venue, or are simultaneously cited by many publications, they can be considered to be interrelated.

In this paper, SemCiR (Semantic Citation Recommendation System) is proposed, which is a citation recommendation system based on a novel semantic distance measure. The contributions of this work are:

- Presenting a new relational similarity measure for computing the relatedness of two publications. This measure is based on six different types of relations between publications.
- Introducing a novel semantic distance measure based on the combination of the textual and relational similarity measures. Further, a citation recommendation system is proposed that uses this measure to compute the semantic distance between a text and the candidate publications.
- Using a genetic algorithm to identify the importance of different relational features in the proposed citation recommendation system.
- Demonstrating the impact of the citation context as a textual feature, on improving the quality of recommendation results.

The rest of the paper is organized as follows. Section 2 briefly reviews the related works. The proposed approach is described in section 3. Section 4 is dedicated to the evaluation, and finally, section 5 concludes the paper and provides some directions for future work.

2. Related work

In this section, the related works on citation analysis as well as citation recommendation are briefly reviewed.

2.1 Citation analysis

A citation is "the acknowledgment that one document receives from another" (Smith, 1981). In general, citation analysis deals with the analysis of relations between a citing document and the document it cites. Many concepts have been used in the citation analysis, for instance:

- Bibliographic coupling: two documents that have at least one common reference are said to be bibliographically coupled (Kessler, 1963).
- Co-citation analysis: two documents are said to be co-cited if there is a third document that cites both of them (Small, 1973).
- Citation context: citation context is the piece of text that the citation is placed inside. A citation context provides an explicit description of the cited work from the point-of-view of the citing author (Small, 1982).

Different applications of citation analysis are described in (Smith, 1981). For instance, Garfield (1972) introduced a metric called Journal Impact Factor (JIF) for evaluating quality of journals. It computes the average frequency with which a journal's papers are cited. The limitations of JIF are discussed by Seglen (1997), and there are works attempting to address these issues. For instance, Papavlasopoulos (2010) used artificial neural networks to combine various bibliometric factors like JIF, the immediacy index, and the cited half-life index. Zhang (2011) integrated bibliometric factors with those that use structural position of journals, like PageRank algorithms, to evaluate the influence of journals. H-index (Hirsch, 2005) is a measure of productivity and the impacts of the papers of an author. Since it is based on a row citation count and do not consider the impact of cited papers and self-citations, Szymanski *et al.* (2012) proposed a more accurate measure of impact for author ranking.

Another area in which the citation analysis has been used is information retrieval. Ritchie *et al.* (2008b) combined the terms of a given document and its citation contexts to index that document and concluded that it improves the retrieval performance in comparison with indexing only the terms of a document. Fujii (2007) and Strohman *et al.* (2007) confirmed that using citation information, in addition to text, generates better results in document retrieval. Citation analysis is also used for improving classification and clustering methods (Aljaber *et al.*, 2010; Couto *et al.*, 2010).

The current paper uses citation analysis for measuring the similarity of publications in a citation recommendation system.

2.2 Citation recommendation systems

Based on their purpose, current citation recommendation systems can be divided into three categories. The first category tries to complete the citation list of an input text that some of its citations have already been specified by the author.

For example, McNee *et al.* (2002) proposed an approach that uses collaborative filtering (Adomavicius *et al.*, 2005). Their recommendation algorithm analyses the citation graph and builds different rating matrices. Torres *et al.* (2004) presented different techniques for building a hybrid recommender system (Burke, 2002) that uses collaborative filtering and context-based filtering algorithms. They concluded that hybrid algorithms generate better recommendations than non-hybrid ones.

The second category of works, which the current paper belongs to, receives a text as input, and generates the recommended list of its citations. Strohman *et al.* (2007) used a two-step recommendation algorithm. The first step produces a candidate list of citations, and the second step ranks the candidates to find the best ones. A linear combination of text features and citation graph attributes like publication year, citation count, and Katz distance (Liben-Nowell and Kleinberg, 2003) is used for ranking. They concluded that Katz distance, which is a measure of distance in a graph, is the most important feature.

Tang and Zhang (2009) utilized a topic-based approach based on a two-layer Restricted Boltzmann Machine model, called RBM-CS. It discovers the topic distributions of papers and the citation relationships, simultaneously. The Boltzman machine, after learning these relations, is used for recommending citations for the input text.

Bethard *et al.* (2010) introduced some features, such as topic similarity and author behavioral patterns, and used the weighted sum of these features as a model to retrieve the citations of the input text. The weights for these features are achieved through a learning algorithm.

The proposed approach of He *et al.* (2010) consists of two phases: candidate set generation and ranking. The candidate set is generated by a number of heuristics based on features like common authors, and abstract similarity. Ranking is performed according to the context-aware textual similarity between the input text, and the text of each publication in the candidate set.

The third category of citation recommendation systems tries to recommend citations for specific locations within the input text. He *et al.* (2010), in addition to presenting an algorithm for the global recommendation, proposed an approach which recommends citations for the specific locations marked by a "[?]" string in the input. The candidate set generation in this approach is similar to the global recommendation. However, a ranking algorithm is performed according to the similarities between citation contexts (i.e. the text around the [?] mark) in the input text, and the publications of the candidate set. Later, they extended their work in (He *et al.*, 2011), so that citation marks ([?]) are not required in the input text, and the recommendation algorithm is capable of finding the appropriate locations for citations.

Tang and Zhang (2009), after recommending citations for the whole input text, assign some of them to each sentence in the input text, according to a retrieval model.

3. The proposed approach

Generally speaking, every recommender system is composed of three main elements (Burke, 2002):

1. background data: the data that the system has before the recommendation process begins;
2. input data: the data that is given to the system in order to get recommendations from it; and
3. recommendation algorithm: the algorithm that processes the background and input data, and generates recommendations.

In this section, the proposed approach is described in terms of these elements.

3.1 Background data

Background data is composed of a dataset and the results of the preprocessing performed on this dataset. The dataset contains data about N_P distinct publications P_i ($1 \leq i \leq N_P$), so that:

$$P_i = (Id_i, T_i, abs_i, refList_i, citList_i, citctxList_i, authList_i, V_i, year_i)$$

Where:

Id_i:	A unique number assigned to P_i.
T_i:	Title of P_i.
abs_i:	Abstract of P_i.
$refList_i$:	$\{ P_j \mid P_i$ references $P_j \}$.
$citList_i$:	$\{ P_j \mid P_i$ is cited by $P_j \}$.
$citctxList_i$:	$\{ctx_j \mid$ there is a reference to P_i, in context $ctx_j\}$.

authList$_i$:	List of the authors of P_i.
V_i:	Venue (i.e. conference or journal) of P_i.
year$_i$:	Publication year of P_i.

Since the recommendation algorithm involves searching within the texts of publications, a text preprocessing is required to improve the efficiency of the recommendation algorithm. This is performed by building an index over the texts of the publications. For each publication P_i in the dataset, *text*$_i$, which is composed of one or more elements from $\{T_i, abs_i, ctxList_i\}$, is indexed.

3.2 Input data

The input data is a piece of text, composed of a number of paragraphs. Although it may include different elements, such as title and abstract, the proposed approach treats the whole input as a single unit and generates a list of publications that the input text should cite.

3.3 Recommendation algorithm

The proposed recommendation algorithm has two steps. The first step generates a set of candidate publications, and the second step ranks this set to generate the final list of recommendations. In the following sections, these two steps are described in detail.

3.3.1 Step1: candidate set generation. To generate the candidate set, first, an initial candidate set is created and then extended by adding more publications.

Initial candidate set generation. Since the input data of the recommendation algorithm is only raw text and it does not have features like list of authors, references, and venue, the algorithm can only use textual features to generate an initial candidate set of publications, *initCandSet*. This set is created by selecting a constant number *C* of publications that have the most textual similarity to the input text.

The index built in the preprocessing phase is utilized in this process and a function *textSim(input, P$_i$)* is defined for calculating text similarity between the input text and *text*$_i$, for each publication P_i $(1 \leq i \leq N_P)$. The details of this function are described in section 0.

Candidate set extension. After *initCandSet* is generated, it includes *C* publications that each of them, in addition to the textual features, has relational features like list of authors, venue, and list of references. These features can be used to define different relations between publications.

In this paper, six different relations, $R_1, R_2, \ldots, R_6$, are defined from publication P_i to publication P_j as below:

$$R_1 : P_i \in citeList_j$$

$$R_2 : P_i \in refList_j$$

$$R_3 : authList_i \cap authList_j \neq \emptyset$$

$$R_4 : V_i = V_j$$

$$R_5 : refList_i \cap refList_j \neq \emptyset$$

$$R_6 : citList_i \cap citList_j \neq \emptyset$$

It is possible to use these relations for extending the initial candidate set. This extension process starts by adding the elements of *initCandSet* to an empty candidate set, *candSet*. Then, for every publication P_i in the *initCandSet*, its neighbors from the background data are added to the *candSet*. Two publications P_i and P_j are said to be neighbors, if and only if, at least one of the six relations exists between them. After this extension process, the candidate set has N_C publications.

It worth describing the idea behind the relations defined above. If there is a relation of R_1 or R_2 between two publications P_i and P_j, it evidences some level of relatedness between them, because the authors of one publication (P_i in case of R_1, and P_j in case of R_2) have explicitly confirmed the relatedness by giving a reference to the other publication. Further, since publications of an author or a venue are usually focused on a specific domain, they can be considered as related. This is the idea behind R_3 and R_4. Relations R_5 and R_6 are based on the concepts of bibliographic coupling and co-citation analysis, correspondingly, which are two core concepts in the citation analysis (see section 0).

3.3.2 Step 2: ranking. After generating the candidate set, the ranking is performed, which uses a novel approach to rank the elements of the candidate set based on their semantic distances from the input text. This approach uses the concepts of textual similarity computed during the initial candidate set generation, and relational similarity described below.

Relational similarity. Using the six relation types introduced in section 0, the relational similarity of two publications P_i and P_j is defined by Formula 1:

$$relSim(P_i, P_j) = \frac{1}{6}\sum_{k=1}^{6} W_k F_k(P_i, P_j) \tag{1}$$

where F_k $(1 \leq k \leq 6)$ is a function that returns a value in the interval [0, 1] as the similarity of publications P_i and P_j based on the corresponding relation R_k. Further, W_k $(1 \leq k \leq 6)$ is the weight assigned to the function F_k. These six functions are defined as:

$$F_1(P_i, P_j) = \begin{cases} 1 \text{ if there is a relation } \mathrm{R}_1 \text{ from } \mathrm{P}_i \text{ to } \mathrm{P}_j \\ 0 \text{ otherwise} \end{cases}$$

$$F_2(P_i, P_j) = \begin{cases} 1 \text{ if there is a relation } \mathrm{R}_2 \text{ from } \mathrm{P}_i \text{ to } \mathrm{P}_j \\ 0 \text{ otherwise} \end{cases}$$

$$F_3(P_i, P_j) = \frac{|authList_i \cap authList_j|}{|authList_i \cup authList_j|}$$

$$F_4(P_i,P_j) = \begin{cases} 1 \text{ if there is a relation } \mathrm{R}_4 \text{ from } \mathrm{P}_i \text{ to } \mathrm{P}_j \\ 0 \text{ otherwise} \end{cases}$$

$$F_5(P_i,P_j) = \frac{|refList_i \cap refList_j|}{|refList_i \cup refList_j|}$$

$$F_6(P_i,P_j) = \frac{|citList_i \cap citList_j|}{|citList_i \cup citList_j|}$$

The return value of 0 for functions F_1, F_2 and F_4 indicates the absence of the corresponding relation between the two publications, while the return value of 1 indicates its existence.

Functions F_3, F_5, and F_6 are respectively associated with relations R_3, R_5, and R_6, and they are based on similar ideas. Therefore, only F_3 is described here. The more common authors two publications have, the more related they could be considered. However, the ratio of the common authors to the total number of authors of the two publications is also important. For instance, two common authors out of three authors indicate a stronger relation in comparison with two common authors out of six. Therefore, the return value of F_3 is directly proportional to the number of common authors between them, and also inversely proportional to the total number of their distinct authors.

It is reasonable to consider assigning a weight to each of these functions, to cope with the fact that their corresponding relations have different contributions to the overall similarity of two publications. There are different methods for assigning these weights. For instance, an expert can determine these values based on his judgment about the importance of each relation. Another approach is to use an evolutionary algorithm. In this work, a genetic algorithm has been developed for this reason. It is described in section 0.

Semantic distance. In order to rank the elements of the candidate set, the semantic distances between the input text and each publication P_i in the candidate set are computed using Formula 2:

$$semanticDist(input,P_i) = \frac{1}{1+semanticSim(input,P_i)} \tag{2}$$

where *semanticSim(input,*P_i*)*, the semantic similarity of the input text and publication P_i, is computed by Formula 3.

$$semanticSim(input,P_i) = \frac{1}{C}\sum\nolimits_{j=1}^{C}\left[relSim(P_i,Q_j)\times normTextSim(input,Q_j)\right] \tag{3}$$

where Q_j is a publication from the *initCandSet*, C is the size of the *initCandSet*, and *normTextSim(input,* Q_j*)* is the normalized textual similarity between input text and Q_j, as defined by Formula 4.

$$normTextSim(input,Q_j) = \frac{textSim(input,Q_j)}{\max\limits_{(1\leq K\leq C)} textSim(input,Q_k)} \tag{4}$$

The idea behind Formula 3 can be described using Figure 1. In this figure, a weighted directed graph G = (V, E) is depicted, which is in fact the result of the candidate set generation phase. There is a vertex in *G* for each of the publications of the *initCandSet*, candSet, and also for the input text.

Since each publication Q_j from the *initCandSet* is selected based on its textual similarity with the input, there is an edge from the node corresponding to Q_j, i.e. v_{qj}, to the node corresponding to the input, i.e. v_{input}. The weight of this edge is equal to *normTextSim(input, Q_j)*.

Further, since each publication P_i from the *candSet* is a neighbor of some publication Q_j from *initCandSet*, there is an edge from the node corresponding to P_i, i.e. v_{pi}, to v_{qj}. The weight of this edge is equal to *relSim(P_i, Q_j)*.

In order to rank each publication P_i of the *candSet*, it is required to compute its similarity to the input. As illustrated in Figure 1. there is no edge from v_{pi} to v_{input}. However, there is a set of *C* paths of the form $[v_{pi}, v_{qj}, v_{input}]$, from v_{pi} to v_{input}.

For each publication P_i from *candSet*, its similarity to the input is calculated through each of the *C* paths from v_{pi} to v_{input}, by multiplying the weight of the edges $< v_{pi}, v_{qj} >$ and $< v_{qj}, v_{input} >$. Finally, the average of these *C* values is used as the semantic similarity of P_i to the input, i.e. *semanticSim(input, P_i)*.

The reason behind multiplying weights of the two edges is that it appropriately computes the contribution of *relSim(P_i, Q_j)* to the semantic similarity of P_i to the input, through the path $[v_{pi}, v_{qj}, v_{input}]$. In the other words, the normalized textual similarity acts as a weight for the relational similarity.

Finally, the rank of the elements of *candSet* is determined by sorting them based on their semantic distances from the input text, in ascending order. Then, the first *M* elements are returned as the recommended citations.

4. Evaluation

SemCiR has been implemented in Java and experimentally evaluated. In this section, these experiments are discussed. The experiments are conducted on a laptop with a

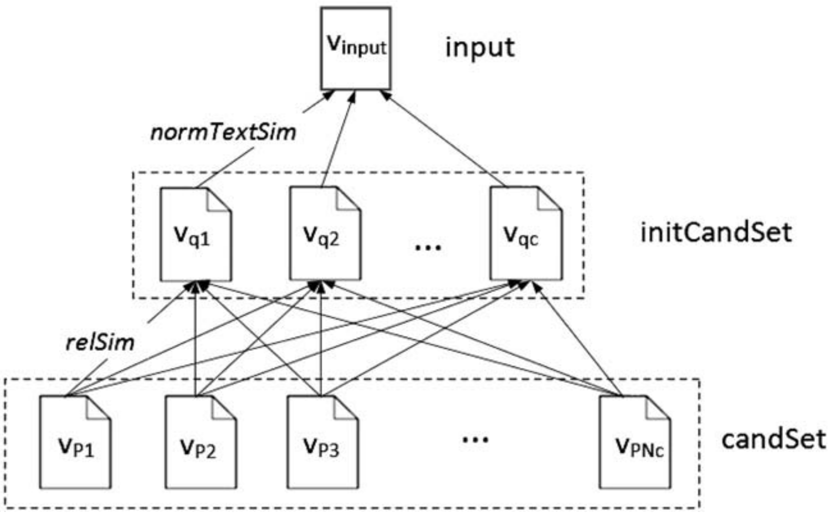

Figure 1.
Relations between the elements of initCandSet, candSet, and input

CPU of Intel Corei5 2.27 GHz, 3.00 GB RAM, and Microsoft Windows 7 operating system.

In this section, first, the setup of the experimental environment is described, and then, both automatic evaluation and manual evaluation are discussed.

4.1 Experiment setup

Various tasks are performed to prepare the experimental environment. They are described in this section in terms of the three main elements of SemCiR, i.e. background data, input data and recommendation algorithm.

4.1.1 Background data. The background data is a bibliographic dataset collected from *CiteSeerX* (http://citeseerx.ist.psu.edu). A crawler has been developed that uses *OAI* service (http://citeseerx.ist.psu.edu/oai.html) of *CiteSeerX* for information harvesting. However, since some details, e.g. citation contexts, are not provided by *OAI* service, the crawler browses web pages of *CiteSeerX* to automatically gather these details.

After collecting data of about 30,000 publications, a filtering was performed to remove: publications published after 2007 (texts of these publications are later used as the input data in the experiments), and publications with many missing values (e.g. publications whose abstract and title were missing). This filtering process led to about 12,000 publications stored in a *MySQL* database. Some statistics about the resulting dataset are presented in Table I.

Since the proposed approach uses textual similarity, a text pre-processing is performed to prepare the dataset for future text processing. This preprocessing is performed using the indexing component of *Lucene* library (http://lucene.apache.org). In the experiments, each publication P_i from the dataset is indexed over Id_i and $text_i$.. Further, a customized *Lucene* analyzer is used for the purpose of tokenization. It performs basic operations like removing numbers and stop words, and stemming.

4.1.2 Input data. In order to evaluate the proposed recommendation algorithm, publications published in 2008, 2009 or 2010 have been selected from the collected data as the test data. Each of these publications P_i is considered as a pair of the form:

$\mathrm{P_i} = (\mathrm{Text_i}, \mathrm{refList_i})$.where $Text_i$ is the concatenation of T_i, abs_i, and $refctxList_i$, and $refctxList_i$: $\{ctx_i \mid$ there is a reference to P_j, in context ctx_i inside $P_i\}$

Then, a filtering is performed to remove each publication that has its title or abstract is missing, or its *refctxList* has less than three elements, or its *refList* has less than five elements.

Total number of publications (N_p)	12,000
Number of citations (N_R), i.e. P_i cites P_j	57,225
Number of cited publications	7,182
Average number of citations per publication	8
Average number of references per publication	6
Average number of publications of each author	2
Average number of authors of each publication	3
Average number of publications of each venue	3
Average length of abstracts (words)	180
Average length of titles (words)	11

Table I.
Some statistics about the dataset used in experiments

For each publication P_i from about 550 remaining publications, $Text_i$, has been used as the input data of the recommender system, and $refList_i$ has been used as the expected output in the experiments.

4.1.3 Recommendation algorithm. This section describes preparation of different elements of the recommendation algorithm.

- *Text similarity*. The function *textSim(intput, P_j)* mentioned in Section 0, is used in the initial candidate set generation phase. It was implemented by the searching component of *Lucene*. Given the input text, it uses the same analyzer which has been used for indexing, to split the input text into a token stream. Then, this stream is used to build a *BooleanQuery*. This query is executed on the index to retrieve the list of *C* publications from the dataset, ordered according to their textual similarities with the input text, which is calculated by the Lucene's default text similarity. This list is used as the input to the candidate set extension step.
- *Parameter M*. After candidate set extension, the ranking process selects *M* publications from the candidate list. In the experiments, ten different values ranged over [25, 250] were used for *M*.
- *Parameter C*. In order to determine the appropriate value of *C*, a simple experiment was conducted. A set of 100 publications were randomly chosen from the test data, and they were used for executing the recommendation algorithm with different values of *C* from the interval [5, 50], and with $M = 25$. For each value of *C*, the average of the execution time, *recall*, cocited_*probability*, and *NDCG* were calculated over the 100 publications. These metrics are described in Section 0. The results are shown in Figures 2-5. As indicated in these figures, the greater is the value of *C*, the longer is the execution time, and generally the results are better, in terms of the three evaluation metrics. It is interesting to note that for $C = 25$, there is a considerable growth in the *recall*, cocited_*probability* and *NDCG*, while the execution time has not yet increased drastically. Therefore, in order to support online recommendation, the value of *C* was set to 25, to create a balance between execution time and the evaluation metrics.
- *Parameter N_C*. The proposed approach does not impose an explicit limit on N_C, i.e. the size of the candidate set. In the candidate set generation phase, all the

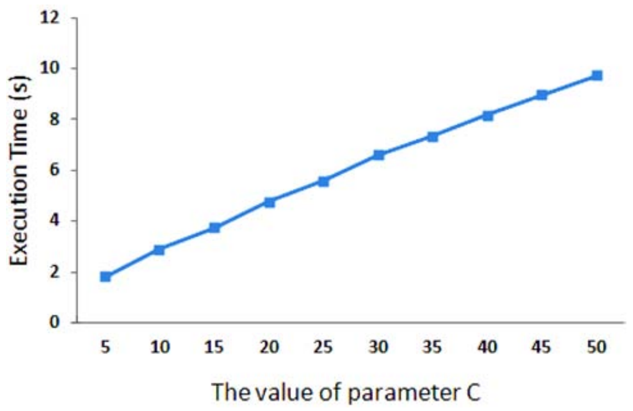

Figure 2.
Execution time for various values of C

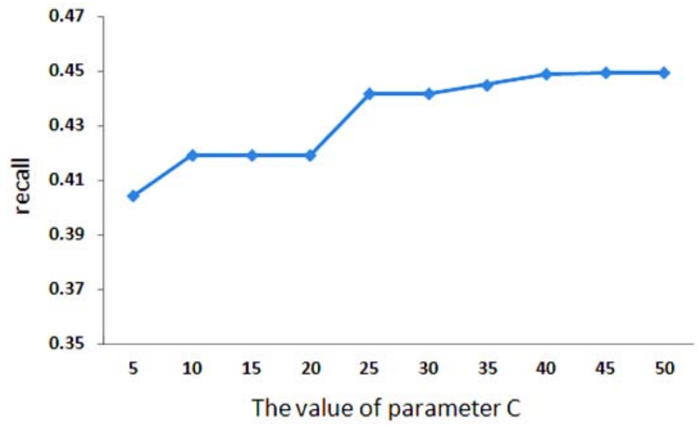

Figure 3.
Recall for various values of C

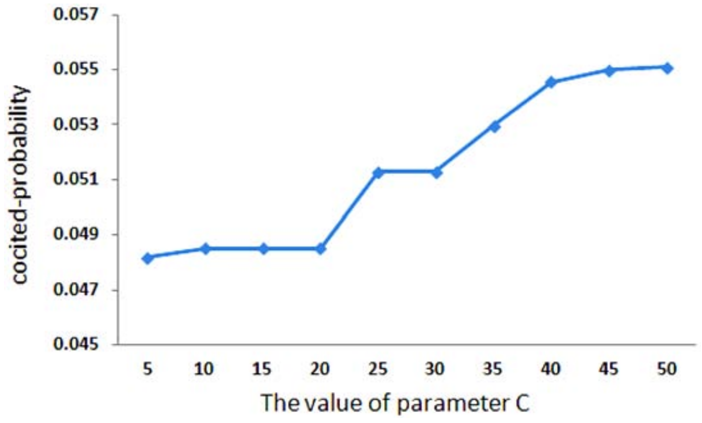

Figure 4.
Cocited_probability for various values of C

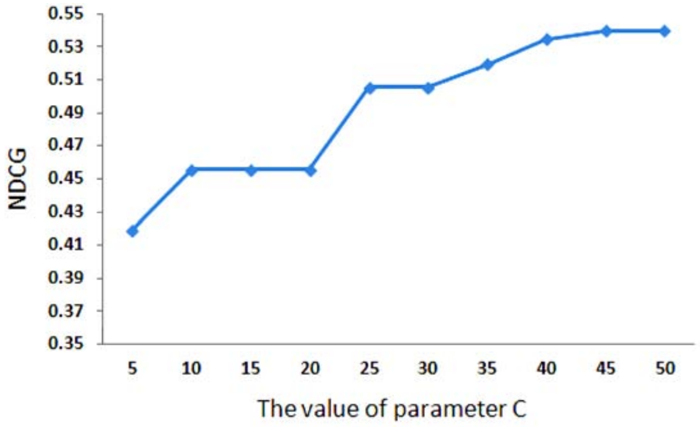

Figure 5.
NDCG for various values of C

background publications must be checked with regard to the six relations, and their neighbors are added to the candidate set. There is no need to limit the candidate set (i.e. ignore some neighbors), as the cost of relation checking for every publication has been already paid. The smaller is the candidate set, the faster is ranking phase. However, since the ranking phase has a lightweight computation (compared to the candidate set generation), limiting the size of the candidate set, after it is generated, does not have a considerable overall benefit. It might be worth noting that in the experiments, the average size of the candidate set has been 2,921 and its standard deviation has been 1,300.

- *Weight assignment.* A genetic algorithm was developed for determining the weights of the six functions F_1, $F_2, \ldots, F_6$. In this GA, each chromosome has six genes, each for indicating one of the six weights. The first generation includes 50 chromosomes with random values in the interval [0, 1]. To calculate fitness of each chromosome, the value of its genes are used as the weights of the functions, and the recommendation algorithm is executed on a small set (about 50 inputs) of input data. Then, the output of the algorithm is evaluated in terms of the three metrics (described in Section 0), and the sum of the normalized values of these metrics is used as the fitness value of the corresponding chromosome.

After running the genetic algorithm for 20 generations, the best chromosome has been used as the output of the weight assignment process. Table II shows this result.

4.2 Automatic evaluation

Usually, in order to automatically evaluate a citation recommendation algorithm, the text of a test publication is given to it, and the list of top *M* recommendations is compared with the list of actual references of that publication. In this section, such an approach is used to compare the proposed approach with other alternatives.

4.2.1 Metrics. Three metrics are used in the automatic evaluation. In these metrics, $recList_i$ is the list of top-M recommendations generated by the recommendation algorithm for a test publication P_i.

Recall. For each test publication P_i, recall_i is calculated by Formula 5 and then the recall of the system is calculated by computing the average recall over all the test publications.

$$recall_i = \frac{|refList_i \cap recList_i|}{|refList_i|} \tag{5}$$

	n
W_1	0.9
W_2	0.2
W_3	0.4
W_4	0.2
W_5	0.5
W_6	0.8

Table II. The weights of the six functions

Co-cited_probability. For a test publication P_i, it is possible that some recommended citations do not exist in $refList_i$, however, they might be acceptable or even better than the elements of $refList_i$. In most related works, an expert-based evaluation method is used for deciding the quality of such recommendations. However, in (He *et al.*, 2010, 2011) a new metric called *cocited_probability* is defined. For every two publications P_i and P_j, cocited_*probability* of P_i to P_j is computed as:

$$cocited\_probability(P_i, P_j) = \frac{|citList_i \cap citList_j|}{|citList_i|} \tag{6}$$

For each test publication $P_{i\text{-}}$, cocited_*probability*$_i$ is computed by Formula 7, and then the *cocited_probability* of the system is calculated by computing the average *cocited_probability* over all the test publications.

$$cocited\_probability_i = \frac{1}{Y \times L} \sum_{j=1}^{Y} \sum_{k=1}^{L} cocited\_probability(P_j, P_k) \tag{7}$$

where,

$$P_j \in refList_i \text{ and } Y = |refList_i|$$

$$P_k \in [recList_i - refList_i] \text{ and } L = |recList_i - refList_i|$$

NDCG. The quality of a recommendation algorithm not only depends on the quality of each recommended element, but also to their order. The Normalized Discounted Cumulative Gain (*NDCG*) is a well-known metric in information retrieval for considering this issue. He *et al.* (2010) has described *NDCG* in the context of citation recommendation systems. Their description has been used for measuring *NDCG* in the experiments.

4.2.2 Comparison. In the experiment, six different recommendation methods are compared with each other. These methods differ in terms of:

- the textual elements over which the index is created;
- candidate set generation method; and
- ranking method.

Next, these six methods are described. Their specification is briefly presented in Table III.

Table III.
A comparison of the methods' specifications

	Index components	Candidate set generation method	Ranking method
Method1	$\{T_i, abs_i\}$	Textual similarity (*Lucene*)	Textual similarity (*Lucene*)
Method2	$\{T_i, abs_i\}$	SemCiR	SemCiR
Method3	$\{T_i, abs_i\}$	SemCiR	SemCiR + GA
Method4	$\{T_i, abs_i, citctxList_i\}$	SemCiR	SemCiR + GA
Method5	$\{T_i, abs_i, citctxList_i\}$	SemCiR	(He *et al.*, 2010)
Method6	$\{T_i, abs_i, citctxList_i, refctxList_i\}$	(He *et al.*, 2010)	(He *et al.*, 2010)

(1) *Method1:* The index is created over $text_i = \{T_i, \text{abs}_\text{i}\}$. For each publication in the test data, $text_i$ is given as a query to the *Lucene* searching component and the results are considered as the ranked list of recommended citations. Therefore, the candidate set generation and the ranking is implicitly performed by *Lucene*, which uses only textual similarity.

(2) *Method2:* The index is created over $text_i = \{\text{T}_\text{i}, abs_i\}$. Candidate set generation and ranking are performed by the proposed approach, with the value of 1 for all W_k ($1 \leq \text{k} \leq 6$). It means that this method assigns equal importance to all the six relation types.

(3) *Method3:* This method is identical to Method2 except that it uses weights that are assigned by the genetic algorithm, as shown in Table II. This method is considered to evaluate the effect of considering different importance for different relation types.

(4) *Method4:* The index is created over $text_i = \{T_i, \text{abs}_\text{i}, citctxList_i\}$. The candidate set generation and ranking are the same as Method3. This method is considered to determine the effect of using citation contexts for improving the quality of recommendations. It must be noted that, the citation contexts provided by *CiteSeerX* are included in the background data. They have fixed length and according to the experiments of Ritchie *et al.* (2008a), a fixed length for citation context is the most effective choice with regard to information retrieval.

(5) *Method5:* The goal of this method is to specifically evaluate the proposed ranking method, and compare it to one of the recent related works. He *et al.* (2010) have presented a ranking method for citation recommendation, and have compared it with different alternatives, concluding that it is better than others. In Method5, the index is created over $text_i = \{T_i, \text{abs}_\text{i}, \text{citctxList}_\text{i}\}$, the candidate set generation method is the same as the one used in Method2, but ranking method is as described in (He *et al.*, 2010). This ranking method is based on a textual similarity measure which utilizes citation contexts.

(6) *Method6:* This method implements the approach presented by He *et al.* (2010). It is included in the experiments in order to compare the proposed approach with (He *et al.*, 2010), in terms of both the candidate set generation, and ranking methods. Here, the index is created over $text_i = \{T_i, \text{abs}_\text{i}, \text{citctxList}_\text{i}, refctxList_i\}$. He *et al.* (2010) have evaluated different candidate set generation methods, and have concluded that $LC100 + G1000$ method is the best one, based on the tradeoff between the two factors of coverage and candidate set size. For an input text $text_{input} = \{T_{input}, \text{abs}_\text{input}, \text{refctxList-}_\text{input}\}$, *LC100* identifies for each context *ctx* from $refctxList_{input}$, the top-100 publications P_i which $refctxList_i$ are the most similar publications to *ctx*. Further *G1000* means top-1000 publications P_i that T_i and abs_i are the most similar to $text_{input}$.

4.2.3 Result analysis. In this section, the six methods are compared to each other. Figures 6-8 represent the results. Further, a brief time analysis is provided.

Method1 and Method2 use the same attributes for indexing, but different recommendation algorithms. Method1 uses only textual similarity, while Method2 uses a combination of textual and relational similarity. By comparing these two methods it is possible to evaluate the role of relational similarity in the recommendation algorithm.

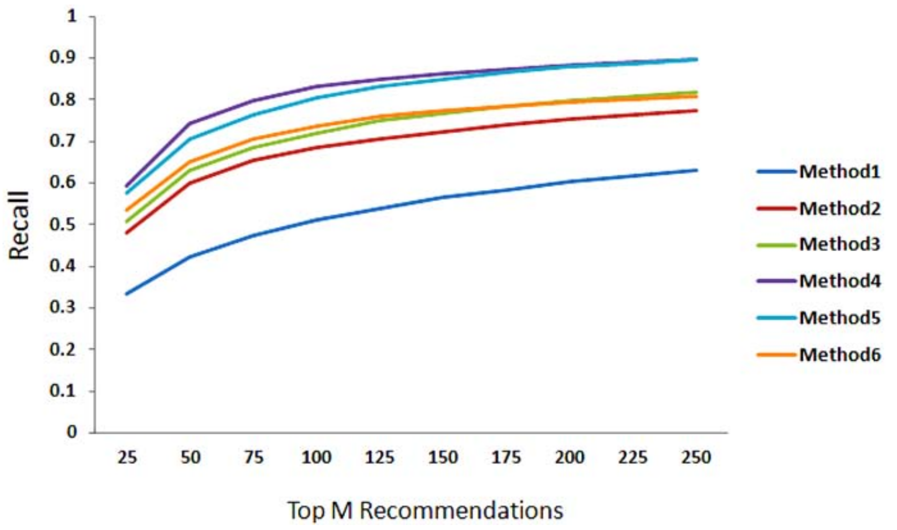

Figure 6.
Comparing the results in terms of recall

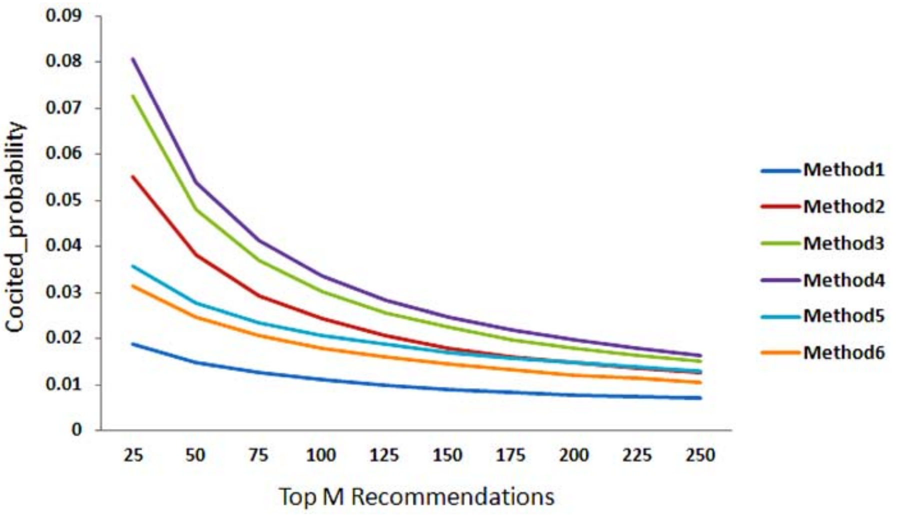

Figure 7.
Comparing the results in terms of cocited_probability

As illustrated in Figures 6-8, Method2 considerably outperforms Method1 in terms of all three metrics. It supports the idea that exploiting relational similarity, in addition to textual similarity, is effective in developing citation recommendation systems.

The reason is that the expected citations of a publication are not only publications that are textually similar to it, but also those that are related to it. One of the disadvantages of textual similarity is its weakness in finding the related publications, since two related publications might not have much similar text. Considering the relational similarity in addition to the textual similarity can remedy the effect of this weakness.

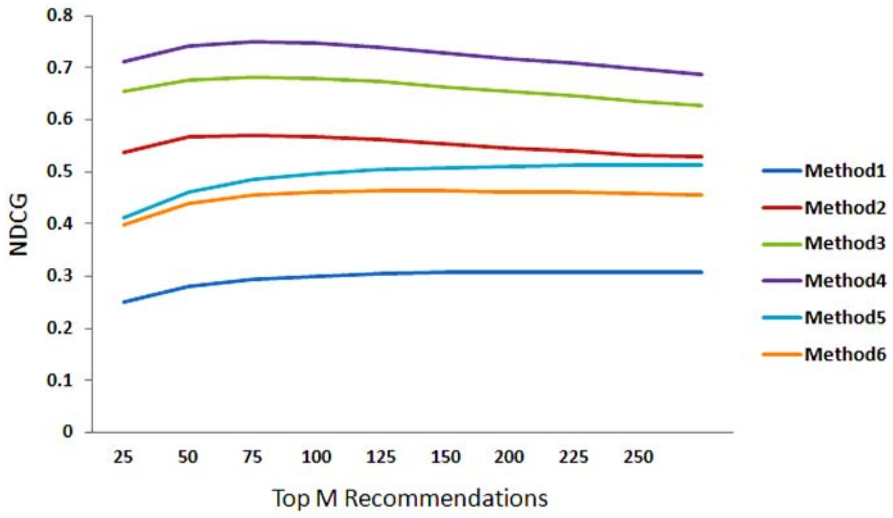

Figure 8.
Comparing the results in terms of NDCG

The difference between Method2 and Method3 is that Method3 uses GA-based weight assignment, while Method2 actualy does not use weights. As Figures 6-8 show, Method3 has outperformed Method2 in all the three metrics. This means that it is a good idea to differentiate between importance of different relation types, and the developed GA has been successful in determining the importance values.

Method3 and Method4 both implement SemCiR. The difference is that Method4 uses also citation context in addition to title and abstract. As shown in Figures 6-8, Method4 has generated better results. It stresses the positive effect of considering citation context in the proposed citation recommendation approach. The reason is that $citctxList_i$, i.e. the list of citation contexts of publication P_i, indicates contexts that other authors have considered as related to P_i. Therefore, if an input text is textually similar to these contexts, it can be considered as related to P_i, and it is reasonable that it cites P_i. In other words, using citation context in addition to title and abstract, helps to overcome the weakness of textual similarity in finding the related publications.

Method5 is used to compare the proposed approach of Method4 with a recent related work, only in terms of the ranking method. He *et al.* (2010) do not take the relational similarity into account, however, they use citation context. Method4 has improved all the three evaluation metrics. The reason is that Method4 uses citation context and relational similarity, which both improve the relatedness of the recommendations.

Method6 is the proposed approach of He *et al.* (2010), while Method5 uses the approach of SemCiR for candidate set generation, and the ranking method of He *et al.* (2010). Method5 uses both relational and textual similarity for generating candidate set, while Method6 uses only textual similarity. Therefore, by comparing Method5 and Method6 it is possible to evaluate the proposed candidate set generation method, and identify the effect of considering relational similarity. As illustrated in Figures 6-8, Method5 has better results. This means that the candidate set generated by the proposed method has better quality in terms of covering more related publications.

Method4 completely implements the proposed approach of this paper, and Method6 is the complete implementation of the approach presented by He *et al.* (2010). As illustrated in Figures 6-8, Method4 considerably outperforms Method6 in terms of all the three metrics. This again stresses the important role of the relational similarity in improving the quality of the recommended citations.

Finally, based on the comparison of the six methods, these points can be concluded:

- Combining the relational features with textual features leads to better results, compared to relying only on the textual similarity. This is evidenced by comparing Method1 and Method2.
- Using citation contexts in the textual similarity measure improves the quality of the recommendations. This is supported by comparing Method3 with Method4.
- A method which combines both textual and relational features, and also uses citation context, is better than a method which uses only textual features, including citation context. It is also better than a method which combines both textual and relational features but does not use citation context. This is shown by comparing Method4 with Method6, and also Method3 with Method4.
- Considering different importance for different relation types is helpful in improving the quality of recommedations. Comparison of Method2 and Method3 supports this fact.

Time analysis. It is interesting to compare the execution time of the three methods Method1 (the baseline), Method4 (SemCiR) and Method6 (He *et al.*, 2010) in terms of:

- total execution time;
- candidate set generation time; and
- ranking time.

The results presented in Table IV are averaged over all the test publications.

As expected, Method1 is the fastest one since it uses only *Lucene* textual similarity. However, as illustrated in Figures 6-8, its recommendations have low quality. Both Method4 and Method6 that have improved the recommendation quality, have higher execution times.

It is worth noting that the difference between the execution times of Method4 and Method6 is not considerable, and they both spend most of their time in generating the candidate set.

In order to evaluate the scalability of the proposed approach, a time complexity analysis is provided in the Appendix.

4.3 Manual evaluation

The problem of the automatic evaluation method is its circularity (Strohman *et al.*, 2007). It evaluates the quality of recommendations in terms of their matching with the list of actual references of the test publications. This implicitly means that the list of

Table IV. Time analysis results

	Total execution time (s)	Candidate set generation time (s)	Ranking time (s)
Method1	0.149	N/A	N/A
Method4	5.599	5.435	0.164
Method6	6.427	5.826	0.601

actual references is considered as the golden standard, while this cannot be generally considered as true.

The manual evaluation, in which the experts are asked to measure the appropriateness of the recommendations, doesn't have this problem. However, a full manual evaluation is time-consuming and labor-intensive. Here, an initial effort in manual evaluation of SemCiR is briefly discussed.

In this experiment, 50 test publications on the subject of the Semantic Web were selected and their text was used as the input of the recommendation algorithm. Three methods Method1, Method4 and Method6 have been separately used as the recommendation algorithm. Each test publication, along with its associated top 25 recommendations that are not in the actual reference list of the test publication, is given to the experts. The experts are three PhD students with at least five years of research experience on the Semantic Web. For each pair of test publication and recommendation, the experts are asked to give a score in the scale of 0 to 10, where 0 means no relevance, and 10 means full relevance between the recommendation and the test publication.

Table V shows the average, standard deviation and coefficient of variation (Hendricks and Robey, 1936) of the experts' scores for each of the three methods. Here, a great value of average along with a small value of standard deviation means that most of the recommendations have received good scores from the experts. Hence, the less is the value of coefficient of variation, the better is the quality of the recommendations.

As shown in Table V, Method4 (the proposed system), has better results than Method1 (the baseline), and Method6 (He *et al.* (2010)). Method1 and Method6 use only textual similarity, and analysis of results has shown that this has led to recommending some publications that are textually similar but semantically not relevant to the test publications. However, since Method6 uses also citation contexts, it has better results than Method1.

As an example, for the test publication titled "Executing SPARQL queries over the web of linked data", the 12th recommendation of Method1 is a publication titled "Supporting top-k join queries in relational databases". Due to its frequent use of terms like "query", "query processing", "query plan", "Top-k", it is almost textually similar to the test publication, however, the average score assigned to this recommendation by experts is 3 from 10, since they believe it has very low relevance to the subject of distributed SPARQL query processing over Linked Data.

5. Conclusion

In this paper, a new citation recommendation system, called SemCiR, is proposed that inputs a text and recommends publications that it should cite. Its goal is to help researchers in finding publications related to their research area.

	Relevance scores assigned by the experts		
	Average	Standard deviation	Coefficient of variation
Method1	6.46	2.37	0.37
Method4	8	1.63	0.2
Method6	6.86	2.27	0.33

Table V. Results of the manual experiment

To compute the relatedness of two publications, the proposed recommendation system employs a novel relational similarity measure, which is based on six different relational features of publications. Each feature is considered to have different contribution to the relational similarity, and therefore a genetic algorithm is developed to assign proper weights to these features. Further, a semantic distance measure is introduced that combines textual and relational similarity. This measure is used by the recommendation algorithm of SemCiR.

The experimental evaluation showed that:

- the proposed system generates better results, in comparison with methods which only use textual similarity;
- utilizing citation contexts improves the quality of the recommendations;
- considering different importance for the relational features improves the results; and
- the execution time of the recommendation algorithm is appropriate for online recommendation.

This was also confirmed by the time complexity discussion provided.

Additionally, an initial manual evaluation demonstrated that the quality of the recommended citations is promising, although it is interesting to conduct more exhaustive manual experiments. For instance, being able to ask authors of the test publications why they have not cited a specific recommended citation, can provide much help in improving the quality of the recommendation system.

Since the proposed relational similarity measure is promising in measuring the relatedness of publications, another future work is to assess its applicability in other contexts, for instance: to estimate similarity of journals, conferences, and experts based on the overall distance between their publications; or to detect possible research plagiarism, especially those that are not detectable by traditional text-only techniques (e.g. paraphrased and translated plagiarism).

References

Adomavicius, G. and Tuzhilin, A. (2005), "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions", *Journal of IEEE Transaction Knowledge Data Engineering*, Vol. 17 No. 6, pp. 734-49.

Aljaber, B., Stokes, N., Bailey, J. and Pei, J. (2010), "Document clustering of scientific texts using citation contexts", *Journal of Information Retrieval*, Vol. 13 No. 2, pp. 101-31.

Bethard, S. and Jurafsky, D. (2010), "Who should I cite? Learning literature search models from citation behavior", *ACM Conference on Information and Knowledge Management, Toronto, October 26-30*, pp. 609-618.

Burke, R. (2002), "Hybrid recommender systems: survey and experiments", *Journal of User Modeling and User-Adapted Interaction*, Vol. 12 No. 4, pp. 331-70.

Couto, T., Ziviani, N., Calado, P., Cristo, M., Gonçalves, M., Moura, E.S.D. and Brandão, W.C. (2010), "Classifying documents with link-based bibliometric measures", *Journal of Information Retrieval*, Vol. 13 No. 4, pp. 315-45.

Cutting, D.R. and Pdersen, J.O. (1997), "Space optimizations for total ranking", *Proceedings of RIAO'97, Computer-Assisted Information Searching on the Internet*, pp. 401-12.

Fujii, A. (2007), "Enhancing patent retrieval by citation analysis", *Proceedings of the ACM SIGMOD Conference on Management of Data, Beijing, China, June 12-14*, pp. 793-794.

Garfield, E. (1972), “Citation analysis as a tool in journal evaluation”, *Journal of Science*, Vol. 178 No. 1972, pp. 471-9.

He, Q., Kifer, D., Pei, J., Mitra, P. and Giles, C.L. (2011), “Citation recommendation without author supervision”, *Proceedings of WSDM’11, February 9-12, 2011, Hong Kong*, pp. 755-64.

He, Q., Pei, J., Kifer, D., Mitra, P. and Giles, C.L. (2010), “Context-aware citation recommendation”, *Proceedings of the 19th International World Wide Web Conference (WWW), The Raleigh Convention Center, Raleigh, NC*, pp. 421-30.

Hendricks, A.W. and Robey, K.W. (1936), “The sampling distribution of the coefficient of variation”, *Annals of Mathematical Statistics*, Vol. 7 No. 3, pp. 129-32.

Henzinger, M.R., Motwani, R. and Silverstein, C. (2003), “Challenges in web search engines”, *Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, 11 August*, pp. 1573-9.

Hirsch, J.E. (2005), “An index to quantify an individual’s scientific research output”, *Proceedings of the National Academy of Sciences*, Vol. 102 No. 46, pp. 16569-72.

Kessler, M. (1963), “Bibliographic coupling between scientific papers”, *Journal of American Documentation*, Vol. 14 No. 1, pp. 10-25.

Liben-Nowell, D. and Kleinberg, J. (2003), “The link prediction problem for social networks”, in *CIKM 2003, Proceeding of the 12th International Conference Information and Knowledge Management, 3-8 November, New Orleans, LA*, pp. 556-9.

McNee, S., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S., Rashid, A., Konstan, J. and Ried, J. (2002),),”On the recommending of citations for research papers”, *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work, New Orleans, LA, November 16-20*, pp. 116-125.

Papavlasopoulos, S., Poulos, M., Korfiatis, N. and Bokos, G. (2010), “A non-linear index to evaluate a journal’s scientific impact”, *Journal of Information Sciences*, Vol. 180 No. 11, pp. 2156-75.

Ritchie, A., Robertson, S. and Teufel, S. (2008a), “Comparing citation contexts for information retrieval”, in *Proceeding of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, October 26-30*, pp. 213-222.

Ritchie, A., Teufel, S. and Robertson, S. (2008b), “Using terms from citations for IR: some first results”, *Proceedings of ECIR, Glasgow, UK, March 30-April 3*, pp. 211-221.

Seglen, P.O. (1997), “Why the impact factor of journals should not be used for evaluating research”, *British Medical Journal*, Vol. 314, pp. 498-502.

Small, H. (1973), “Co-citation in the scientific literature: a new measurement of the relationship between two documents”, *Journal of the American Society of Information Science*, Vol. 24 No. 4, pp. 265-9.

Small, H. (1982), “Citation context analysis”, in Dervin, B. and Voigt, M.J. (Eds), *Progress in Communication Sciences*, Vol. 3, Ablex Publishing, New York, NY, pp. 287-310.

Smith, L.C. (1981), “Citation analysis”, *Journal of Library Trends*, Vol. 30 No. 1, pp. 83-106.

Strohman, T., Croft, W.B. and Jensen, D. (2007), “Recommending citations for academic papers”, In *Proceedings of the 30th Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Amsterdam, July 23-27, 2007*, pp. 705-706.

Szymanski, B.K., de la Rosa, J.L. and Krishnamoorthy, M. (2012), “An internet measure of the value of citations”, *Journal of Information Sciences*, Vol. 181 No. 1, pp. 18-31.

Tang, J. and Zhang, J. (2009), “A discriminative approach to topic-based citation recommendation”, *Proceedings of PAKDD, Bangkok, April 27-30*, pp. 572-9.

Torres, R., McNee, S.M., Abel, M., Konstan, J.A. and Riedl, J. (2004), "Enhancing digital libraries with TechLens", *Proceedings of IEEE/ACM Joint Conference on Digital Libraries, ACM/IEEE JCDL'2004, Tucson, AZ, June 7-11*, pp. 228-36.

Zhang, C., Liu, X., Xu, Y.C. and Wang, Y. (2011), "Quality-structure index: a new metric to measure scientific journal influence", *Journal of the American Society for Information Science and Technology*, Vol. 62 No. 4, pp. 643-53.

Appendix. Time complexity analysis

In this section, the time complexity of the proposed algorithm is discussed.

The first step contains two main phases:

(1) Initial candidate set generation: this is performed by the search component of *Lucene* which according to (Cutting and Pedersen, 1997) its time complexity for finding top-C publications is $O(P.logC)$,where P is the number of postings for each query term. In the worst case, where a query term exists in all the publications, the value of P is equal to N_P, therefore, by ignoring the constant $C = 25$, the time complexity is $O(N_p)$.

(2) Candidate set extension: for each publication of *initCandSet*, the existence of the six relations from that publication to every publication of the background data is checked. The cost of this operation is $O(C\ {}^{*} N_p\ {}^{*}\ cost_{check})$, where $cost_{check}$, the cost of checking existence of the six relations, is $\text{cost}_{\text{check}} = \text{O}(\log(\text{N}_\text{R}) + \log(\text{N}_{\text{auth}}) + \log(\text{N}_\text{P}))$. Due to space limitation, the detail of this computation is omitted.

Therefore, the overall time complexity of the candidate set generation step is: $O(N_p) + \text{O}(C\ {}^{*}N_P\ {}^{*} [log(N_R) + log(N_{auth}) + log(N_P)])$.

Finally, since $N_R \geq N_{auth}$ and $N_R \geq N_P$, the above expression is summarized to $O(N_P\ {}^{*}log(N_R))$.

In the ranking step, *semanticDist(input, P_i)* is computed using the relational and the textual similarity values. This has the cost of $O(C\ {}^{*}N_C)$, and since $N_C \leq N_P$ it can be summarized to $O(N_P)$. Finally, it is required to select M publications from *candSet* with the least distance from the input text. This can be performed in $O(M\ {}^{*}N_C)$. Since $N_C \leq N_P$, and by ignoring the small constant M, this computation has the time complexity of $O(N_P)$. Therefore, the total cost of ranking is $O(N_P)$.

Finally, by combining the time complexity of the two steps of the proposed algorithm, its total time complexity is:

$$O(N_P\ {}^{*}\ log(N_R)) + O(N_P) = O(N_P\ {}^{*}\ log(N_R))$$

About the authors

Fattane Zarrinkalam is currently doing her ME in Software Engineering at Ferdowsi University of Mashhad, Iran. She received her BE in computer engineering in 2009 from Ferdowsi University of Mashhad, Iran. Her research interests include citation analysis, recommender systems, semantic web and linked data. Fattane Zarrinkalam can be contacted at: f_zk84@yahoo.com

Mohsen Kahani is currently an Associate Professor and Chief Information Office (CIO) of Ferdowsi University of Mashhad, Iran. He received his BE in 1990, from the University of Tehran, Iran, his ME in 1994, and his PhD in 1998 both from University of Wollongong, Australia. His research interests include information technology, software engineering, semantic web and linked data.