Knowl Inf Syst (2017) 51:287–310
DOI 10.1007/s10115-016-0973-5

REGULAR PAPER

Building semantic kernels for cross-document knowledge discovery using Wikipedia

Peng Yan[1] · **Wei Jin**[1]

Received: 26 February 2014 / Revised: 2 March 2016 / Accepted: 22 July 2016 /
Published online: 5 August 2016

Abstract Research into text mining has progressed over the past decade. One of the main challenges now is gauging the difficulty of taking advantage of outside knowledge in the discovery process. In this work, to address the limitations of the traditional bag-of- words model and expand the search scope beyond the document collections at hand, we present a new text mining approach incorporating Wikipedia as the background knowledge. Various semantic kernels are built out of the extensive knowledge derived from Wikipedia and applied to the search scenario of detecting potential semantic relationships between topics. We demonstrate the effectiveness of our approach through comparing with competitive baselines, as well as alternative solutions where only part of Wikipedia resources (e.g., the Wiki-article contents or the associated Wiki-categories) is considered.

Keywords Semantic relatedness · Cross-document knowledge discovery · Document representation

1 Introduction

Fast-growing text information offers an excellent opportunity for text mining, i.e., the automatic discovery of knowledge. Mining semantic relationships/associations between concepts from text is important for inferring new knowledge and detecting new trends. More commonly, text documents are represented as a bag-of-words (BOW), and semantic relatedness between words is measured by statistical information gathered from the corpus such as term frequency (TF), inverse document frequency (IDF), and the widely used Cosine similarity weighting scheme, referred to as vector space model (VSM) [9,10,23]. Clearly, this

✉ Peng Yan
pengyancdl@gmail.com

Wei Jin
wei.jin@ndsu.edu

[1] Department of Computer Science, North Dakota State University, 1320 Albrecht Boulevard, Fargo, ND 58102, USA

is a considerable oversimplification of the problem because a lot of the semantics in a document is lost when just replacing its text with a set of terms, such as the order of terms/concepts ("terms" and "concepts" are used interchangeably in this paper) and the frontiers between sentences or paragraphs. While entities could be treated as concepts and represented by index representation, the correlation between entities is lost. Due to the lack of an effective way of capturing semantics in texts, for certain tasks, especially fine-grained information discovery applications, such as mining relationships between concepts, the traditional VSM demonstrates its inherent limitations. In this work, we present a new approach that attempts to address the above problems by utilizing the Wikipedia repository as background knowledge. Through leveraging Wikipedia, the currently largest human-built encyclopedia, we provide a better semantic representation of any text and a more appropriate estimation of semantic relatedness between concepts. Meanwhile, this integration also sufficiently complements the existing information contained in text corpus and facilitates the construction of a more comprehensive representation and retrieval framework.

1.1 Problem definition and motivation

Our previous work [9,10] introduced a special case of text mining focusing on detecting semantic relationships between two concepts across documents, which we refer to as concept chain queries (CCQ). Formally, a CCQ involving concepts A and B (e.g., "George W. Bush" and "Bin Ladin") has the following meaning: find the most plausible relationship between concept A and concept B assuming that one or more instances of both concepts occur in the corpus, but not necessarily in the same document. The proposal of this new interpretation is based on the observation of the inherent limitations of current Web search engines and domain special search tools. A traditional search involving, for example, two person names, will attempt to find documents mentioning both of these individuals. In the event that there are no documents containing both names, either no documents are returned or just documents with one of the names ranked by relevancy. Even if two or more interrelated documents contain both names, the existing search tools cannot integrate information into one relevant and meaningful answer. The goal of this research is to explore automated solutions to sift through these extensive document collections and automatically discover these significant but unapparent links.

Previous attempts for solving CCQ or similar problems [1,3,9–11,18,23,24,26] have not exploited any background or external knowledge, and all the discovered results are limited to the existing information in the given document collection. In this work, we propose a new model to answer CCQ through constructing semantic kernels to embed extensive knowledge from Wikipedia to complement or enhance the existing information. Conceptually, the semantic kernels proposed here define a quantitative representation that provides a better estimation of semantic relatedness between concepts and can effectively alleviate the semantic loss problem of the traditional VSM in representing texts. Under this context, any texts can be represented as a weighted mixture of a predetermined set of natural concepts from Wikipedia, which are provided by humans themselves and can be easily explained. We believe this integration will have impact far beyond the proposed context and benefit a wide range of natural language processing applications in need of large scale world knowledge.

1.2 Research contributions

The contributions of this work can be briefly summarized as follows:

1. A new Wiki-enabled cross-document knowledge discovery model has been proposed and implemented which effectively complements the existing information contained in any document collection at hand.
2. Semantic kernels are introduced to embed extensive knowledge from Wikipedia. Different information resources (e.g., Wikipedia article contents, categories, and anchor texts) can be selected as the foundation of building corresponding kernels based on the user's interest. Although the discussion in this paper focuses on relationship discovery, we also envision applying the proposed semantic kernel methods to other important problems such as classification and clustering.
3. A better document and query representation framework are provided through incorporating a high-dimensional space of natural concepts derived from Wikipedia. Note that the space of terms and relationships considered now is not limited to those present in the document collection. Instead, it can be significantly enriched by incorporating relevant concepts and associations that span all Wikipedia articles, thus more background knowledge can be integrated. Additionally, this solution can also alleviate the semantic loss problem of the traditional text representation using vector space model through incorporating semantic information captured from Wikipedia.

The remainder of this paper is organized as follows: Sect. 2 discusses related work. Section 3 introduces the general approach of incorporating Wikipedia knowledge to answer concept chain queries. In Sect. 4, we discuss the proposed semantic kernels in detail. Experimental results are presented and analyzed in Sect. 5. Section 6 concludes this work and describes future directions.

2 Related work

There have been a number of text mining algorithms for capturing relationships between concepts [9,10,18,20,23]. However, those approaches are built on the traditional bag-of-words (BOW) representation with no or little background knowledge taken into account, which inevitably exist the above-mentioned limited search scope. Bollegala et al. [2] developed an approach for semantic relatedness calculation using returned page counts and text snippets produced by a Web search engine. Salahli [16] proposed another Web oriented method that calculated semantic relatedness between terms using a set of determiners (special words that are supposed to be highly related to terms of interest). However, the performance of these approaches highly relies on the generated output from search engines and has not reached the satisfying level. Suchanek et al. [22] introduced a system called SOFIE for extracting ontological facts from natural language documents. They leveraged the knowledge of an existing ontology to gather and reason about new fact hypotheses and reported a high precision even from unstructured documents. Lehmann et al. [12] developed a tool called RelFinder for exploring connections between objects in a Semantic Web knowledge base. Hahn et al. [6] introduced Faceted Wikipedia Search in concern of the incapability of searching infobox data using Wikipedia's search engine. Hotho et al. [8] exploited the WordNet to improve the BOW text representation, and Martin [13] developed a method for transforming the noun-related portions of WordNet into a lexical ontology to enhance knowledge representation. These WordNet-based techniques have shown their advantages

of improving the traditional BOW-based representation to some degree but suffer from relatively limited coverage and painful maintenance. Hoffart et al. [7] built YAGO2, an extended version of YAGO, from Wikipedia, GeoNames, and WordNet. Gabrilovich and Markovitch [4] applied machine learning techniques to Wikipedia and proposed a new method to enrich document representation from this huge knowledge repository. However, with the process of feature generation so complicated, a considerable computational effort is required. Wang and Domeniconi [25] embedded background knowledge derived from Wikipedia into a semantic kernel to enrich document representation for text classification. It is able to capture the semantic closeness of synonyms and perform word sense disambiguation for polysemous terms. The empirical evaluation demonstrates that their approach successfully achieves improved classification accuracy with respect to the BOW technique. However, their method is based on a thesaurus built from Wikipedia and constructing the thesaurus itself requires a considerable amount of effort. Milne [14] proposed a Wikipedia-based link vector model (WLVM) to improve semantic relatedness computing using only the link structure and titles of articles, while ignoring their textual content. But subsequent experiments done by other researchers have shown that solely relying on the hyperlink structure of Wikipedia and article titles makes this approach fall well behind the explicit semantic analysis (ESA) [5] technique and only outperform some of the measures provided by Strube [21]. ESA proposed by Gabrilovich and Markovitch [5] is a smart method for fine-grained semantic representation of unrestricted natural language texts. It represents the meaning of any text as a weighted mixture of Wikipedia-based concepts (articles), which they built the so-called interpretation vector to capture and measures semantic relatedness according to the cosine distance between the two interpretation vectors built for two text fragments. Their analysis is explicit in the sense that ESA manipulates manifest concepts derived from Wikipedia compared with [3]. But at the same time, they also pointed out the problem of possibly containing noise concepts in the built vector, especially for text fragments containing multi-word phrases (e.g., multi-word names like George W. Bush). Our approach is motivated by Gabrilovich and Markovitch's proposed Explicit Semantic Analysis technique, and we adapt the original technique to better suit our relationship discovery context. When constructing an interpretation vector for topic representation, we have also developed a series of heuristic techniques to filter out the noisy and irrelevant concepts as mentioned in their original paper.

3 Concept chain queries involving Wikipedia

As mentioned in Sect. 1.1, given two topics of interest (topics are also concepts. They are called topics simply because they are concepts specified by the user), concept chain queries (CCQ) is interpreted in our work as finding the best concept chain across multiple documents that may connect these two topics. Here, we add another level of intelligence: If no relationships are identified in the existing document collection, is there a connection between A and B that can be discovered from the Wikipedia knowledge base? The query output takes the form of chains of entities, as in A $\rightarrow$ C1 $\rightarrow$ C2 $\rightarrow, \cdots, \rightarrow$ B, each relating to and connecting to other concepts in the chain that partially answer the user's information need. Figure 1 below shows an example for the relationship query involving "Bin Ladin" and "Omar Abdel Rahman". The identified connection is Bin Ladin $\rightarrow$ Al Qaeda $\rightarrow$ Abdullah Yusuf Azzam $\rightarrow$ Omar Abdel Rahman. In this example, discovered links may tell a story that "Bin Ladin" who inspired the September 11 attacks founded "Al Qaeda"; "Abdullah Yusuf Azzam", an

Paragraph 1:
Osama bin Mohammed bin Awad bin Laden was the founder of ***al-Qaeda***, the Sunni militant Islamist organization that claimed responsibility for the September 11 attacks on the United States, along with numerous other mass-casualty attacks against civilian and military targets.

Paragraph 2:
In 1989, after the Soviets pulled out of Afghanistan, ***Azzam*** and his deputy ***Osama bin Laden*** decided to keep their movement permanent and founded the ***Al Qaeda***.
… …
However, it was reported that ***Bin Laden*** and ***Azzam*** also had a major dispute on where ***Al Qaeda*** should focus their operations
… …
Azzam is thought to had influence on jihadists such as ***al-Qaeda*** with the third stage of his "four-stage process of jihad"

Paragraph 3:
Although ***Abdel-Rahman*** was not convicted of conspiracy in the Sadat assassination, he was expelled from Egypt following his acquittal. He made his way to Afghanistan in the mid-1980s where he contacted his former professor, ***Abdullah Azzam***, co-founder of Maktab al-Khadamat (MAK) along with ***Osama bin Laden***.
… …
Rahman built a strong rapport with ***bin Laden*** during the Soviet war in Afghanistan and following ***Azzam***'s murder in 1989 ***Rahman*** assumed control of the international jihadists arm of MAK/***Al Qaeda***.

Fig. 1 Relationships discovered between "Bin Ladin" and "Omar Abdel Rahman" from Wikipedia articles: "Osama bin Laden", "Abdullah Yusuf Azzam" and "Omar Abdel Rahman"

Islamic scholar and theologian, was also one of the founders of "Al-Qaeda"; Azzam is Abdel Rahman's former professor and he built a strong rapport with bin Laden during the Soviet war in Afghanistan. Notice this chain spans three related Wikipedia articles: "Osama bin Laden", "Abdullah Yusuf Azzam" and "Omar Abdel Rahman" and cannot be discovered solely based on the existing document collection.

4 The proposed technique

The proposed solution includes exploration of both of the existing document collection and the Wikipedia data. In the following, we detail the algorithms used for each component.

4.1 The proposed framework

4.1.1 Concept extraction and ontology mapping

The 9/11 counterterrorism corpus is used as the dataset for this research. This involves processing a large open source document collection pertaining to the 9/11 attack, including the publicly available 9/11 commission report. Concept extraction involves running an information extraction engine, Semantex [19] on the corpus. Semantex tags named entities, common relationships associated with person and organization, as well as providing subject-verb-object (SVO) relationships. We extract as concepts all named entities, as well as any noun or noun phrases participating in SVO relationships. All named entity instances are retained as instances of their respective named entity concept category. Concepts that are

Table 1 Portion of terrorism ontology

Semantic type	Instances
Human action	Attack, killing, covert action
Leader	Vice president, chief, governor
Country	Iraq, Afghanistan, Pakistan, Kuwait
Diplomatic building	Consulate, Pentagon, UAE Embassy
Government	Bush administration, White house, National security council
Person	Deputy national security adviser, chairman, executive director

not names entities undergo filtering and mapping phases. Table 1 illustrates some portion of concept-ontology matching information that is extracted by the system.

4.1.2 Concept profiles

Profiles for the two input topics are built respectively. A profile is essentially a set of concepts that together represent the corresponding topic. Related concepts are extracted from relevant documents which co-occur with the topic in the sentence level. They are further grouped and ranked by their belonging semantic types. This results in a vector of concept vectors, one for each semantic type.

$$\text{Profile}(T_j) = \{\{w_{j,1,1}m_{1,1}, \ldots, \}, \ldots, \{w_{j,2,1}m_{2,1}, \ldots, \}\} \tag{1}$$

where $m_{x,y}$ represents the concept m_y that belongs to the semantic type x, $w_{j,x,y}$ is the computed weight for $m_{x,y}$. To calculate weights, we used a variation of TF*IDF weighting scheme and then normalize the weights:

$$w_{j,x,y} = u_{j,x,y}/\text{highest}(u_{j,x,d}) \tag{2}$$

where $d = 1, 2, \ldots, l$ and there are totally l concepts for semantic type x, and $u_{j,x,y} = n_{j,x,y} * \log\,(N/n_{x,y})$. Here, N is the number of sentences in the collection, $n_{x,y}$ is the number of sentences in which $m_{x,y}$ occurs and $n_{j,x,y}$ is the number of sentences in which topic T_j and concept $m_{x,y}$ co-occur. The weight is then normalized by the highest $u_{j,x,d}$, the highest value observed for the concepts with semantic type x, producing weights that are in [0,1] within each semantic type.

4.1.3 Generating paths between concepts

This stage finds potential conceptual connections in different levels, creates the concept chain and ranks them according to the weight of the corresponding selected concept. The algorithm is composed of the following steps where the user input is two topics of interest designated, A and C. This stage finds potential conceptual connections in different levels, creates the concept chain and ranks them according to the weight of the corresponding selected concept. The algorithm is composed of the following steps where the user input is two topics of interest designated, A and C.

1. Conduct independent searches for A and C. Build the A and C profiles. Call these profiles AP and CP, respectively.

2. Compute a B profile (BP) composed of terms in common between AP and CP. The weight of a concept in BP is the sum of its weights in AP and CP. This is the first level of intermediate potential concepts.
3. Expand the concept chain using the created BP profile together with the topics to build additional levels of intermediate concept lists which (1) connect the topics to each concept in BP profile in the sentence level within each semantic type, and (2) also normalize and rank them.

Output: Levels of potential concepts ranked by their weights within each semantic type. A potential conceptual connection between A and C is a path starting from topic A, proceeding through sequential levels of intermediate concepts until reaching the ending topic C.

4.2 The proposed framework using Wikipedia data

This section details how we perform concept chain queries on Wikipedia. In comparison with the method developed by Yan [28,30], different types of semantic kernels are designed to incorporate different kinds of Wikipedia knowledge into CCQ. The overall procedure for this integration is described in Fig. 2. Given two topics of interest A and C, we first generate multiple levels of semantic profiles connecting A and C using the techniques presented in Sect. 3, and then those profiles are further enriched by the inclusion of relevant Wikipedia concepts. The weights for each intermediate concept identified from the document collection are also re-ranked through incorporating more semantic knowledge derived from Wikipedia.

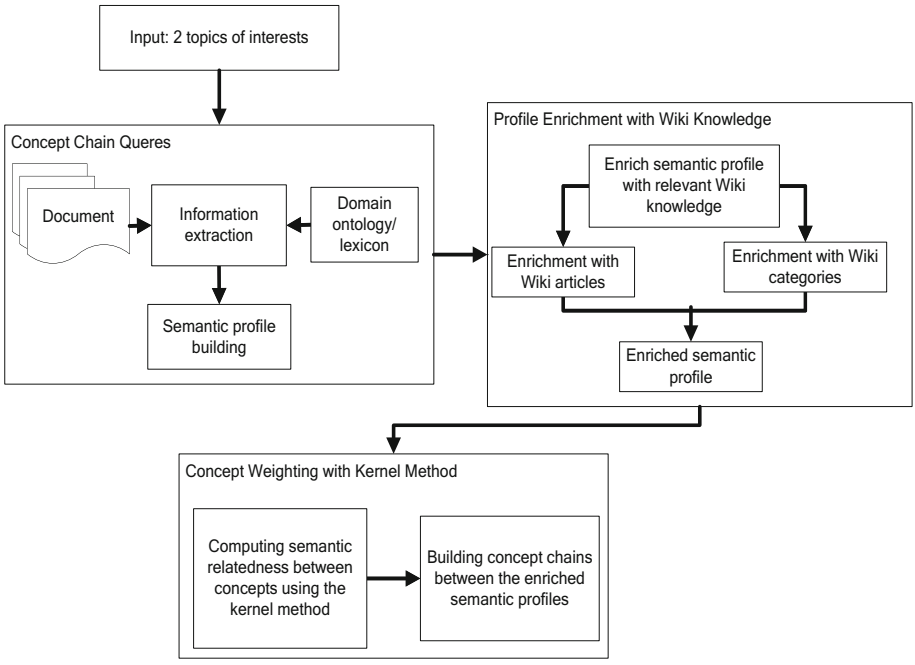


Fig. 2 Answering concept chain queries using kernel methods on Wikipedia

4.2.1 The kernel methods

This section introduces semantic kernels, aiming to alleviate the problem of semantic loss in using the traditional bag-of-words and associated vector space model, where term weights are evaluated only by considering statistical information collected from the document collection such as term frequency (TF), inverse document frequency (IDF). The basic idea of kernel methods is to embed the data in a new suitable feature space (with more information integrated), such that solving the problem in the new space is easier (e.g., linear). To be exact, the new space here stands for the space that incorporates Wikipedia knowledge, and the kernel represents the semantic relationship between two concepts/topics uncovered from this new space.

Given a topic T of interest, the profile of T is now represented as profile$(T) = \langle w_1, w_2, \ldots w_n, c_1, c_2, \ldots c_m \rangle$ where w_i represents concepts from the document collection and c_i represents the new concepts identified from Wikipedia. In the following, we will introduce semantic kernels used to achieve this enrichment.

Given a dictionary containing N number of concepts, a topic T of interest can be represented using a weighted vector as shown below:

$$\phi : T \mapsto \phi(T) = (tf(t_1, T), tf(t_2, T), \ldots, tf(t_N, T)) \in \mathbb{R}^N \tag{3}$$

where $tf(t_i, T)$ represents the co-occurrence frequency of concept t_i and topic T in the sentence level. Based on this representation, we define the semantic kernel capturing the relationship between any two topics T_1 and T_2 as follows:

$$\begin{aligned} k(T_1, T_2) &= \phi(T_1)\phi(T_2)^T \\ &= (tf(t_1, T_1) \quad tf(t_2, T_1) \quad \ldots \quad tf(t_N, T_1)) \times \begin{pmatrix} tf(t_1, T_2) \\ tf(t_2, T_2) \\ \ldots \\ tf(t_N, T_2) \end{pmatrix} \\ &= \sum_{i=1}^{N} tf(t_i, T_1) tf(t_i, T_2) \end{aligned} \tag{4}$$

With the above representation, we are able to measure semantic relatedness between two topics/concepts using a linear kernel [17]. However, two semantically equivalent topics may be mapped to dissimilar feature space if they differ a lot in their co-occurring vocabularies, and thus being classified as irrelevant due to the lack of capability of capturing the underlying semantic meaning of concepts. To address this problem, we propose to embed proper background knowledge into this topic representation. We introduce kernel matrix M that attempts to incorporate outside knowledge from Wikipedia to enriching this representation through $\tilde{\phi}(T) = \phi(T)M$. A semantic kernel between topics T_1 and T_2 is then defined as below:

$$\begin{aligned} k(T_1, T_2) &= \phi(T_1) M M^{\mathrm{T}} \phi(T_2)^{\mathrm{T}} \\ &= \phi(T_1) M (\phi(T_2) M)^{\mathrm{T}} \\ &= \tilde{\phi}(T_1) \tilde{\phi}(T_2)^{\mathrm{T}} \end{aligned} \tag{5}$$

The semantic matrix M can be constructed by creating a sequence of successive embeddings to add additional refinement to the semantics of the representation. One possible solution for M is:

$$M = RP \tag{6}$$

where R is a diagonal matrix representing the concept importance or relevance, and P is a proximity matrix defining semantic relatedness between concepts. Giv-en that $\phi(T)$ is composed of a number of real values indicating the number of oc-currences of each concept, R can be defined as the inverse document frequency and $\phi(T)R$ forms a variation of the TF-IDF weighting. The proximity matrix P can be defined to address the semantic loss of the TF-IDF weighting scheme by relating semantically related concepts together.

$$P_{i,j} = \begin{cases} \text{non-zero} & \text{if} \quad i \neq j, \text{and } t_i \text{ and } t_j \text{ are semantically related} \\ 1 & \text{if} \quad i = j \end{cases} \tag{7}$$

Formally a semantic kernel between two topics T_1 and T_2 is defined below:

$$\begin{aligned} k(T_1, T_2) &= \phi(T_1)RP(RP)^{\mathrm{T}}\phi(T_2)^{\mathrm{T}} \\ &= (\phi(T_1)R)PP^T(\phi(T_2)R)^{\mathrm{T}} \\ &= \tilde{\phi}(T_1)PP^T\tilde{\phi}(T_2)^{\mathrm{T}} \\ &= \tilde{\tilde{\phi}}(T_1)(\tilde{\tilde{\phi}}(T_2))^T \end{aligned} \tag{8}$$

In the following, we will introduce how to build proximity matrices to capture different knowledge contained in Wikipedia.

4.2.2 Proximity matrices for capturing article contents and categories

There are different sources of information from Wikipedia to be used as the basis to construct the proximity matrix P. In this work, we take advantage of Wikipedia article contents and Wikipedia categories. As introduced in Sect. 4.2.1, given a topic T of interest, we now represent it as a weighted vector containing related concepts identified from the existing document collection, along with relevant Wikipedia concepts: $\phi(T) = \langle\langle w_1, w_2, \ldots, w_n\rangle, \langle c_1, c_2, \ldots, c_m\rangle\rangle$ where w_i is a topic-related word identified from the document collection, and c_i is a relevant Wikipedia concept retrieved. The proximity matrix for the topic T is then defined as follows, which is composed of four sub-matrices:

- The word-to-word sub-matrix: the upper left sub-matrix in Fig. 3 is a symmetrical matrix with all of the diagonal entries being 1 and off-diagonal entries representing the similarities between words appearing in the documents.
- The word-to-concept (or concept-to-word) sub-matrix: the upper right and lower left matrices represent the similarities between a word in the document collection and a concept retrieved from Wikipedia. Note that they are actually equivalent as we have similarity(w_i, c_j) = similarity(c_j, w_i).
- The concept-to-concept sub-matrix: the lower right matrix capturing the similarity between two Wikipedia concepts, which is also a symmetrical matrix with diagonal entries being 1 and off-diagonal entries being the similarities between two Wikipedia concepts.

The value of each entry in the 4 sub-matrices is calculated using the adapted Explicit Semantic Analysis technique on Wikipedia article contents [27] or on the Wikipedia categorical information [29], which is basically the cosine similarity between respective interpretation vectors built for each pair of entities in the sub-matrices, and then normalized by the highest value in off-diagonal elements in each sub-matrix.

	w_1	w_2	⋯	w_n	c_1	c_2	⋯	c_m
w_1	1	$s(w_1,w_2)$	⋯	$s(w_1,w_n)$	$s(w_1,c_1)$	$s(w_1,c_2)$	⋯	$s(w_1,c_m)$
w_2	$s(w_2,w_n)$	1	⋯	$s(w_2,w_n)$	$s(w_2,c_1)$	$s(w_2,c_2)$	⋯	$s(w_2,c_m)$
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋱	⋮
w_n	$s(w_n,w_1)$	$s(w_n,w_2)$	⋯	1	$s(w_n,c_1)$	$s(w_n,c_2)$	⋯	$s(w_n,c_m)$
c_1	$s(c_1,w_1)$	$s(c_1,w_2)$	⋯	$s(c_1,w_n)$	1	$s(c_1,c_2)$	⋯	$s(c_1,c_m)$
c_2	$s(c_2,w_1)$	$s(c_2,w_2)$	⋯	$s(c_2,w_n)$	$s(c_2,c_1)$	1	⋯	$s(c_2,c_m)$
⋮	⋮	⋮	⋱		⋮	⋮	⋱	⋮
c_m	$s(c_m,w_1)$	$s(c_m,w_2)$	⋯	$s(c_m,w_n)$	$s(c_m,c_1)$	$s(c_m,c_2)$	⋯	1

Fig. 3 Proximity matrix with enriched features

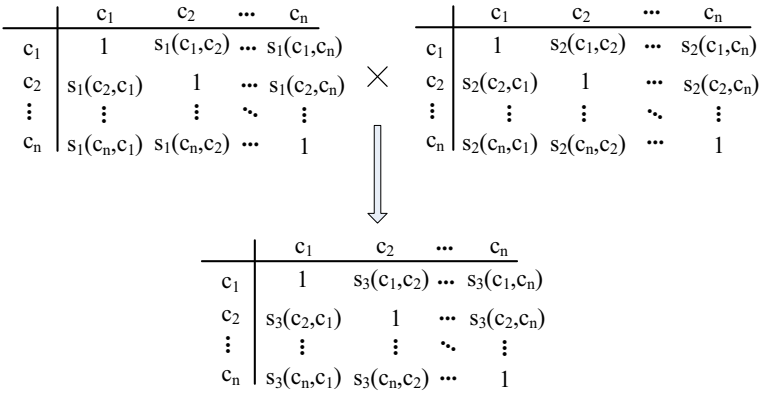


Fig. 4 Hybrid proximity matrix

4.2.3 The hybrid proximity matrix

Suppose the proximity matrix built using the content of Wikipedia articles is P_{content}, and the proximity matrix built using the categorical information derived from Wikipedia is P_{category}, we further define a hybrid proximity matrix as shown in Fig. 4, which attempts to integrate the information from both Wikipedia articles and categories through multiplying the two proximity matrices $P_H = P_{\text{content}} P_{\text{category}}$. Here, we do not distinguish between the words in the document collection and the concepts from Wikipedia in the hybrid proximity matrix P_H.

$$P_{i,j} = \begin{cases} 1 & \text{if} \quad i = j \\ \text{Sim}_H(c_i, c_j)/\text{Sim\_Max}_H & \text{if} \quad i \neq j \end{cases} \tag{9}$$

where $\text{Sim}_H(c_i, c_j) = \sum_{i,j=1}^{n} \text{Sim}_{\text{content}}(c_i, c_j) \cdot \text{Sim}_{\text{category}}(c_j, c_i)$ is the combined similarity between c_i and c_j from P_{content} and P_{category}, and Sim_Max_H is the maximum value in P in off-diagonal elements.

Table 2 Concept vector for the topic "Abdel Rahman"

	Abdullah_Azzam	Bin_Ladin	New_York_City_ Landmark_Bomb_Plot	Maktab_al-Khidamat
Abdel Rahman	0.22	0.12	0.0	0.0

4.2.4 Kernel methods applied to topic representation enrichment

We now demonstrate how the kernel methods can be used to enrich the topic representation with Wikipedia knowledge.

4.2.4.1 Topic representation enrichment with the content-based proximity matrix Suppose a topic T is represented by a weighted vector of words from the document collection and concepts from Wikipedia:

$$\phi(T) = \langle\langle \text{tfidf}(w_1), \text{tfidf}(w_2), \ldots, \text{tfidf}(w_n)\rangle, \langle \text{tfidf}(c_1), \text{tfidf}(c_2), \ldots, \text{tfidf}(c_m)\rangle\rangle \quad (10)$$

where $\text{tfidf}(w_i)$ is the TF-IDF value of the word w_i in the document collection and $\text{tfidf}(c_i)$ is the TF-IDF value of Wikipedia concept c_i over Wikipedia data. Before concept enrichment is conducted, the TF-IDF values of all relevant Wikipedia concepts, if they do not appear in the document collection, are initialized to zero as shown in Table 2. For example, the initial concept vector for the input topic "*Abdel Rahman*" is illustrated below. The first entry "*Abdullah_Azzam*" is a highly influential Palestinian Sunni Islamic scholar and theologian. He is also known as a teacher and mentor of Osama bin Laden who was the founder of al-Qaeda and responsible for the September 11 attacks. The third entry "*New_York_City_Landmark_Bomb_Plot*" is a planned follow-up to the February 1993 World Trade Center bombing designed to inflict mass casualties on American soil by attacking well known landmark targets throughout New York City in the USA. "*Abdel Rahman*" is one of the conspirators of it. The last entry "*Maktab_alKhidamat*" was the forerunner to al-Qaeda which was founded in 1984 by Abdullah Azzam and Osama bin Laden to raise funds and recruit foreign mujahidin for the war against the Soviets in Afghanistan. The last two entries do not appear in the document collection, so their corresponding TF-IDF values are set to zero.

We now define the proximity matrix using Wikipedia knowledge to update the above topic representation vector using the techniques introduced in Sect. 4.2.2. After obtaining the proximity matrix for the given topic "*Abdel Rahman*", which is shown in Table 3, we multiply its initial concept vector by this proximity matrix and then get the new concept vector with enriched concepts derived from Wikipedia along with their respective relevance scores with regard to the topic, as shown in Table 4.

4.2.4.2 Topic representation enrichment with the category-based proximity matrix Similarly, Wikipedia categories can also be utilized to enrich the topic representation. For example, given the topic: "Blind Sheikh", the corresponding document-level concept vector is constructed as below in Table 5.

The first two entries "Khallad" and "Salameh" have nonzero TF-IDF values as they occur in the document collections, whereas the last two entries do not. The proximity matrix using the Wikipedia categories is shown in Table 6. We then multiply the initial concept vector for

Table 3 Article content-based proximity matrix built for the topic "Abdel Rahman"

	Abdullah_Azzam	Bin_Ladin	New_York_City_ Landmark_Bomb_Plot	Maktab_al-Khidamat
Abdullah_Azzam	1	1	0.0006	0.0024
Bin_Ladin	1	1	0.0009	0.0021
New_York_City_ Landmark_Bomb_Plot	0.0006	0.0009	1	0.0
Maktab_al-Khidamat	0.0024	0.0021	0.0	1

Table 4 Enriched concept vector for the topic "Abdel Rahman"

	Abdullah_Azzam	Bin_Ladin	New_York_City_ Landmark_Bomb_Plot	Maktab_al-Khidamat
Abdel Rahman	0.34	0.34	0.0002	0.0008

Table 5 The concept vector for the topic "Blind Sheikh" built from the document collection

	Khallad	Salameh	Islamic_Terrorism	Jihadist_ Organizations
Blind Sheikh	0.20	0.16	0.0	0.0

Table 6 Category-based proximity matrix for the topic "Blind Sheikh"

	Khallad	Salameh	Islamic_Terrorism	Jihadist_ Organizations
Khallad	1	0.73	0.82	0.39
Salameh	0.73	1	1	0.53
Islamic_Terrorism	0.82	1	1	0.81
Jihadist_Organizations	0.39	0.53	0.81	1

Table 7 Enriched concept vector for the topic "Blind Sheikh"

	Khallad	Salameh	Islamic_Terrorism	Jihadist_ Organizations
Blind Sheikh	0.32	0.31	0.32	0.16

"Blind Sheikh" by the built proximity matrix, and obtain a new concept vector incorporating new concepts from Wikipedia that are categorically related to the topic of interest as shown in Table 7.

4.2.4.3 Topic representation enrichment with the hybrid proximity matrix The hybrid proximity matrix introduced in Sect. 4.2.3 can also be employed for this purpose, which has both the content of Wikipedia articles and categorical information embedded. For example,

Table 8 Concept vector for the topic "Ayman Zawahiri"

	Bin_Ladin	Essam_al-Qamari	Ali_Sayyid_Muhamed_Mustafa_al-Bakri	Abdullah_Yusuf_Azzam	Afghan_Civil_War
Ayman Zawahiri	0.71	0.0	0.0	0.0	0.0

Table 9 Hybrid proximity matrix for the topic "Ayman Zawahiri"

	Bin_Ladin	Essam_al-Qamari	Ali_Sayyid_Muhamed_Mustafa_al-Bakri	Abdullah_Yusuf_Azzam	Afghan_Civil_War
Bin_Ladin	1	0.82	0.97	1	0.53
Essam_al-Qamari	0.82	1	0.35	0.36	0.15
Ali_Sayyid_Muhamed_Mustafa_al-Bakri	0.97	0.35	1	0.66	0.38
Abdullah_Yusuf_Azzam	1	0.36	0.66	1	0.50
Afghan_Civil_War	0.53	0.15	0.38	0.50	1

Table 10 Enriched concept vector for the topic "Ayman Zawahiri"

	Bin_Ladin	Essam_al-Qamari	Ali_Sayyid_Muhamed_Mustafa_al-Bakri	Abdullah_Yusuf_Azzam	Afghan_Civil_War
Ayman Zawahiri	1.66	1.37	1.62	1.66	0.87

suppose our topic of interest is "*Ayman Zawahiri*", and its original document-level concept vector is as follows in Table 8.

The first entry "*Bin_Ladin*" is a concept appearing in the document collection and has a nonzero TF-IDF value of 0.71. The second and third entries are related Wikipedia articles, and the last two entries are relevant Wikipedia categories. We then compute the hybrid proximity matrix based on the technique described in 4.2.2, which is shown below in Table 9.

Similarly, through multiplying the initial concept vector by the hybrid proximity matrix, we obtain a new concept vector with four enriched features as illustrated in Table 10. Note that using the hybrid proximity matrix, the newly identified Wikipedia concept, "*Abdullah_Yusuf_Azzam*", who has deep influence on "*Bin Ladin*" and is considered as the Father of Global Jihad, is weighted as the top related concept to "*Ayman Zawahiri*" who is in the list of FBI most wanted terrorists.

4.3 Expediting construction of proximity matrix using map-reduce

As the proximity matrix is built from millions of Wikipedia articles and categories, we propose to expedite this process using a distributed Map-Reduce framework. In particular, we use an efficient single-pass in-memory indexing technique on Wikipedia resources to speed up

```
/* Concept vector C is: 1 x n, proximity matrix is: n x n. Elements in C are stored in the format
("C", 0, j, C_0j) where 0 is row #, j is column #, and C_0j is the value of the element. Similarly,
elements in M are stored in the format ("M", j, k, M_jk). The map function emits a line in the file
to the reduce function. */
function map (Key : lineNum, value : lineContent)
   // value is either ("C", 0, j, C_0j) or ("M", j, k, M_jk)
   if value[0] == "C":
      j = value[2]
      C_ij = value[3]
      for k = 0 to n - 1:
         emit((0, k), (C, j, C_0j))
   else:
      j = value[1]
      k = value[2]
      M_jk = value[3]
      emit((0, k), (M, j, M_jk))

reduce(key, valueList):
   // key is (i, k), valueList is a list of ("C", j, C_0j) and ("M", j, M_jk)
   for each x in valueList
        if (x[0] == "C")
           put <x[1], x[2]> into hash_C
        else
           put <x[1], x[2]> into hash_M
   result = 0
   for j = 0 to n - 1:
      result += hash_C[j] * hash_M[j]
   emit(key, result)
```

Fig. 5 Map-reduce algorithm for computing the product of a concept vector and its proximity matrix

the construction of concept vectors and then apply a map-reduce solution to improve the computation of vector-matrix product in a parallel manner. Appropriate "Map" and "reduce" functions are designed and illustrated in Fig. 5. Basically, the concept vector and proximity matrix are stored in a file as follows: suppose $e_{i,j}$ is an element of the concept vector C (1 by n) or the proximity matrix M (n by n), then $e_{i,j}$ is stored as one line in the file in the format of "C/M, i, j, $e_{i,j}$". The detailed algorithm is shown below.

5 Empirical evaluation

A challenging task in the evaluation was constructing an evaluation data set, since there are no standard data sets available for quantitatively evaluating concept chains. The objectives of this section are to evaluate how the various semantic kernels proposed perform in capturing the semantic relationships between concepts.

5.1 Processing Wikipedia dumps

Wikipedia offers free copies of the entire content in the form of XML files. It is an ever-updating knowledge base and releases the latest dumps to interested users regularly. The version used in this work was released on April 05, 2011, which was separated into 15 compressed XML files and altogether occupied 29.5 GB after decompression. An open source

tool MWDumper [15] was used to import the XML dumps into our MediaWiki database, and after the parsing process, we identified 5,553,542 articles and 794,778 categories.

5.2 Evaluation data

An open source document collection pertaining to the 9/11 attack, including the publicly available 9/11 commission report was used in our evaluation. The report consists of Executive Summary, Preface, 13 chapters, Appendix and Notes. Each of them was considered as a separate document resulting in 337 documents. The whole collection was processed using Semantex [19], and concepts were extracted and mapped to the counterterrorism domain ontology. A variety of query pairs were selected by the assessors covering various scenarios (e.g., ranging from popular entities to rare entities) and used as our evaluation data.

We chose pairs of topics covering various scenarios in the counterterrorism corpus and the topics were mostly named entities. For each topic pair, the relevant paragraphs for either topic, respectively, were manually inspected: We selected those where there was a logical connection between the two topics. This process generated 34 query pairs in 9/11 corpus. After achieving agreement among all annotators, we then selected chains of lengths ranging from 1 to 4 in terms of the number of associations. This process resulted in 37 chains in 9/11 corpus which were used as truth chains for later experiments.

5.3 Experimental results

We have implemented a baseline model based on Srinivasan's closed text mining algorithm [20]. We run the aforementioned 34 query pairs using 4 methods to generate concept chains: the VSM-based method, the Wiki-article content-based kernel, the Wiki-category-based kernel and the hybrid kernel. For a more detailed efficiency analysis, the main computation time was spent on building the proximity matrix. The performance of constructing the proximity matrices for each query pair is further tested under a test cluster containing 3 nodes on Ubuntu 14.04 LTS 64-bit, and 1 master and 3 slaves (the master also takes a role as a slave in some cases) are adopted. Each node has 4 Intel(R) cores (each @ 2.53 GHz), and

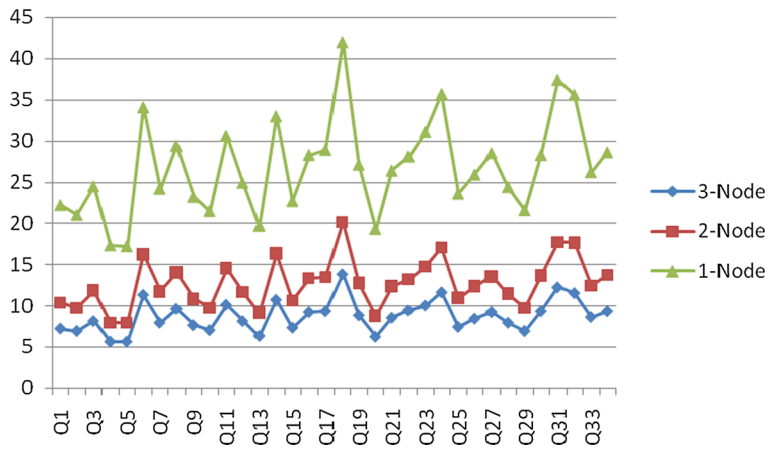


Fig. 6 Runtime analysis for building the proximity matrices for query pairs

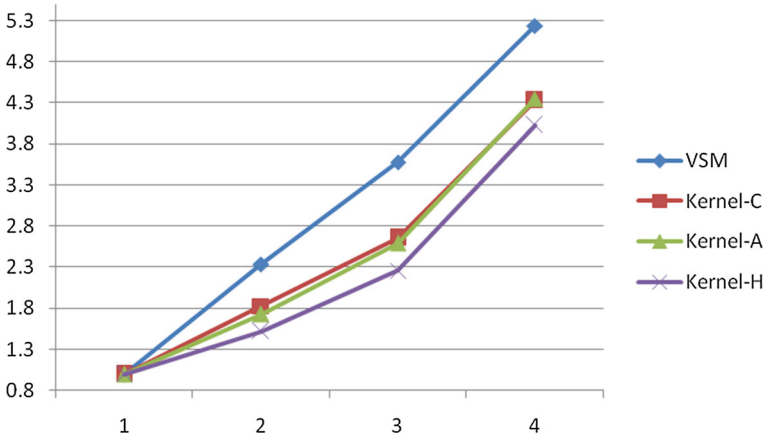


Fig. 7 Average rank of detected concept chains

Table 11 Effect of using the article-content-based kernel

	Baseline/article-content-based kernel					
	S_5	S_{10}	S_{15}	S_{20}	S_{30}	S_{40}
L_1						
Baseline	0.8844	0.8689	0.8700	0.8668	0.8597	0.8546
W5	0.9048	0.8861	0.8842	0.8808	0.8798	0.8798
W10	0.9074	0.8889	0.8870	0.8836	0.8826	0.8826
W15	0.9086	0.8902	0.8884	0.8850	0.8840	0.8840
W20	0.9067	0.8886	0.8868	0.8834	0.8825	0.8825
L_2						
Baseline	0.9174	0.9081	0.8998	0.8959	0.8917	0.8888
W5	0.9155	0.9106	0.9007	0.8974	0.8945	0.8928
W10	0.9226	0.9139	0.9075	0.9041	0.9011	0.8995
W15	0.9272	0.9184	0.9120	0.9087	0.9057	0.9040
W20	0.9306	0.9217	0.9154	0.9120	0.9090	0.9074
L_3						
Baseline	0.9180	0.9109	0.9003	0.8964	0.8922	0.8893
W5	0.9157	0.9109	0.9009	0.8976	0.8946	0.8930
W10	0.9228	0.9142	0.9077	0.9044	0.9014	0.8997
W15	0.9275	0.9187	0.9123	0.9090	0.9059	0.9043
W20	0.9309	0.9220	0.9157	0.9124	0.9093	0.9077
L_4						
Baseline	0.8444	0.8265	0.8109	0.8027	0.7919	0.7865
W5	0.8456	0.8271	0.8115	0.8023	0.7932	0.7901
W10	0.8473	0.8279	0.8119	0.8031	0.7941	0.7916
W15	0.8479	0.8290	0.8127	0.8041	0.7967	0.7933
W20	0.8562	0.8295	0.8135	0.8055	0.7985	0.7965

Table 12 Effect of using the category-based kernel

	Baseline/category-based kernel					
	S_5	S_{10}	S_{15}	S_{20}	S_{30}	S_{40}
L_1						
Baseline	0.8844	0.8689	0.8700	0.8668	0.8597	0.8546
W5	0.9202	0.9030	0.9011	0.8979	0.8969	0.8969
W10	0.9347	0.9195	0.9175	0.9148	0.9138	0.9138
W15	0.9437	0.9299	0.9280	0.9255	0.9246	0.9246
W20	0.9497	0.9370	0.9352	0.9329	0.9320	0.9320
L_2						
Baseline	0.9174	0.9081	0.8998	0.8959	0.8917	0.8888
W5	0.9185	0.9103	0.9042	0.9012	0.8987	0.8974
W10	0.9252	0.9168	0.9106	0.9076	0.9050	0.9037
W15	0.9297	0.9211	0.9150	0.9120	0.9093	0.9080
W20	0.9329	0.9282	0.9183	0.9152	0.9126	0.9113
L_3						
Baseline	0.9180	0.9109	0.9003	0.8964	0.8922	0.8893
W5	0.9185	0.9102	0.9037	0.9005	0.8976	0.8960
W10	0.9253	0.9167	0.9103	0.9071	0.9041	0.9025
W15	0.9298	0.9211	0.9148	0.9116	0.9086	0.9070
W20	0.9331	0.9283	0.9181	0.9149	0.9120	0.9104
L_4						
Baseline	0.8444	0.8265	0.8109	0.8027	0.7919	0.7865
W5	0.8469	0.8268	0.8111	0.8023	0.7911	0.7871
W10	0.8498	0.8289	0.8127	0.8034	0.7959	0.7891
W15	0.8532	0.8301	0.8141	0.8056	0.7988	0.7937
W20	0.8583	0.8323	0.8153	0.8084	0.8015	0.7995

the cluster has a total memory of 48G and 12 cores. Figure 6 shows the runtime comparison of building the proximity matrices for each of the 34 query pairs in different execution environments (i.e., environments with 1 node, 2 nodes, and 3 nodes, respectively). The X-axis represents the runtime measured by minutes, and Y-axis corresponds to the query pairs. From Fig. 6, it is obvious to see that the main factor affecting the runtime is the number of CPU cores and the computation time is significantly reduced with the increase of CPU cores. The average runtime for processing all query pairs in the 3-node environment is about 8.7 minutes.

The evaluation then looks at (1) whether the target chains were found as the top choice by different models, and (2) if not the top choice, the ranks of the found truth chains. The query evaluation resulted in the discovery of 24 truth chains by each method. Then, we divide the 24 detected chains into 4 groups according to their lengths, and compute the average rank for each of the 4 groups as follows:

$$\text{rank}(\text{group}_t) = \frac{1}{s}\sum_{q=1}^{s}\text{rank}(\text{chain}_q)$$

Table 13 Effect of using the hybrid kernel

	Baseline/hybrid kernel					
	S_5	S_{10}	S_{15}	S_{20}	S_{30}	S_{40}
L_1						
Baseline	0.8844	0.8689	0.8700	0.8668	0.8597	0.8546
W5	0.9267	0.9104	0.9084	0.9054	0.9044	0.9044
W10	0.9402	0.9259	0.9239	0.9213	0.9204	0.9204
W15	0.9482	0.9353	0.9334	0.9311	0.9302	0.9302
W20	0.9522	0.9403	0.9385	0.9364	0.9355	0.9355
L_2						
Baseline	0.9174	0.9081	0.8998	0.8959	0.8917	0.8888
W5	0.9168	0.9121	0.9057	0.9033	0.8994	0.8980
W10	0.9263	0.9173	0.9144	0.9090	0.9058	0.9041
W15	0.9332	0.9285	0.9194	0.9162	0.9101	0.9092
W20	0.9334	0.9295	0.9190	0.9173	0.9147	0.9139
L_3						
Baseline	0.9180	0.9109	0.9003	0.8964	0.8922	0.8893
W5	0.9199	0.9168	0.9055	0.9013	0.8988	0.8972
W10	0.9273	0.9188	0.9130	0.9097	0.9061	0.9037
W15	0.9298	0.9233	0.9167	0.9144	0.9093	0.9085
W20	0.9354	0.9297	0.9187	0.9162	0.9134	0.9122
L_4						
Baseline	0.8444	0.8265	0.8109	0.8027	0.7919	0.7865
W5	0.8435	0.8261	0.8116	0.8021	0.7938	0.7893
W10	0.8479	0.8285	0.8129	0.8038	0.7973	0.7927
W15	0.8525	0.8328	0.8143	0.8069	0.8022	0.7988
W20	0.8592	0.8374	0.8162	0.8093	0.8050	0.8017

where $t = \{1, 2, 3, 4\}$, s is the number of chains in group$_t$, rank(chain$_q$) $= \frac{1}{n}\sum_{i=1}^{n}$ rank(c_i), where n is the number of concepts in chain$_q$, and c_i is one of the concepts constituting chain$_q$. Figure 7 illustrates the improvement of the average rank of the concept chains by the kernel method compared with the VSM-based method. The X-axis indicates the 4 groups, and the Y-axis indicates the average rank of truth chains detected in each group. VSM indicates the VSM-based method, and Kernel-C, Kernel-A and Kernel-H indicate the models using Wiki-article-based kernel, the Wiki-category-based kernel and the hybrid kernel, respectively.

Tables 11, 12, 13 make a comparison between the search results of our baseline where the corpus-level tf-idf-based statistical information is used to generate chains without the involvement of Wikipedia and various Wiki-enabled models proposed in this work. The table entries can be read as follows: S_N / W_N means the top N concepts are kept in the search results (S_N for the concepts appearing in the documents and S_N for the concepts derived from Wikipedia). L_N indicates the resulting chains of length N. The entries in the three tables stand for the precision values defined as follows. For example, the entries in the row "Baseline" in Table 11 represent the precision ratios when the top 5, 10, 15, 20, 30, 40 concepts are kept, respectively, in the ranked list of all detected concepts.

Table 14 Instances of key semantic relationships

Chain length	Query pair	Resulting chain
L2 (Length 2)	abdel_rahman : : blind_sheikh	abdel_rahman → new_york_city_landmark_bomb_plot → blind_sheikh
	george bush : : bin_ladin	george_bush → richard_a._clarke → bin_ladin
	alexis : : lloyd_salvetti	alexis → janice_kephart_roberts → lloyd_salvetti
	adel : : ffi	adel → afghanistan → ffi
	marty_miller : : oakley	marty_miller → unocal → oakley
	gore : : stephen_hadley	gore → clarke → stephen_hadley
	donovan : : wall_street	donovan → intelligence_group → wall_street
L3 (Length 3)	atta : : dekkers	atta → mohammed_atta_al_sayed → marwan_al-shehhi → dekkers
	amal : : sudanese	amal → bahrain → cia → sudanese
	karachi : : usama_asmurai	karachi → june_14_terrorist_attack_outside_us_consulate_in_karachi → may_8_bus_attack_in_karachi → usama_asmurai
	binalshibh : : pistole	binalshibh → fbi → minneapolis → pistole
	martha_stewart : : saudi_arabia	martha_stewart → al_jawf,_saudi_arabia → khaled_of_saudi_arabia → saudi_arabia
L4 (Length 4)	kenya : : mohamed	kenya → mihdhar_hazmi → afghanistan → shanksville → mohamed
	gore : : stephen_hadley	gore → suicide_hijackings → white house → national_security_council → stephen_hadley
	crawford : : khalilzad	crawford →bill_clinton → afghan → deuty_secretary_state_richard_armitage → khalilzad

$$P = \frac{\text{Concepts found and relevant}}{\text{Total concepts found}}$$

where N is the number of query pairs used in our experiments. We chose various pairs of topics covering various scenarios in the counterterrorism corpus, and the topics were mostly named entities. For each topic pair, the relevant paragraphs for either topic, respectively, were then manually inspected: We selected those where there was a logical connection between the two topics. This process generated 34 query pairs in 9/11 corpus.

Specifically, Table 11 shows the improvement achieved by integrating the Wiki-article content-based kernel over the baseline. Table 12 presents the result when the relevant Wiki-categories are used to build the semantic kernel. Table 13 demonstrates the overall benefit when utilizing the hybrid semantic kernel where both article content and categories are incorporated. It is easy to observe that the search performance is improved with the integration of Wikipedia knowledge, and the best performance is observed when both the Wiki-article content and categories are involved.

Table 14 shows newly discovered semantic relationships where linking concepts can only be acquired by integrating information from multiple documents or fromWikipedia knowl-

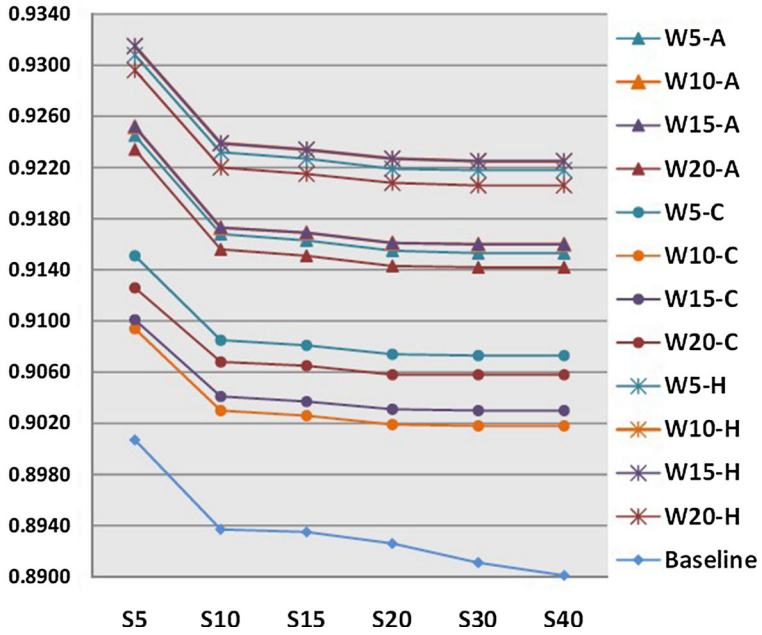


Fig. 8 Adapted MAP for chains of length 1

edge (i.e., not contained in the existing document collection). For instance, for the query pair: "Atta : : dekkers", two intermediate important persons connecting them were identified: "Mohammed_Atta_al_Sayed" was an Egyptian hijacker and one of the ringleaders of the September 11 attacks and "Marwan_al-Shehhi" was the hijacker-pilot of United Airlines Flight 175, crashing the plane into the South Tower of the World Trade Center as part of the September 11 attacks.

We also used the adapted MAP measure as shown in the below formula:

$$\text{AdaptedMAP}(Q) = (\Sigma P(k_{s,w}))/|Q|$$

where Q is a set containing all query pairs, $s = \{5, 10, 15, 20, 30, 40\}$ and $w = \{5, 10, 15, 20\}$ indicate the top N concepts were kept in the search results (s for the concepts appearing in the documents and w for the concepts derived from Wikipedia). $P(k_{s,w})$ is the precision where the top s concepts from documents and the top w concepts from Wikipedia were kept.

Figures 8, 9, 10, 11 interpret the search results using the MAP measure. *SN* where $N = \{5, 10, 15, 20, 30, 40\}$ and *WN* where $N = \{5, 10, 15, 20\}$ have the same meaning as in Tables 11, 12, 13. The baseline is the VSM-based model. For *WN-X* where $X = \{A, C, H\}$, A indicates the article content-based kernel, C indicates the category-based kernel and H indicates the hybrid kernel. We observe that the kernel-based approach consistently achieves better performance for different lengths than the baseline solution, and the hybrid kernel achieves the highest MAP values for chains of different lengths.

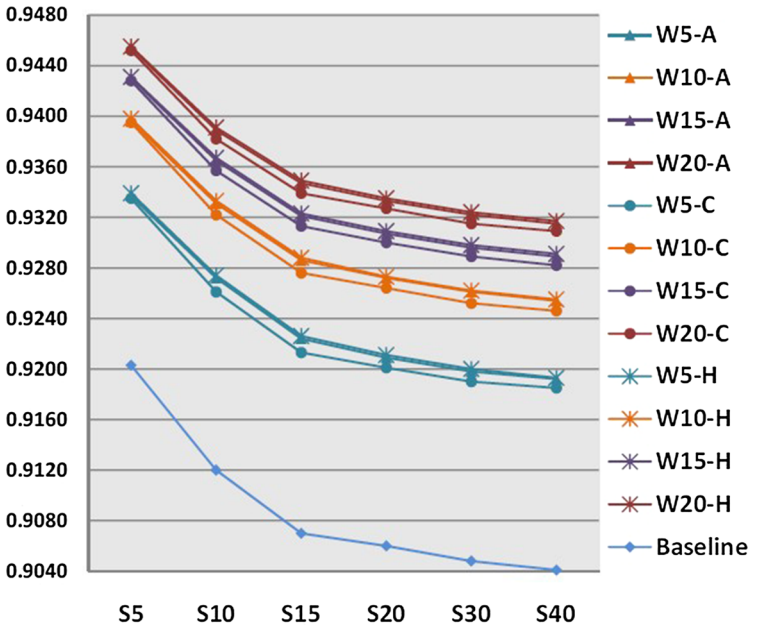


Fig. 9 Adapted MAP for chains of length 2

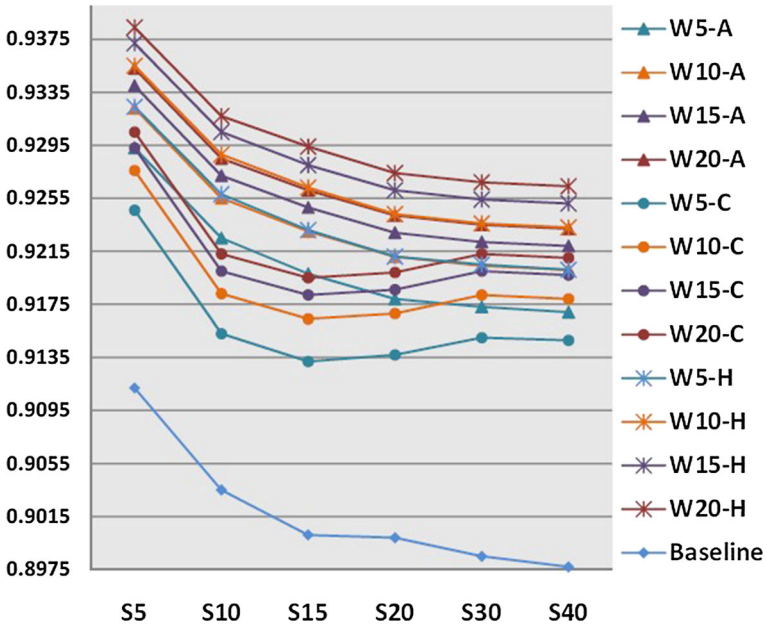


Fig. 10 Adapted MAP for chains of length 3

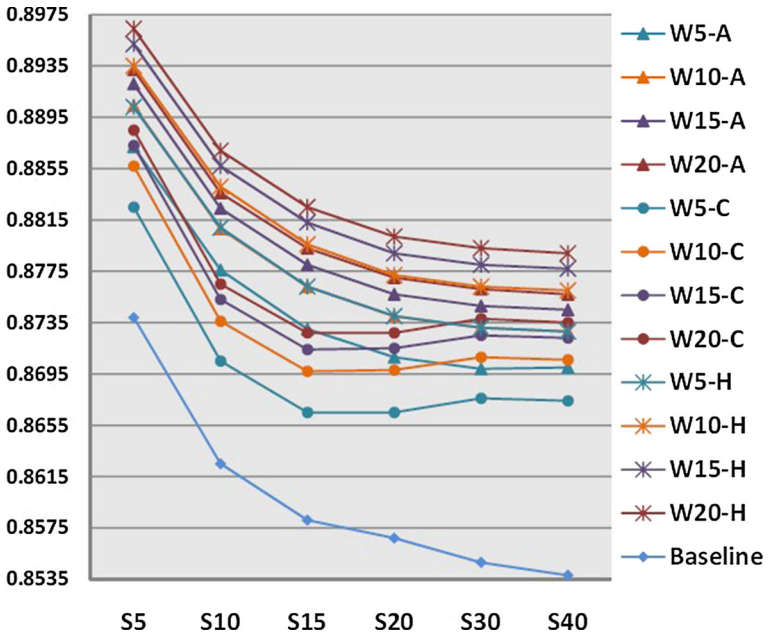


Fig. 11 Adapted MAP for chains of length 4

6 Conclusion and future work

In this work, we propose a new solution for cross-document knowledge discovery through building semantic kernels that integrate different background knowledge from Wikipedia. The kernel methods focus on providing a best estimation of semantic relatedness between concepts in a new space that embeds different information resources from Wikipedia. Over 5,000,000 Wikipedia articles and 700,000 Wikipedia categories are explored, which expands the relationship search scope beyond those appearing in the document collection at hand literally. Specifically, the explicit semantic analysis (ESA) technique is adapted to help measure concept closeness and a Wiki-content-based semantic kernel and a Wiki-category-based semantic kernel are built to capture semantic relatedness between concepts in terms of relevant Wikipedia article contents and associated categories, respectively. Moreover, a hybrid semantic kernel integrating both Wiki-articles and categories is also designed and evaluated. Empirical evaluation demonstrates the proposed approaches achieve much broader and well-rounded coverage of significant relationships between concepts.

In addition to the relationship discovery scenario, the proposed semantic kernels can also be easily applied to various important data mining tasks such as classification and clustering. Other than articles and categories in Wikipedia, we will also be exploring and evaluating the usage of other valuable resources in Wikipedia, e.g., anchor texts and infoboxes, in contributing to this task in our future work.

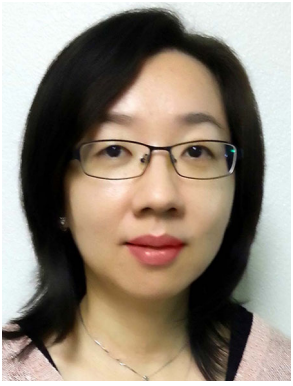
Acknowledgements This research work is supported in part by the NSF Grant (IIS-1452898) and NSF/North Dokota EPSCoR IIP Seed Grant (EPS-0814442).

References

1. Budanitsky A, Hirst G (2006) Evaluating wordnet-based measures of lexical semantic relatedness. Comput Linguist 32(1):13–47
2. Bollegala D, Matsuo Y, Ishizuka M (2007) Measuring semantic similarity between words using web search engines. In: Proceedings of the 16th international conference on World Wide Web, pp 757–766
3. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41(6):391
4. Gabrilovich E, Markovitch S (2006) Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge. AAAI 6:1301–1306
5. Gabrilovich E, Markovitch S (2007) Computing semantic relatedness using Wikipedia-based explicit semantic analysis. IJCAI 7:1606–1611
6. Hahn R, Bizer C, Sahnwaldt C, Herta C, Robinson S, Bürgle M, Düwiger H, Scheel U (2010) Faceted Wikipedia search. Bus Inf Syst 47:1–11
7. Hoffart J, Suchanek FM, Berberich K, Weikum G (2013) YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia. Artif Intelli 194:28–61
8. Hotho A, Staab S, Stumme G (2003) Wordnet improves text document clustering. In: Proceedings of the SIGIR 2003 semantic web workshop
9. Jin W, Srihari RK (2006) Knowledge discovery across documents through concept chain queries. In: Proceeding of the sixth IEEE international conference on data mining workshops, pp 448–452
10. Jin W, Srihari RK, Ho HH, Wu X (2007) Improving knowledge discovery in document collections through combining text retrieval and link analysis techniques. In: Proceeding of the seventh IEEE international conference on data mining, pp 193–202
11. Jin W, Srihari RK (2007) Graph-based text representation and knowledge discovery. In: Proceedings of the 2007 ACM symposium on applied computing, pp 807–811
12. Lehmann J, Schüppel J, Auer S (2007) Discovering unknown connections-the DBpedia relationship finder. CSSW 113:99–110
13. Martin P (2003) Correction and extension of WordNet 1.7. In: Conceptual structures for knowledge creation and communication, pp 160–173
14. Milne D (2007) Computing semantic relatedness using Wikipedia link structure. In: Proceedings of the New Zealand computer science research student conference
15. MWDumper. Software. http://www.mediawiki.org/wiki/Manual:MWDumper
16. Salahli MA (2009) An approach for measuring semantic relatedness between words via related terms. Math Comput Appl 14(1):55
17. Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge
18. Srihari RK, Lamkhede S, Bhasin A (2005) Unapparent information revelation: a concept chain graph approach. In: Proceedings of the 14th ACM international conference on information and knowledge management, pp 329–330
19. Srihari RK, Li W, Niu C, Cornell T (2003) Infoxtract: a customizable intermediate level information extraction engine. In: Proceedings of the HLT-NAACL 2003 workshop on software engineering and architecture of language technology systems, pp 51–58
20. Srinivasan P (2004) Text mining: generating hypotheses from MEDLINE. J Am Soc Inf Sci Technol 55(5):396–413
21. Strube M, Ponzetto SP (2006) WikiRelate! Computinga semantic relatedness using Wikipedia. AAAI 6:1419–1424
22. Suchanek FM, Sozio M, Weikum G (2009) SOFIE: a self-organizing framework for information extraction. In: Proceedings of the 18th international conference on World wide web, pp 631–640
23. Swanson DR, Smalheiser NR (1999) Implicit text linkages between Medline records: using Arrowsmith as an aid to scientific discovery. Libr Trends 48(1):48–59
24. Swanson DR (1991) Complementary structures in disjoint science literatures. In: Proceedings of the 14th annual international ACM SIGIR conference on research and development in information retrieval, pp 280–289
25. Wang P, Domeniconi C (2008) Building semantic kernels for text classification using Wikipedia. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, pp 713–721
26. Wong SM, Ziarko W, Wong PC (1985) Generalized vector spaces model in information retrieval. In: Proceedings of the 8th annual international ACM SIGIR conference on research and development in information retrieval, pp 18–25

27. Yan P, Jin W (2012) Improving cross-document knowledge discovery using explicit semantic analysis. In: Proceedings of the 14th international conference on data warehousing and knowledge discovery, pp 378–389
28. Yan P, Jin W (2013) A new approach for improving cross-document knowledge discovery using Wikipedia. In: Proceedings of the 18th international conference on application of natural language to information systems, pp 291–296
29. Yan P, Jin W (2013) Mining semantic relationships between concepts across documents incorporating Wikipedia knowledge. Advances in data mining. Applications and theoretical aspects, pp 70–84
30. Yan P, Jin W (2015) Improving cross-document knowledge discovery through content and link analysis of Wikipedia knowledge. In: Transactions on large-scale data-and knowledge-centered systems XXI, pp 161–184

Peng Yan is currently the founder and CEO of Askingdata (http://www.askingdata.com). He received his Ph.D in Computer Science from North Dakota State University in 2013, with his interests focused on Data Mining and Text Mining. His experience ranges from work with IBM creating automation frameworks for software test, research with North Dakota State University developing novel text mining algorithms, to work with WoWiWe Instruction Co. designing intelligent tutors and research with 3M Company in Data Analysis, Data Visualization and Data Mining.



Wei Jin is currently an associate professor at the Department of Computer Science, North Dakota State University. She received her Ph.D and MS degrees in Computer Science and Engineering from State University of New York at Buffalo, in 2008 and 2007, respectively. She also earned a ME degree from Institute of Computing Technology, Chinese Academy of Sciences in 2002.