Document Similarity Analysis via Involving Both Explicit and Implicit Semantic Couplings

Qianqian Chen, Liang Hu, Jia Xu, Wei Liu and Longbing Cao
Advanced Analytics Institute
University of Technology Sydney, Australia
Email: {Qianqian.Chen-1, Liang.Hu-2, Jia.Xu-3}@student.uts.edu.au
{Wei.Liu, LongBing.Cao}@uts.edu.au

***Abstract*—Document similarity analysis is increasingly critical since roughly 80% of big data is unstructured. Accordingly, semantic couplings (relatedness) have been recognized valuable for capturing the relationships between terms (words or phrases). Existing work focuses more on explicit relatedness, with respective models built. In this paper, we propose a comprehensive semantic similarity measure: Semantic Coupling Similarity (SCS), which (1) captures intra-term pair couplings within term pairs represented by patterns of explicit term co-occurrences in a document set, (2) extracts inter-term pair couplings between term pairs indicated by implicit couplings between term pairs through indirectly linked terms and paths between terms after term connections are converted to a graph presentation; and (3) semantic coupling similarity, integrating intra- and inter-term pair couplings towards a comprehensive capturing of explicit and implicit couplings between terms across documents. SCS caters for both synonymy and polysemy, and outperforms baseline methods consistently on all real data sets.**

I. Introduction

Textual information forms probably the major proportion of the big online data. With the rapid development of the Internet and Internet-based business, a critical opportunity, and accordingly challenge, is to understand the semantic similarity between terms (queries), text or documents by directly exploring their coupling relationships [1], [2], [3], [4] besides complex techniques such as natural language processing. This has shown to be promising as a recent popular research task in information retrieval [5], [6], ontological engineering [7], [8], [9], and document analysis [4], [10].

There are both intrinsic textual/linguistic complexity (such as natural language ambiguation) and various couplings (such as co-occurrence) [3] that drive the semantic relatedness between terms and documents. This makes it very challenging in analysing semantic similarity in information retrieval and document analysis, such as document clustering, document classification, and document query and filtering. Consequently, often a query hits a large number of documents in which few of them are relevant. This calls for the crucial need of further research on semantic similarity by deeply exploring the couplings within and between terms/documents, which is highly important for accurate information queries and document processing.

The problem of document semantic similarity can be further decomposed to explore the coupling relationships and similarity between terms (words or phrases) which form a document. This is to build a feature space that consists of all necessary terms with their relatedness captured and embedded in a similarity (or distance) learning model. Accordingly, a document analysis algorithm can then be built to analyze the semantic similarity between documents via exploring the intrinsic term couplings and similarity [2]. For this, a critical task is to measure intrinsic semantic couplings which is fundamental for information retrieval and other related natural language processing applications, such as text summarization, textual entailment, information extraction, etc.

Challenges are hidden in the various couplings between terms and documents, for instance, meronymy, antonymy, functional association, and others [11]. Recent efforts on measuring semantic relatedness can be roughly characterized into two categories: corpus-based statistical measures and topological measures. More specifically, semantic relatedness estimated by corpus-based statistical means such as vector space models [5], [6], [10], compute the co-occurrence frequency patterns of terms and textual contexts across corpus; probabilistic models [12], [13], [14], [15] are developed to discover the distribution properties of each term over topics and the topic distribution over each document. Instead, topological approaches [7], [9], [16], [17] capture the relatedness between terms or concepts by using ontologies to define the distance between them; most of such methods [9], [18], [19], [20], [21] rely on pre-existing knowledge resources that are represented by a directed or undirected graph consisting of vertices, for example, semantic networks and taxonomies.

A typical issue which hasn't been studied deeply is to effectively capture the sophisticated couplings not only between explicitly linked terms but also implicitly related terms in both statistical and topological aspects.

This paper addresses the above issue. We explore the semantic couplings of pairwise terms by involving three types of coupling relationships: (1) the intra-term pair couplings, reflecting the explicit relatedness within term pairs that is represented by the relation strength over probabilistic distribution of terms across document collection; (2) the inter-term pair couplings, capturing the implicit relatedness between term pairs by considering the relation strength of their interactions with other term pairs on all possible paths via a graph-based representation of term couplings; finally, (3) coupled semantic couplings, effectively combining the intra- and inter-relatedness. The corresponding term semantic similarity mea-

sures are then defined to capture such couplings for analyzing term and thus document similarity. This approach effectively addresses both synonymy (many words per sense) and polysemy (many senses per word) in a graphical representation, which is overlooked by previous models.

Specifically, the main contributions in our work lie in three factors:

- A statistical measure to capture the **semantic intra-term pair couplings** within term pairs by adapting a *relation strength function* to calculate the similarity between a pair of terms as per their probabilistic distributions, which counts the term pair occurrence frequency tpf-idf across the document set.
- A graph-based measure to capture the **semantic inter-term pair couplings** between term pairs by measuring the relation strength of every term pair distribution, which is calculated by the tpf-ipf weighting scheme on all possibly indirectly connected paths when term connections are plotted into a graph.
- An effective semantic couplings representation captures the comprehensive semantic relatedness across documents, via a **semantic coupling similarity (SCS)** measure that combines the intra- and inter-term pair couplings. It can be applied directly into document similarity analysis.

The proposed measures are compared with typical document representation models on various benchmark data sets in terms of document clustering performance. Our model produces outcomes that are great significant and exceed the performance of benchmark methods consistently on all data sets.

The remainder of this paper is organized as follows: Section 2 reviews and evaluates the related work of semantic relatedness representations from corpus-based and topological measures. Section 3 proposes the term semantic coupling measure. Section 4 shows its applications in document analysis. Section 5 demonstrates the experimental results of clustering analysis on real document sets. Finally, conclusion and future work are described in Section 6.

II. Related Work

Building a high-quality semantic relatedness representation model is a challenging task due to the complexity of nature language. A number of methods have been developed recently to exploit the semantic similarity and relatedness between terms to enhance efficiency of document representation. In this section, we provide a brief review on the basis of different method they use, roughly these methods can be characterized into the following categories.

A. Corpus-based Methods

Early research on corpus-based methods usually build on Bag of Words (BOW) model, it treats all the words in a document as index terms bounded with weights to reflect their importance, but disregards the order, structure, meaning, grammar, etc. of the words, only keeps multiplicity. Traditional document representations like VSM [22], gains the limitation of the term independence assumption, ignores the semantic relatedness between terms accordingly, which leads to a great loss of text semantic information.

On the basis of VSM, a diversity of extended models have been proposed like GVSM [6], CVM-VSM [5] and GTCV-VSM [10], they incorporate context vectors into VSM to model the term dependency. The term context vectors, which are not only determined by the occurrence frequency, but also the influence of terms in the semantic descriptions of other terms, store terms semantic similarities to the other terms. After widen to the corpus level, document representation is further semantically enriched, the semantic relation for terms can be achieved from the total contextual information across the whole document collection, as a result, for information retrieval the document-query similarities is based on the semantic-matching.

Some statistical document analysis with probabilistic topic-based models that using machine learning methods like LSA [13], PLSA [14], LDA [12], sLDA [15], improve the performance of information retrieval by overcoming unavoidable negative influences of BOW, such as sparseness, synonyms and polysemy. In topic modeling, documents are mixtures of topics, where a topic is a probability distribution over terms in a vocabulary. Semantic topics are concisely derived from the co-occurrence of a large number of terms from documents, and are used to transform documents to locate in low-dimensional topic space. Recently, most work of topic modeling focused on specific tasks, such as to consider the influences of context [23], [24], time [25], [26] and sentiment [27], [28].

An interesting effort made by Cheng et al. [4] in CRM is to capture the semantic relation of terms by considering both intra- and inter-term relations based on non-iidness learning [2]. The Jaccard distance is adapted to capture the intra-term relation as the similarity of terms, and the inter-term relation is computed by integrating the intra-term relation over a pair of terms with a link term. Although his approach considered the impact of link terms, still fails to avoid the negative effects of polysemy and synonymy, further only one link term used is not enough to express the semantic relatedness completely.

B. Topological Methods

Semantic relatedness is estimated by defining a topological similarity, by using lexical ontologies to measure the distance between terms or concepts. These approaches rely on handcrafted resources such as thesauri, taxonomies, semantic network or encyclopedias, as the context of comparison [8].

Previous semantic measures based on lexical ontologies use a taxonomy(tree), which is a hierarchical network representation consists of concepts and relations between these concepts, to compute the semantic similarity between two concept nodes by some measures of distance. In early research the main assumption is that to capture the the similarity between two concepts is to find the shortest-path linking the two concept nodes in a taxonomy graph [17]. More advanced methods consider the semantic relatedness of concepts based on the information content(IC) they share on taxonomy structure. Sanchez et al. [9] proposed a IC-based model to better capture the semantic relatedness in an ontology for the particular

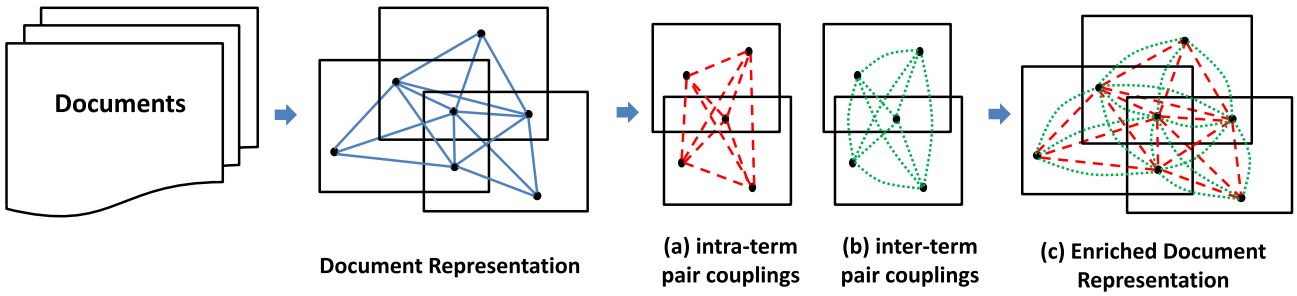

Fig. 1: An overview of term pair semantic coupling analysis

concept, they computes the IC of one term by the ratio of the number of its hypernyms divided by the number of its descendants in WordNet.

In contrast to previous works that focus on one relation(is-a) or other taxonomic relations, a number of semantic measures have been proposed to capture different type of semantic relations considering both hierarchical and non-hierarchical concepts in an ontology(graph). Maguitman and Menczer [16] defined a graph-based semantic similarity measure on ODP that generalizes the tree-based similarity, used MaxProduct fuzzy composition to define fuzzy membership value for each concept, then the semantic similarity of two concepts can be calculated from the fuzzy membership matrix. Another graph-based approach on GBSRO [7] proposed six stages to deal with particular aspect of relatedness and presented by adjacency matrices, and all matrices are integrated into one to represent the final semantic relatedness across all concepts.

Another graph-based model is semantic network, which is a simple representation scheme reflects semantic relations between concepts, that uses a graph of labeled nodes and labeled, directed arcs to encode knowledge. It was popular in the 60s and 70s, nowadays is further developed and widely applied into Semantic Link Network(SLN) [29], Resource Description Framework(RDF) [30], and WordNet [31], [32], [17].

Recently, topological models are created on semantic networks based on various sources of background knowledge-which are also the combination of topological models with statistical methods, Thesaurus-based measures [9], [18], for example, use WordNet as a broad coverage lexical network to measure different type of relation and textual entailment; Wikipedia-based measures, like WikiRelate [20], ESA [19], WLM [21], TSA [33], and CLEAR [34], combine both the familiar methods that previously used to WordNet and corpus-based techniques to represent the relatedness of text on a much bigger vocabulary.

These topological models study on the semantic relation among objects in the process of mapping the physical world into the cyber world, and the various practical applications make it efficient for users to define semantic relation based on the representation built by these models.

In summary, different efforts have been made to address semantic similarity issues from various aspects. Due to the intrinsic complexities of natural language, there is more work to do on deeply exploring term semantic relationships and representing semantic similarity. SCS is built to solve natural language ambiguity, non-iidness theory [2] is adapted here to handle unstructured textial data and complex relationships of concepts. In the next section, our proposed research methodology is discussed, which attempt to capture the semantic relatedness in a coupled thought, by combining the statistical-based and graph-based method together, to detailedly mine the explicit and implicit relatedness of terms pairs.

III. Term Pair Semantic Coupling Analysis

In this section, a novel approach is proposed to capture the semantic couplings of term pairs from two aspects: the semantic intra-term couplings and the semantic inter-term couplings. Figure 1 illustrates the intuitionistic understanding of our measure: (a) it calculates the semantic intra-term couplings of term pairs by considering their occurrence frequency across the document set; (b) it further constructs bridges consisted of linked term pairs between term pairs to compute the semantic inter-term couplings; and (c) it integrates the intra- and inter-term couplings to capture the complete semantic couplings and similarity.

A. Semantic Intra-couplings within Term Pairs

The semantic intra-term coupling within term pairs (we call intra-term pair coupling, intra-term coupling, or simply intra-couplings in this paper) is to explore the explicit semantic relatedness between terms. Typical research on term explicit semantic relatedness is to consider the statistical analysis of term co-occurrence patterns. It assumes that terms are regarded relational if they co-occur in the same document; the more frequently they co-occur, the stronger relation they have. Accordingly, the explicit relation between terms can be estimated based on the term co-occurrence frequency across all documents.

The weighting scheme tf-idf is used as a weighting factor to reflect the importance of a term to a document in a collection or corpus. The term frequency $tf(t, d)$ is the number of times

term t occurs in document d, the document frequency $df(t)$ is the number of documents in which t occurs at least once, and the inverse document frequency idf can be calculated as $idf(t, D) = log(\frac{|D|}{df(t)})$, where $|D|$ is the total number of documents. idf is low if t occurs in many documents and will be high if it occurs in few documents. Then $tf\text{-}idf$ is computed as the product $tfidf(t, d, D) = tf(t, d) \times idf(t, D)$. A high weight in $tf\text{-}idf$ is reached by a high term frequency in the given document and a low document frequency of the term in the whole collection of documents, which proves that using $tf\text{-}idf$ weighting scheme to consider term relation is not only based on the co-occurrence frequency but also takes the term discriminative ability into account.

However, these methods have two main limitations. One is that they place undue emphasis on the documents where terms co-occur; the other is that $tf\text{-}idf$ based on one single term may lead to synonymy and polysemy, due to the semantic meaning of one term in different documents can be various. To solve these problems, we propose the $tpf\text{-}idf$ scheme as an improvement of $tf\text{-}idf$, defined as follows:

Definition 1. tpf**-**idf, short for *term pair occurrence frequency - inverse document frequency*, reflects the importance of a term pair to a document in a corpus. tpf counts the number of times a term pair occurs in a document. The $tpf\text{-}idf$ scheme is formatted as:

$$tpfidf((t_i, t_j), d, D) = tpf((t_i, t_j), d) \times idf((t_i, t_j), D) \quad (1)$$

where (t_i, t_j) stands for a term pair, and d is a single document in a document collection D.

The term pair occurrence frequency matrix M_{tpf} is represented as:

$$M_{tpf} = \begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_K \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & \cdots & t_K \end{array} \\ \begin{pmatrix} 0 & tpf_{12} & \cdots & tpf_{1K} \\ tpf_{21} & 0 & \cdots & tpf_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ tpf_{K1} & tpf_{K2} & \cdots & 0 \end{pmatrix} \end{array}$$

which represents the occurrence frequency of every term pair in the document set. K is the total number of terms in the document collection.

By using $tpf\text{-}idf$ scheme, terms appear as pairs, the meaning of a single term is more semantically complete compared with $tf\text{-}idf$. It is used to depict the real explicit relatedness of term pairs by adapting statistical distance measures for a solid statistical significance.

Firstly, for $\forall(t_k, t_i) \in D$ $(k, i \in [1, K], k \neq i)$, we represent:

$$P^{Ia}(t_k|t_i) = \frac{tpfidf_{(t_k,t_i)}}{\sum_{k=1}^{K} tpfidf_{(t_k,t_i)}} \quad (2)$$

as the probability of the term pair (t_k, t_i) in document set D, $tpfidf_{(t_k,t_i)}$ is $tpf\text{-}idf$ of the term pair (t_k, t_i).

Then the probabilities over all term pairs given t_i are defined as:

$$\begin{aligned} P^{Ia}(t_i) &= \{P^{Ia}(t_1|t_i), P^{Ia}(t_2|t_i), \cdots, P^{Ia}(t_k|t_i)\} \\ &= \{P^{Ia}(t_k|t_i)\}_{k=1}^{K} \end{aligned} \quad (3)$$

Furthermore, we adapt *relation strength similarity* [35] to estimate the intra-term couplings of term pairs. The relation strength similarity defines how close two adjacent vertexes are. It supports various similarity and distance measures, their conversions are used to determine the relative closeness of term pairs that being considered.

Definition 2. Given a document set D, a term pair (t_i, t_j) in D, the **intra-term pair couplings (IaR)** of (t_i, t_j) is represented on a *relation strength function* (RS) as follows:

$$IaR(t_i, t_j) = RS(P^{Ia}(t_i), P^{Ia}(t_j)) \quad (4)$$

where *cosine similarity* is introduced to quantify the similarity between $P^{Ia}(t_i)$ and $P^{Ia}(t_j)$ which are the probabilities over all term pairs given t_i and t_j respectively.

The value of $IaR(t_i, t_j)$ falls into $[0, 1]$, $IaR(t_i, t_j) = 1$ when $t_i = t_j$. This measure is symmetric, generally $IaR(t_i, t_j) = IaR(t_j, t_i)$. A larger value indicates more similar distributions of t_i and t_j, it leads to a stronger explicit intra-couplings. The procedure of computing semantic intra-couplings of term pair (t_i, t_j) is summarized in Algorithm 1.

Algorithm 1: Semantic Intra-term Couplings

Input: Document-Term matrix D
Output: $IaR(t_i, t_j)$

```
Construct M_tpf;
for term t_i in M_tpf do
    for term t_j (t_j ≠ t_i) in M_tpf do
        Compute P^Ia(t_j|t_i) (Equation (2));
    end
    Compute P^Ia(t_i) (Equation (3));
end
for term pair (t_i, t_j) (t_i ≠ t_j) do
    Compute IaR(t_i, t_j) (Equation (4));
end
```

Semantic intra-term coupling captures the explicit relatedness of term pairs by considering their occurrence frequency patterns and probability distributions across the document set; especially it considers the relatedness of terms that appear individually in different documents. However, this method still lacks of the exploration of underlying relatedness of term pairs, which results in incomplete semantic couplings.

The implicit coupling of term pairs is addressed in the following subsection by taking the similarity of their interactions with other term pairs into account.

B. Semantic Inter-couplings between Term Pairs

Assume that a document set may be drawn as a graph with nodes and edges to reflect the terms and their relatedness separately, the intra-term coupling introduced above only captures the explicit relatedness of two adjacent nodes in the graph, but fails to consider the relatedness of term pairs in a global view, for the reason that the intra-term coupling fails to capture the semantic relatedness of term pairs by taking the interactions of other terms in the document set into consideration. In this section, we propose an approach to capture this kind of implicit relatedness based on the graph theory.

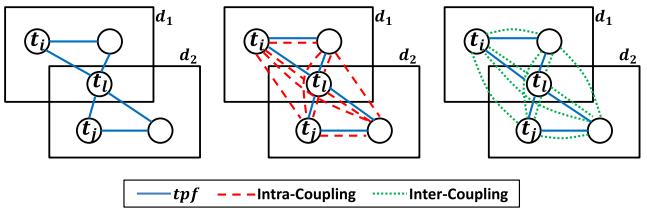

Fig. 2: Semantic couplings within and between term pairs

1) Term Pair Frequency Graph: On the basis of M_{tpf}, a graph can be constructed as a representation of terms and their frequency pattern. As shown in Figure 2, the term pair frequency graph G_{tpf} is an ordered pair,

$$G_{tpf} = (T, E_{tpf})$$

comprising a set T of terms as vertexes, $T = \{t_k | k \in [1, K]\}$; together with a set E_{tpf} as edges to reflect the tpf of every term pair, which are 2-element subsets of T. G_{tpf} is not a complete graph, some term pairs are unconnected by an edge, for example, t_i and t_j in Figure 2, meaning that t_i and t_j do not co-occur in the same document, i.e. $tpf(t_i, t_j) = 0$.

To avoid ambiguity, this type of graph may be described precisely as undirected and simple.

2) Intra-coupling Graph: Based on the intra-couplings of all term pairs across document collection, an undirected and simple graph is constructed to represent terms and their intra-couplings, formalized as:

$$G_{IaR} = (T, E_{IaR})$$

where the term set T stands for vertexes, the edge set E_{IaR} stands for edges to draw lines between every two vertexes. An edge is related with two vertexes, and the intra-coupling is represented as an unordered pair of the vertexes with respect to the particular edge.

G_{IaR} is a complete graph of G_{tpf}, every two vertexes are related, which means the intra-coupling captures the explicit relatedness of all term pairs, including the two terms from different documents. However, to reflect the semantic relatedness of term pairs completely, G_{IaR} fails to provide a reasonable way to consider the influence of all other term pairs. This triggers the question of how to draw a special "line" to connect them, namely to capture the implicit relatedness of them, which will be addressed in following sections.

3) Inter-coupling Graph: Firstly, for a term pair in G_{tpf}, no matter it is connected or not, intuitively, it can be related through other terms, as in the first part of Figure 2, there exist paths starting at t_i and ending at t_j, $t_i \to t_l \to t_j$ for instance. Therefore, to capture the inter-couplings between term pairs across a document set (we call inter-term pair coupling, inter-term coupling, or simply inter-couplings), no matter how the two terms appear separately in different documents or they co-occur in some documents, their interactions with other terms play a major role. In other words, we discover routes containing other terms to connect every term pair in G_{tpf}. The definition of *path* is given as:

Definition 3. A **path** is a subgraph of G_{tpf}, containing a finite sequence of edges which connect a sequence of vertexes,

$$\begin{aligned} Path(t_i, t_j) = \big\{ (T^P_{i;j}, E^P_{i;j}) \mid\ & t_i, t_{l_1}, \cdots, t_{l_n}, t_j \in T^P_{i;j}, \\ & e_{il_1}, e_{l_1 l_2}, \cdots, e_{l_n j} \in E^P_{i;j},\ t_i \neq t_j, \\ & T^P_{i;j} \subset T,\ E^P_{i;j} \subset E_{tpf},\ n \in [1, \theta] \big\} \end{aligned} \tag{5}$$

where t_i is the initial vertex and t_j is the terminal vertex, t_{l_n} stands for the terms between them on $Path(t_i, t_j)$, n is the number of these terms. θ is a user-defined threshold to limit the number of t_{l_n}, i.e., the length of a path. As shown in the third part of Figure 2, the bold lines are the paths connecting term pair (t_i, t_j).

The definition of *path* has three critical assumptions:

- The paths of a term pair at least go through one another term, edges that connect term pairs directly are not defined as paths;
- The longer the path is, the weaker the coupling is, only the paths with their length falling into $[2, \theta + 1]$ are chosen;
- The path here defined is simple, meaning that no vertexes (and thus no edges) are visited repeatedly.

We further define these vertexes between two terms on one path as:

Definition 4. All n vertexes between t_i and t_j on $Path(t_i, t_j)$ construct a **link term set** T_{link}, formalized as:

$$T_{link} = \big\{ t_l \mid t_l \in T^P \backslash (t_i, t_j),\ T^P \in Path(t_i, t_j) \big\} \tag{6}$$

where T^P contains all vertexes on $Path(t_i, t_j)$. To be simple, link terms are the total terms on a path except the first and last vertexes. So for all term pairs in G_{tpf}, their semantic inter-term couplings can be captured by considering the relatedness of every term pair on all possible paths.

Furthermore, as we discussed above, it is understandable that every term pair is inter-related since there always exists at least one path from one term to the other through link terms. The inter-coupling graph G_{IeR} based on G_{tpf} is represented as:

$$G_{IeR} = (T, E_{IeR})$$

where E_{IeR} stands for the inter-coupling of every two terms, which is calculated by the relatedness of all term pairs on all possible paths between them on G_{tpf}. The detailed algorithm of inter-coupling is concluded in the following section.

4) Semantic Inter-couplings between Term Pairs: Similarly, to calculate the inter-couplings of term pairs, we need to concern the relation strength of every term pair on all possible paths. tpf-idf scheme is further improved as tpf-ipf to represent the impact of a term pair to paths, which is defined as:

Definition 5. tpf-ipf, short for *term pair occurrence frequency - inverse path frequency*, reflects the importance of a term pair to all possible paths between paired terms. For a term pair, ipf is computed by path frequency pf, which counts the

number of paths in which the term pair occurs. The tpf-ipf scheme is formatted as:

$$tpfipf((t_i,t_j),d,m) = tpf((t_i,t_j),d) \times log(\frac{m}{pf(t_i,t_j)}) \quad (7)$$

where m is the total number of $Path(t_i,t_j)$.

According to the tpf-ipf scheme, the weight of a random term pair (t_k,t_i) in graph G_{IeR} is, for $\forall(t_k,t_i) \in D(k,i \in [1,K], k \neq i)$,

$$W(t_k|t_i) = \frac{tpfipf_{(t_k,t_i)}}{\sum_{k=1}^{K} tpfipf_{(t_k,t_i)}} \quad (8)$$

where $tpfipf_{(t_k,t_i)}$ is the tpf-ipf of the term pair (t_k,t_i),

Secondly, for term pairs in G_{tpf}, no matter whether they are connected or not, there are various paths going through link terms to connect them. For $\forall t_k,t_i \in T, t_{l_n} \in T_{link}(k,i,l_n \in [1,K], k \neq i \neq l_n)$, the weight of one path through $t_{l_1},\cdots,t_{l_n}$ between term pair (t_k,t_i) in G_{tpf} is:

$$W_{t_{l_1},\cdots,t_{l_n}}(t_k|t_i) = W(t_{l_1}|t_i) \cdot \prod_{p=1}^{n-1} W(t_{l_{p+1}}|t_{l_p}) \cdot W(t_k|t_{l_n}) \quad (9)$$

In this way, on all possible paths from t_i to t_k, those edges passed more frequently, the value of tpf-ipf is larger, and it has more weight. In addition, a longer path goes through more edges, the value of product is smaller, and the weight of a long path is lighter.

Thirdly, for m possible paths from t_i to t_k, we acquire the weight of m paths between term pair (t_k,t_i) in G_{tpf} as:

$$W_m(t_k|t_i) = \sum_{q=1}^{m} W_q(t_k|t_i) \quad (10)$$

We normalize it as the weight of a term pair on all possible paths divided by the weight of all term pairs on all possible paths in graph, it is the probability of a term pair (t_k,t_i) on all m paths:

$$P^{Ie}(t_k|t_i) = \frac{W_m(t_k|t_i)}{\sum W_m(t_k|t_i)} \quad (11)$$

Then, the probability distribution of t_i, consisted of the probabilities over all term pairs on m possible paths for given t_i, is formalized as:

$$\begin{aligned} P^{Ie}(t_i) &= \{P^{Ie}(t_1|t_i), P^{Ie}(t_2|t_i), \cdots, P^{Ie}(t_k|t_i)\} \\ &= \{P^{Ie}(t_k|t_i)\}_{k=1}^{K} \end{aligned} \quad (12)$$

Finally, the inter-coupling $IeR(t_i,t_j)$ of a term pair (t_i,t_j) in D is represented as the relation strength of two possibility distributions to measure the similarity between them,

Definition 6. Given a document set D, the **inter-term couplings (IeR)** between a term pair (t_i,t_j) in D is represented in terms of relation strength considering all possible paths $Path(t_i,t_j)$ with n link terms, $n \in [1,\theta]$.:

$$IeR_n(t_i,t_j) = RS_n(P^{Ie}(t_i), P^{Ie}(t_j)) \quad (13)$$

where $IeR_n(t_i,t_j)$ is the nth order inter-coupling which stands for the relation strength of (t_i,t_j) with n link terms.

$IeR(t_i,t_j)$ is the integration of n order inter-coupling of (t_i,t_j),

$$IeR(t_i,t_j) = \frac{1}{n}\sum_{n=1}^{\theta} IeR_n(t_i,t_j) \quad (14)$$

The value of $IeR(t_i,t_j)$ is bounded to $[0,1]$, the larger the value is, the more similar distributions t_i and t_j have, the closer the terms are inter-related.

Algorithm 2 calculates the semantic inter-term couplings $IeR(t_i,t_j)$ of term pairs (t_i,t_j), which considers both directly and indirectly linked terms.

Algorithm 2: Semantic Inter-term Couplings

Input: Document-Term matrix D, User-defined threshold θ
Output: $IeR(t_i,t_j)$

```
Construct M_tpf;
for term t_i in M_tpf do
    for term t_j(t_j ≠ t_i) in M_tpf do
        Search all possible paths Path(t_i,t_j) with n
          link terms, n ∈ [1,θ];
        Compute P^Ie(t_j|t_i) (Equation (11));
    end
    Compute P^Ie(t_i) (Equation (12));
end
for term pair (t_i,t_j) (t_i ≠ t_j) do
    Compute IeR(t_i,t_j) (Equation (14));
end
```

Accordingly, the semantic coupling is further enriched by exploring the semantic inter-term couplings, due to that it is not based on terms themselves, but on interactions with all other terms in a document set.

C. Semantic Couplings of Term Pairs

The semantic intra-term coupling captures the explicit relatedness of term pairs based on the occurrence frequency pattern of every term pair across corpus, the semantic inter-term coupling further explores the implicit relatedness by considering the occurrence frequency patterns of all linked term pairs on all possible paths. Further, they are integrated as a *Semantic Coupling Similarity* (SCS) , to capture the semantic relatedness of term pairs completely and comprehensively.

Definition 7. Given a document set D, the **Semantic Coupling Similarity (SCS)** of a term pair (t_i,t_j) in D is:

$$SCS(t_i,t_j) = (1-\alpha) \cdot IaR(t_i,t_j) + \alpha \cdot IeR(t_i,t_j) \quad (15)$$

where $IaR(t_i,t_j)$ and $IeR(t_i,t_j)$ represents the intra- and inter-coupling of (t_i,t_j), respectively. $\alpha \in [0,1]$ is a parameter to control the weight of intra- and inter-coupling, here we take the simplest way, i.e. linear combination to show the performance.

The value of $SCS(t_i,t_j)$ is bounded in $(0,1]$, it equals to 1 when $t_i = t_j$. The higher the value is, the stronger

semantic coupling exists, the closer they are semantic-related, the more similar the terms are. Five important properties are further identified from the calculation procedure and served as a foundation of our SCS approach.

Property 1: Identity Property

The semantic coupling similarity of a term pair reaches the highest value 1 when the terms have the identical meaning, which means the distance between them is zero.

Property 2: Symmetrical Property

On the undirected graphs G_{IaR}, G_{IeR} and G_{SCS}, there is only one type of relation for term pairs on each graph, then the order is disregarded, so that the semantic couplings for term pairs are symmetrical.

Property 3: Positive Property

The value of $SCS(t_i, t_j)$ is always non-negative and larger than 0, ranged in $(0, 1]$.

Property 4: Minimal Distance Property

An early edge-based model of semantic relatedness assumes that the semantic distance is based on the number of edges between terms [17], in other words, a shorter distance controls a higher similarity. Our approach also follows the *Shortest Path Length* assumption, for term pair (t_i, t_j) $(t_i \neq t_j)$ on G_{IaR} and G_{SCS}, the minimal distance equals to 1, while on G_{IeR}, it equals to 2.

Property 5: A Path's Finite Length Property

As we identify the SCS as a path length-relative measure, more closely connected term pairs are more semantically related. Consequently, we set a user-determined threshold to limit the maximum length of path to improve computational efficiency.

With the combination of intra- and inter-term couplings, both explicit and implicit couplings of term pairs are discovered. This remarkably captures the semantic richness of documents. Specifically, the main contributions of our proposed SCS measure are summarized as follows:

- The intra-term coupling is calculated from relation strength of probability distributions of terms, it especially fixes the lack of relatedness of term pairs that cross different documents; the inter-term coupling is introduced to capture the implicit couplings of term pairs, which takes the full advantage of the interactions with other terms in a document set.
- Our inter-term coupling method is based on weighted paths with limited length. On one hand, it distinguishes strong link terms from weak link terms, the strong link terms which are visited frequently on all possible paths occupy higher weights; on the other hand, it emphasizes that less link terms build the closer relatedness, only strong link terms are reserved so that the efficiency of calculation is improved.
- SCS is helpful for managing the synonymy and polysemy for two reasons: (1) intra- and inter-term couplings are based on term pair occurrence frequency patterns across corpus (tpf-idf) and all possible paths (tpf-ipf) respectively, accordingly the term-pair occurrence frequency patterns appear across a document set or all possible paths instead of each single term, the semantic meaning for every term pair is richer than individual terms; (2) coupling similarity is built on RS between term distributions. For terms that are semantically similar, their distributions are similar, the value calculated via RS is large; for terms that are subject to synonymy and/or polysemy, the probability values of specific term pairs could be close, but the probability distributions over all term pairs in document collection or all possible paths are quite different. Consequently, RS is surely weaker than real similar term pairs.

In summary, SCS measure represents documents based on the comprehensive couplings of term pairs. In contrast to previous work, SCS can deal with unstructured data and terms coupled in terms of various reasons, addressing natural language ambiguity problems.

IV. Coupled Document Analysis

With SCS, M_{tpf} is further transferred into a $K \times K$ semantic coupling similarity matrix M_{SCS}, whose elements reflect the couplings of each term pair. It is used for document analysis.

$$M_{SCS} = \begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_k \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & \cdots & t_k \end{array} \\ \begin{pmatrix} 1 & SCS_{12} & \cdots & SCS_{1k} \\ SCS_{21} & 1 & \cdots & SCS_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ SCS_{k1} & SCS_{k2} & \cdots & 1 \end{pmatrix} \end{array}$$

Firstly, each document is defined as the mapping:

$$\phi : d \rightarrow \phi(d) = \big\{P(t_1, d), P(t_2, d), \cdots, P(t_k, d)\big\} \quad (16)$$

where $P(t_k, d)$ is the probability of term t_k in document d.

$$P(t_k, d) = \frac{tf(t_k, d)}{\sum_{k=1}^{K} tf(t_k, d)} \quad (17)$$

Secondly, documents are further represented in coupled semantic space considering SCS,

$$\widetilde{\phi}(d) = \phi(d) M_{SCS} \quad (18)$$

Then the document similarity $Sim(d_i, d_j)$ is the product in this new vector space.

$$\begin{aligned} Sim(d_i, d_j) &= \phi(d_i) M_{SCS} M_{SCS}^T \phi(d_j)^T \\ &= \widetilde{\phi}(d_i) \widetilde{\phi}(d_j)^T \end{aligned} \quad (19)$$

Thus, the new document representation $\widetilde{\phi}(d)$ is computed efficiently directly from the original data using Equation (18), documents are represented in a new coupled semantic feature space based on the term occurrence frequency pattern and comprehensive term pair semantic couplings.

TABLE I: Results of different models on various data sets

Model\Data sets	D1: Reuter				D2: TDT2				D3: WebKB			
	Purity	RI	F1-measure	NMI	Purity	RI	F1-measure	NMI	Purity	RI	F1-measure	NMI
HAC models with average linkage												
BOW	0.7589	0.7147	0.6592	0.4949	0.7646	0.8591	0.7036	0.6403	0.3862	0.2873	0.4147	0.2079
LSA	0.7631	0.7541	0.6256	0.5038	0.7918	0.8830	0.7535	0.7347	0.3972	0.4045	0.4447	0.3015
LDA	0.8119	0.8002	0.6612	0.5631	0.8127	0.9054	0.8082	0.7473	0.5585	0.5640	0.5350	0.3157
HDP	0.8194	0.7873	0.6397	0.5579	0.8396	0.8616	0.8008	0.7510	0.5691	0.5380	0.5334	0.3064
CRM	0.8152	0.8124	0.7163	0.5673	0.8408	0.8961	0.8094	0.7283	0.5668	0.4929	0.4444	0.2944
CHAC	**0.8320**	**0.8310**	**0.7414**	**0.5741**	**0.8450**	**0.9259**	**0.8158**	**0.7590**	**0.5713**	**0.5925**	**0.5578**	**0.3678**
HAC models with complete linkage												
BOW	0.6125	0.5280	0.5671	0.4083	0.6167	0.7616	0.5251	0.5823	0.4079	0.3492	0.4433	0.2348
LSA	0.6807	0.6714	0.5688	0.4437	0.6782	0.8196	0.5742	0.6407	0.5753	0.6062	0.4409	0.2538
LDA	0.7562	0.7632	0.5860	0.4777	0.8041	0.8490	0.7338	0.6802	0.6047	0.6121	0.5004	0.2935
HDP	0.7835	0.7426	0.5439	0.4870	0.7782	0.8477	0.6196	0.6474	0.5952	0.6840	0.5029	0.2871
CRM	0.7398	0.7130	0.5667	0.4598	0.7867	0.8030	0.7325	0.6609	0.6168	0.5921	0.5036	0.2935
CHAC	**0.8028**	**0.7748**	**0.6002**	**0.5048**	**0.8202**	**0.9003**	**0.7594**	**0.7188**	**0.6398**	**0.6911**	**0.5082**	**0.3493**

$\widetilde{\phi}(d)$ can be widely applied to document clustering, classification and information retrieval, etc. Here we illustrate the application of $\widetilde{\phi}(d)$ into *hierarchical agglomerative clustering* (HAC), to generate a SCS-based HAC (CHAC), catering for both average and complete term linkages, which measure the *cosine similarity* between two clusters based on the average and minimum of their document similarities, respectively.

V. Experiments and Evaluation

In this section, SCS is incorporated into HAC as CHAC with both average and complete linkages, and compared with similar and typical document representations. 5-fold cross-validation is employed to present parameter tuning and automatically estimate the optimal value of parameter α in Equation (15) on various data sets.

A. Experimental Settings

Three most popular text data sets are chosen in our experiments: Reuters-21578[1], TDT2[1] and WebKB[2]. Detailed information of data sets are summarized in Table II.

TABLE II: Characteristics of data sets

Data Sets	Name	n	m	m_{doc}	k	n_c
D1	Reuters-21578	7085	8933	42	8	886
D2	TDT2	6825	8000	118	7	975
D3	WebKB	4087	7770	79	4	1021

n, m and k are the number of documents, terms and class, respectively. m_{doc} is the average number of terms per document, n_c is the average number of documents per class.

Four generally accepted evaluation metrics of clustering: *Purity*, *Rand Index* (RI), F_1 *measure* and *Normalized Mutual Information* (NMI) are adopted to evaluate the performance of CHAC with baseline approaches. Higher values indicate better clustering solutions.

CHAC is compared with BOW, LSA [13], LDA [12], HDP [36] and CRM [4]. We first use various models to represent document or calculate the document similarity, then apply HAC to either the document representation or the similarity matrix. The MATLAB function *linkage* is used.

The 5-fold cross validation is employed in our experiments, and each fold composes of 80% of data for training and 20% for testing.

B. Experimental Results

Here we compare the performance of CHAC for one link term with baselines on three data sets.

As we mentioned in Equation (15), α is used as a parameter to control the weight of inter-term couplings in SCS. In experiments, it is set from 0 to 1 at an increment of 0.05, where its value associated with the best result in each data set is chosen, for average-link CHAC, D1: $\alpha = 0.25$, D2: $\alpha = 0.35$, D3: $\alpha = 0.50$; for complete-link CHAC, D1: $\alpha = 0.45$, D2: $\alpha = 0.25$, D3: $\alpha = 0.40$. The analysis of parameter tuning and automatically estimation of the optimal value of α will be discussed in the next section.

The technical performance for different document representation models on testing data is evaluated and concluded in Table I. Specifically, for each model, each cell illustrates the practical clustering results considering various evaluation metrics. For each evaluation metric, a larger value indicates a more accurate and reliable model. Obviously, CHAC on both average and complete linkages achieves significant improvement and outperforms all models by considering the given clustering evaluation criteria on various data sets.

The reason lies in that SCS offers a deeper way to capture the semantic relations of term pairs. Unlike BOW, LSA, LDA and HDP methods which overlook the internal interactions between terms, SCS accomplishes a comprehensive consideration of not only the intra (explicit) -term couplings which are captured via term co-occurrence frequency patterns, but also the effect of inter (implicit) -term couplings to represent the indirect contact between terms. SCS also addresses the term ambiguity problems in CRM. Specifically, for a single document, the semantic relation between terms is more fully represented to capture richer semantic contents in a document, so as to achieve better clustering results.

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

[2] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

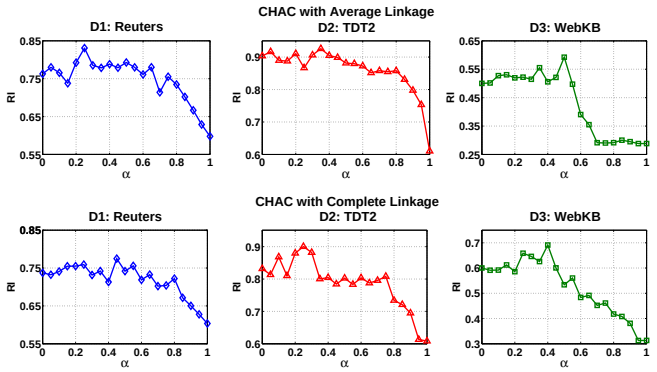

Fig. 3: The tuning of parameter α

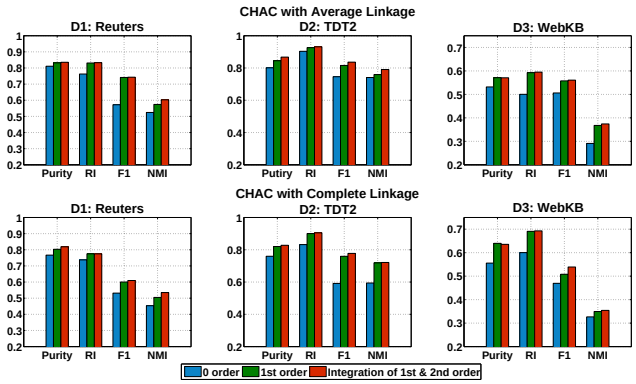

Fig. 4: The impact of inter-coupling ordering

C. Tuning Parameter α

As the parameter α controls the effect of intra- and inter-term couplings, it is essential to optimize α to achieve the best possible performance. By exploiting 5-fold cross-validation, the value of α is automatically estimated. The RI scores calculated from each fold are averaged to reflect the performance of clustering on testing sets.

The growth trends of RI scores on each value of α ranged from 0 to 1 with the increment of 0.05 on different data sets are represented in Figure 3. In particular, it keeps growing from the beginning, then starts to descend after it reaches the peak, it shows that the RI results achieve the best performance at a peak point with respect to a certain value of α. This proves that the combination of intra- and inter-coupling achieves better performance than using intra-coupling only (when $\alpha = 0$) or inter-coupling only (when $\alpha = 1$). For each data set, the automatic selection of α equals to 0.25, 0.35, 0.50 for average-link CHAC, and the corresponding RI scores are 0.8310, 0.9259, 0.5925. For complete-link CHAC, α equals to 0.45, 0.25, 0.40, the corresponding RI scores are 0.7748, 0.9003, 0.6911, respectively.

D. Inter-coupling Ordering

SCS introduces the innovative concept of inter-term couplings with multi-link terms. Here we evaluate the influence of inter-term couplings by comparing the clustering results of inter-couplings with different ordering.

The inter-term coupling algorithm strongly relies on the interactions between link terms. To test the contribution of using link terms and deeply analyze the impact of inter-coupling ordering, we present the comparison of clustering performance by considering 0 (intra-coupling only) order, 1st order and the integration of 1st and 2nd order inter-couplings on three data sets. Similarly, for the experiments of CHAC based on the integration of 1st and 2nd order inter-coupling, we retain the best clustering performance where α follows a specific value. For average-link CHAC, D1: $\alpha = 0.30$, D2: $\alpha = 0.15$, D3: $\alpha = 0.60$, for complete-link CHAC, D1: $\alpha = 0.30$, D2: $\alpha = 0.25$, D3: $\alpha = 0.10$.

In Figure 4, the bar charts compare the impact of inter-coupling ordering in terms of clustering evaluation metrics on the selected data sets. Overall, for every evaluation metric, all three data sets are sensitive to the number of link terms. The performance of CHAC on 1st order inter-coupling has been greatly improved compared to the 0 order inter-coupling; while the trends on the integration of 1st and 2nd order inter-coupling are not so remarkable. The reason is, when no link term exists, relatedness between terms only reflect the explicit relations. After introducing inter-couplings, richer interactions between terms are disclosed, which are abundant or diversified, leading to improved performance. In contrast, more link terms and longer path reflect weaker indirect influence of term couplings.

Based on the significant progress achieved by CHAC on one link term and time complexity, we recommend that CHAC on 1st order inter-term coupling is likely acceptable to our need.

VI. Conclusions

In this paper, we have proposed a novel semantic coupling similarity measure SCS to completely and comprehensively capture the semantic relatedness of term pairs. SCS achieves this in terms of a four-step procedure: (1) Capturing the semantic intra-term couplings of term pairs based on its occurrence frequency information across a document set; (2) Capturing the semantic inter-term couplings of term pairs based on the interactions with link terms on all possible paths after term connections are plotted to a graph structure; (3) Via an optimal combination, a fully semantic coupling of term pairs is achieved; and (4) The original document set can then be represented by a semantic coupling similarity matrix to measure the document similarity.

Experiments on real data sets have shown that SCS-based hierarchical agglomerative document clustering achieves impressive improvement over typical document clustering methods. More specifically, although a path showing term linkage could be quite long, our comprehensive test shows that we may only need one step of term linkage for most of cases for an acceptable level of running time. We are working on theoretical analysis of the effect of the number of link terms, and comparing SCS with the most recent machine learning methods for latent semantic analysis and document classification.

This research opens new opportunities to deeply explore semantic similarity, such as introducing the coupling idea into

the calculation of document pair relatedness, and estimating the time complexity brought by the increase of link terms also needs further improvement.

VII. Acknowledgements

This work is sponsored in part by Australian Research Council Discovery Grant (P130102691).

References

[1] O. Y. Cao, Longbing and P. Yu, "Coupled behavior analysis with applications," *IEEE Trans. on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1378–1392, 2012.

[2] L. Cao, "Non-iidness learning in behavioral and social data," *The Computer Journal*, p. bxt084, 2013.

[3] ——, "Coupling learning of complex interactions," *Information Processing & Management*, 2014.

[4] X. Cheng, D. Miao, C. Wang, and L. Cao, "Coupled term-term relation analysis for document clustering," in *IJCNN*, 2013, pp. 1–8.

[5] H. Billhardt, D. Borrajo, and V. Maojo, "A context vector model for information retrieval," *JASIST*, vol. 53, pp. 236–249, 2002.

[6] S. K. M. Wong, W. Ziarko, and P. C. N. Wong, "Generalized vector spaces model in information retrieval," in *Proceedings of the 8th Annual International ACM SIGIR Conference*, 1985, pp. 18–25.

[7] A. Hawalah and M. Fasli, "A graph-based approach to measuring semantic relatedness in ontologies," in *WIMS*, 2011, p. 29.

[8] P. Li, H. Wang, K. Q. Zhu, Z. Wang, and X. Wu, "Computing term similarity by large probabilistic isa knowledge," in *CIKM*, 2013, pp. 1401–1410.

[9] D. Sánchez, M. Batet, and D. Isern, "Ontology-based information content computation," *Knowledge-Based Systems*, vol. 24, pp. 297–303, 2011.

[10] A. Kalogeratos and A. Likas, "Text document clustering using global term context vectors," *Knowl. Inf. Syst.*, vol. 31, pp. 455–474, 2012.

[11] A. Budanitsky and G. Hirst, "Evaluating wordnet-based measures of lexical semantic relatedness," *Computational Linguistics*, pp. 13–47, 2006.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[13] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JASIS*, vol. 41, pp. 391–407, 1990.

[14] T. Hofmann, "Probabilistic latent semantic indexing," in *SIGIR*, 1999, pp. 50–57.

[15] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in neural information processing systems*, 2008, pp. 121–128.

[16] A. G. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani, "Algorithmic detection of semantic similarity," in *Proceedings of the 14th international conference on WWW*, 2005, pp. 107–116.

[17] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, pp. 17–30, 1989.

[18] J. J. Castillo, "A wordnet-based semantic approach to textual entailment and cross-lingual textual entailment," *International Journal of Machine Learning and Cybernetics*, vol. 2, pp. 177–189, 2011.

[19] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis." in *IJCAI*, vol. 7, 2007, pp. 1606–1611.

[20] M. Strube and S. P. Ponzetto, "Wikirelate! computing semantic relatedness using wikipedia," in *AAAI*, vol. 6, 2006, pp. 1419–1424.

[21] I. Witten and D. Milne, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," in *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, 2008, pp. 25–30.

[22] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, pp. 613–620, 1975.

[23] J. L. Boyd-Graber and D. M. Blei, "Syntactic topic models," in *Advances in neural information processing systems*, 2009, pp. 185–192.

[24] X. Chen, M. Zhou, and L. Carin, "The contextual focused topic model," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 96–104.

[25] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 113–120.

[26] T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda, "Online multiscale dynamic topic models," in *Proceedings of the 16th ACM SIGKDD*, 2010, pp. 663–672.

[27] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *Proceedings of the 18th ACM CIKM*, 2009, pp. 375–384.

[28] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic sentiment mixture: Modeling facets and opinions in weblogs," in *Proceedings of the 16th International Conference on WWW*, 2007, pp. 171–180.

[29] H. Zhuge, *The Knowledge Grid: Toward Cyber-Physical Society*, 2012.

[30] O. Lassila, R. R. Swick *et al.*, "Resource description framework (rdf) model and syntax specification," 1998.

[31] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, "A study on similarity and relatedness using distributional and wordnet-based approaches," in *NAACL*, 2009, pp. 19–27.

[32] C. Fellbaum, *WordNet*, 1998.

[33] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, "A word at a time: Computing word relatedness using temporal semantic analysis," in *WWW*, 2011, pp. 337–346.

[34] G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren, "Large-scale learning of word relatedness with constraints," in *the 18th ACM SIGKDD*, 2012, pp. 1406–1414.

[35] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles, "Collabseer: A search engine for collaboration discovery," in *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, 2011, pp. 231–240.

[36] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the american statistical association*, vol. 101, no. 476, 2006.