Faculty of Engineering and Information Technology
University of Technology, Sydney

Representing Semantic Relatedness

A thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Analytics by Research

by

Qianqian Chen

April 2016

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisors Prof. Longbing Cao, Dr. Guandong Xu and Dr. Wei Liu for the continuous support of my study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis.

I also would like to appreciate my co-author Liang Hu for providing me with continuous support throughout my study and research. Without his professional guidance and persistent help, this thesis would not have been possible.

In addition, I thank my fellow lab mates in Advanced Analystics Institute: Jia Xu and Jingyu Shao for the stimulating discussions, for the hard days we were working together, and for all the fun we have had in the last three years.

Last but not the least, I would like to thank my family, for their unconditional support, both financially and emotionally throughout my whole study.

Qianqian Chen
December 2015 @ UTS

Contents

List of Figures

List of Tables

List of Publications

Papers Published

- **Qianqian Chen**, Liang Hu, Jia Xu, Wei Liu, Longbing Cao (2015), Document Similarity Analysis via Involving both Explicit and Implicit Semantic Couplings. *in* 'Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analysis (**DSAA2015**)'.

Papers to be Submitted/Under Review

- **Qianqian Chen**, Liang Hu, Jia Xu, Longbing Cao (2015) Document Representation Using Hierarchical Tree Augmented Naive Bayes, to be submitted.

- **Qianqian Chen**, Liang Hu, Jia Xu, Longbing Cao (2015) A Novel Document Similarity Representation: Capturing Explicit and Implicit Content Couplings, to be submitted.

Abstract

To do text mining, the first question we must address is how to represent documents. The way a document is organised reflects certain explicit and implicit semantic and syntactical coupling relationships which are embedded in its contents. The effective capturing of such content couplings is thereby crucial for a genuine understanding of text representations. It has also led to the recent interest in document similarity analysis, including semantic relatedness, content coverage, word networking, and term-term couplings.

Document similarity analysis has become increasingly relevant since roughly 80% of big data is unstructured. Accordingly, semantic relatedness has generated much interest owing to its ability to extract coupling relationships between terms (words or phrases). Existing work has focused more on explicit couplings and this is reflected in the models that have been built.

In order to address the research limitations and challenges associated with document similarity analysis, this thesis proposes a semantic coupling similarity measure and the hierarchical tree learning model to fully enrich the semantics within terms and documents, and represent documents based on the comprehensive couplings of term pairs. In contrast to previous work, the models proposed can deal with unstructured data and terms that are coupled for various reasons, thereby addressing natural language ambiguity problems.

Chapter 3 explores the semantic couplings of pairwise terms by involving three types of coupling relationships: (1) 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; and finally, (3) semantic coupling similarity, which effectively combine the intra- and inter-term couplings. The corresponding term semantic similarity measures are then defined to capture such couplings for the purposes of analysing term and document similarity. This approach effectively addresses both synonymy (many words per sense) and polysemy (many senses per word) in a graphical representation, two areas that have up until now been overlooked by previous models.

Chapter 4 constructs a hierarchical tree-like structure to extract highly correlated terms in a layerwise fashion and to prune weak correlations in order to maintain efficiency. In keeping with the hierarchical tree-like structure, a hierarchical tree learning method is proposed. The main contributions of our work lie in three areas: (1) the hierarchical tree-like structure featuring hierarchical feature extraction and correlation computation procedures whereby highly correlated terms are merged into sets, and these are associated with more complete semantic information; (2) each layer is a maximal weighted spanning tree to prune weak feature correlations; (3) the hierarchical tree-like structure can be applied to both supervised and unsupervised learning approaches. In this thesis, the tree is associated with Tree Augmented Naive Bayes (TAN) as the Hierarchical Tree Augmented Naive Bayes (HTAN).

All of these models can be applied in the text mining tasks, including document clustering and text classification. The performance of the semantic coupling similarity measure is compared with typical document representation models on various benchmark data sets in terms of document clustering and classification evaluation metrics. These models provide insightful knowledge to organise and retrieve documents.

Chapter 1

Introduction

1.1 Background

1.1.1 Non-iidness Learning

Most of the classic theoretical systems and tools in statistics, data mining and machine learning are built on the fundamental assumption of *iidness*, which assumes the independence and identical distribution of underlying objects, attributes and/or values. This works well in simple applications and abstract problems with weakened and avoidable relations and heterogeneity, and serves as the foundation of classic analytics, mining and learning theories, algorithms, systems and tools. However, complex behavioral and social problems often exhibit strong couplings and heterogeneity between values, attributes and objects (i.e., *non-iidness*), which cannot be abstracted or weakened to the extent of satisfying the iidness assumption (Cao 2013).

Coupling refers to any relationship (for instance, co-occurrence, neighborhood, dependency, linkage, correlation, or causality) between two or more aspects, such as object, object class, object property (variable), process, fact and state of affairs, or other types of entities or properties (such as learners and learned results) appearing or produced prior to, during and after a target process (such as a learning task). In a learning system, couplings may exist within and/or between aspects, such as entity (objects, object class, instance, or group/community) and its/their properties (variables), context (environment) and its constraints, interactions (exchange of information, material or energy) between entities or between the entity and its/their environment, learning objectives (targets, such as risk level or fraud), the corresponding learning methods (models, algorithms or systems) and resultant outcomes (such as patterns or clusters) (Cao 2014).

Such strong couplings and heterogeneity are particularly embodied in complex behavioural/social systems, for instance, (Wang, Cao, Wang, Li, Wei & Ou 2011), (Cao, Ou & Yu 2012) and (Yu, Wang, Gao, Cao & Chen 2013) apply non-iidness learning to consider the inter-relations of users that are influenced by other users in terms of various aspects of social media, rather than just expanding the traditional iidness-based algorithms.

In classic document analysis, typical algorithms such as the Bag-of-words model take the term independence assumption and ignore the semantics between terms. This, in turn, leads to low learning performance. Typical approaches also expand existing theories and algorithms in a bid to relax this assumption. Furthermore, they map texts into new feature space in order to capture more term relation. However, (Cheng, Miao, Wang & Cao 2013) proposes a new document clustering framework, which analyses semantic couplings between terms appearing within and between documents. This caters for both explicit and implicit semantics, incorporating the intra-term couplings between terms within a document, the inter-term couplings between terms from different documents, and the aggregative term coupling by combining intra-term and inter-term couplings.

Non-iidness learning techniques are general and can be widely used and expanded for analysing complex behavioural and social problems. It is still quite a new concept in the semantic representation area, and a large amount of potential research topics are yet to be developed. For example, how do we capture term pair semantics based on the influence of other ones, how do we manage natural language ambiguity in the process of non-iidness learning, and how do we consider the text structural information? These constitute fundamental and traditional research issues in terms of the challenges associated with document representation as well.

1.1.2 Representing Semantic

With the rapid development of the World Wide Web, an increasing amount of data relating to a myriad of different areas has been presented in the forms of documents on the Internet. However, this makes it difficult to extract relevant documents from the wealth of textual material in response to user queries. The problem can be that the information is misinterpreted due to natural language ambiguities or the information is imprecisely and vaguely defined by the user. This calls for the development of automatic methods for searching and organising text documents so that information of interest can be accessed quickly and the accuracy of matching between queries and documents can be achieved to a high degree. To conduct information retrieval tasks, such as document clustering, document classification, and document filtering, the first step lies in discerning the effectiveness of the approach to improve the precision of the representation of queries and documents, and the discriminability of a given document with respect to other documents.

In information retrieval, the content of a document may be represented as a collection of terms (Strzalkowski 1994). This allows for the design of a high-quality document which may serve as an efficient model to discover all components and reassemble them in a new feature space. It also means that document representation mines the semantic relatedness (similarity or distance) between terms (words or phrases) and breaks up every document into terms and their relations. From the traditional keyword-based representations to the currently wide-developed concept- and topic-based models, it has long been recognised that measuring the semantics of terms in a more efficient way is the top priority for researchers who seek to improve models.

Based on coupling learning, the terms, words and phrases in a document are organised in terms of certain coupling relationships, from semantic, syntactic/linguistic or even subjective perspectives (Gabrilovich & Markovitch 2007). Coupling refers to any relationships (for instance, co-occurrence, neighborhood, dependency, linkage, correlation, or causality) between two or more aspects (Cao 2013). There are intrinsic textual/linguistic complexity couplings (such as natural language ambiguation) and various couplings (such as co-occurrence) that drive the semantics between terms and documents. This makes it very challenging in relation to analysing the semantics in information retrieval and document analysis, e.g. document clustering, document classification, document query and document filtering. Consequently, a query can often hit a large number of documents but few of them are relevant. This calls for further research on semantic representations by deeply exploring the couplings within and between terms/documents and this is particularly important for accurate information queries and document processing.

The problem of document semantic similarity can be further decomposed to explore the coupling relatedness and similarity between the terms (words or phrases) which forms a document. This helps to build a feature space that consists of all the necessary terms with their semantics captured and embedded in a similarity (or distance) learning model. A document analysis algorithm can then be built to analyse the semantic similarity between documents by exploring the intrinsic term couplings and similarity (Cao 2013). For this, it is crucially important to measure intrinsic semantic relatedness which is fundamental for information retrieval and other related natural language processing applications, such as text summarisation, textual entailment, information extraction, etc. Document similarity analysis has also recently emerged as a promising and important topic particularly in relation to semantic relatedness (Strube & Ponzetto 2006, Gabrilovich & Markovitch 2007), content coverage (Holloway, Bozicevic & Börner 2007), word networking (Budanitsky & Hirst 2006, Agirre, Alfonseca, Hall, Kravalova, Paşca & Soroa 2009), term-term couplings (Cheng et al. 2013), and general problems including information retrieval (Billhardt, Borrajo & Maojo

2002, Hliaoutakis, Varelas, Voutsakis, Petrakis & Milios 2006), ontological engineering (Hawalah & Fasli 2011, Li, Wang, Zhu, Wang & Wu 2013), and document clustering/classification (Farahat & Kamel 2011, Kalogeratos & Likas 2012).

Recent literature on semantic representing measures can be roughly characterised into two categories: corpus-based and lexical resource-based measures. Concretely, semantics can be estimated using statistical means such as vector space models which compute the co-occurrence frequency pattern of terms and textual contexts across a corpus; and probabilities models which discover the distribution properties of each term over topics and the topic distribution over each document. Lexical resource-based measures capture the semantic relatedness between terms or concepts by using ontologies to define the distance between them, and most of these methods rely on pre-existing knowledge resources that are represented by a directed or undirected graph consisting of vertices, for example, semantic networks and taxonomies. Semantic representing measures have attracted increasingly research interest in information retrieval and other related natural language processing applications, such as text summarisation, textual entailment, information extraction, etc.

1.2 Research Issues

Textual information forms probably the major proportion of big online data. With the rapid development of the Internet and Internet-based business, it is critical to understand the semantic similarity between terms (queries), text or documents by directly exploring their coupling relationships (Cao 2013, Cheng et al. 2013) as well as complex techniques such as natural language processing. We also need to fix the intrinsic textual/linguistic complexity (such as natural language ambiguation). This has become a promising and popular research task in recent years particularly in the areas of information retrieval (Billhardt et al. 2002, Strzalkowski 1994, Wong, Ziarko & Wong

1985), ontological engineering (Hawalah & Fasli 2011, Li et al. 2013, Sánchez, Batet & Isern 2011), and document analysis (Cheng et al. 2013, Kalogeratos & Likas 2012, Salton, Wong & Yang 1975).

Cao (Cao 2013) presented a high-level picture of the *non-iidness* (independence and identical distribution) learning problem for dealing with strong couplings and heterogeneity in complex behavioral and social applications, and this cannot be abstracted or weakened to the extent of satisfying the iidness assumption. To handle the independency assumption of classic document representations, Cheng et al. (Cheng et al. 2013) applied the non-iidness theory to document analysis by exploring the intrinsic term couplings and similarity. The problem of term coupled semantic similarity was further decomposed to explore the explicit and implicit relationships between terms.

The limitations of (Cheng et al. 2013) include: (1) it considers term coupling by combining intra-term and inter-term couplings as the explicit and implicit relation of terms respectively, but fails to give a clear distinction of intra and inter, which, in turn, results in a repeated calculation of the coupled relation; (2) it is not capable of avoiding synonymy and polysemy problems; (3) it is time-consuming in terms of calculating pairwise term couplings.

Based on the aforementioned current research limitations, we focus on the following research issues:

1.2.1 Capturing Coupled Pairwise Term Semantics

We explore the semantics of pairwise terms by involving three types of coupling relationships: (1) the intra-term couplings, reflecting the explicit relation within term pairs that is represented by the relation strength over probabilistic distribution of terms across document collection; (2) the inter-term 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; and (3) coupled term pair similarity, effectively combining the intra- and inter-term couplings. The corresponding term semantic similarity measures are then defined to capture the semantics for analysing term and document similarity.

This approach effectively addresses the limitation of semantic coupling in a graphical representation, which is overlooked by previous models. The details are as follows:

- The intra-term coupling is calculated from the 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, and this takes full advantage of the interactions with other terms in a document set.

- Our inter-term coupling method is based on weighted paths with limited lengths. On one hand, it distinguishes strong link terms from weak link terms, e.g. the strong link terms which are visited frequently on all possible paths occupy higher weights. On the other hand, it emphasises that less link terms build closer relatedness, and only strong link terms are reserved in order to improve the efficiency of the calculation.

1.2.2 Measuring High Order Semantics by Hierarchical Tree Learning

To further explore term correlation and construct a more semantic-associated document representation model, representing documents as terms and pairwise correlations as completely as possible is still not enough to fully capture the meaning of documents. The reasons lie in (1) the term independence assumption is not necessarily correct in practice, but if we relax the independence assumption or even consider each pairwise term dependence as a complete graph it may suffer from high computational cost; (2) representing documents as terms and pairwise correlations still fails to consider the dependence of term pairs, because it is a term-level document representation with the feature selected individually, and it only considers the meaning of documents based on the mutual information of each pairwise term share, regardless of the real contents and themes that the documents express. Furthermore, the higher order of semantics are overlooked.

We address the above issues by constructing a hierarchical tree-like structure to extract highly correlated terms in a layerwise manner and to prune weak correlations to maintain efficiency. This hierarchical feature extraction structure guarantees that with higher layers, the features are reflected as term sets, with more comprehensive semantic information involved. It is thereby not a simple term-level document representation, but a layer-increased term set-level document representation, and these contain the correlations and similarities between terms and term sets which form documents.

The hierarchical tree is able to capture the dependence between terms, term pairs, or even term sets. The higher the order of the tree, the more semantics it carries. The hierarchical tree is closer to the human understanding of texts, grouping them into different classes by comprehending the topics and contents of the texts.

1.2.3 Managing Natural Language Ambiguities

Ambiguity can be referred to as the ability of having more than one meaning or being understood in more than one way. Natural languages are ambiguous, so computers are not able to understand language the way that people do. The ambiguity problems that we are concerned with specifically refer to *lexical semantic ambiguity*, the type of lexical ambiguity which occurs when a single word is associated with multiple senses, or the semantic relation that holds between two words that can (in a given context) express the same meaning.

The coupling method 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 the corpus and all possible paths respectively, accordingly the term-pair occurrence frequency patterns appear across a document set or all possible paths instead of each single term, and the semantic meaning for every term pair is richer than individual terms;

(2) coupling similarity is built on term distributions. For terms that are semantically similar, their distributions are similar, and the coupling similarity 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, coupling similarity is surely weaker than similar term pairs.

It is also reasonable to adapt hierarchical tree learning into word sense disambiguation. In a tree structure, features are extracted and composed layer by layer, with the higher level of the tree, the features are represented as bigger term sets associated with more comprehensive-connected terms based on their similarities. The term sets contain more integrated and precise semantic senses compared to single terms, which make it possible to avoid misunderstanding of the true meaning.

1.3 Research Contributions

- The proposal of an effective term pair similarity to capture the comprehensive couplings across documents. The similarity combines the intra- and inter-term couplings via statistical and topological strategies (Chapter 3);

- Application of the coupled term pair similarity measure to analyse documents by capturing the semantic related documents. (Chapter 3);

- The proposal of a hierarchical tree learning model to capture high order semantics (Chapter 4);

- The implementation of a hierarchical tree with TAN as the hierarchical tree augmented naive Bayes (HTAN) to text classification (Chapter 4);

- Analysis of both synonymy (many words per sense) and polysemy (many senses per word) (Chapter 3 & 4).

1.4 Thesis Structure

The thesis is structured as follows:

Chapter 2 provides a literature review of the definition of the term semantic, and various document representation models are introduced by three directions based on the different ways to capture the semantics. Furthermore, applications of document representation modeling in the field of information retrieval is reviewed.

Chapter 3 presents a term pair semantic coupling similarity (SCS) measure examined from both statistical and topological perspectives. A graphical depiction associated with statistical approaches is employed to fully enrich the semantic relation of every term pair, and this considers not only the occurrence frequency pattern of the terms themselves, but also the influence of link term sets, so that documents are semantically represented in the best possible way. Additionally, by adapting a path-length based method, we show that SCS can cater for both synonymy and polysemy, and it consistently outperforms baseline methods on most real data sets. The approach has been evaluated and compared with related work when applied to document clustering.

Chapter 4 proposes a multi-level tree structure to address the problem in traditional Bayesian classification methods. The hierarchical tree learning model is (1) constructed as a hierarchical tree-like structure, with hierarchical feature extraction and correlation procedures to capture implicit semantics; (2) at the top layer of the hierarchical tree, texts are represented as term sets which contain highly correlated terms, and the correlation between term sets. In addition, the tree can be widely adapted for supervised and unsupervised learning; and (3) the tree is associated with TAN as the Hierarchical Tree Augmented Naive Bayes (HTAN).

Chapter 5 concludes the thesis and outlines the scope for future work.

Figure 1.1 shows the research profile of this thesis.

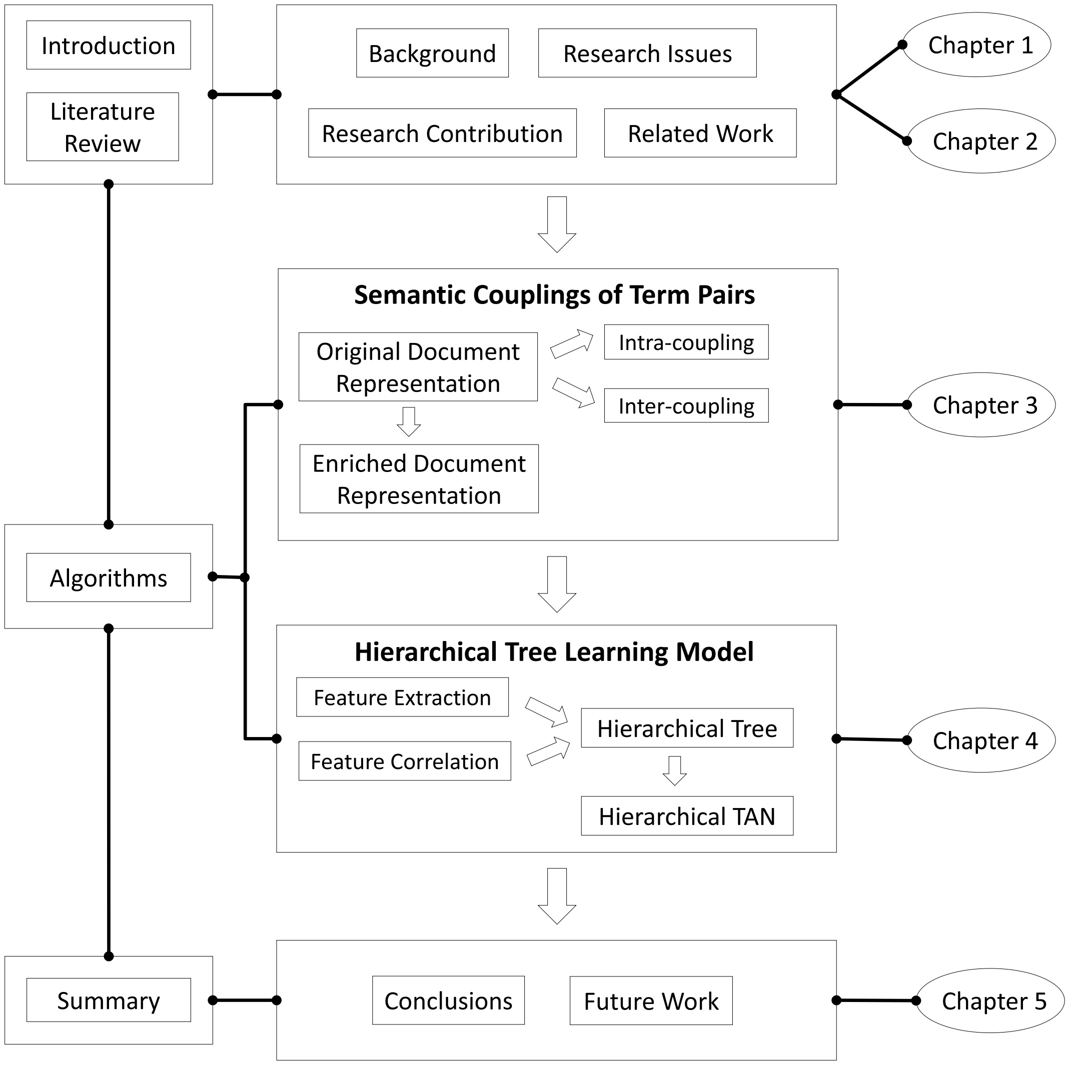


Figure 1.1: The profile of work in this thesis

Chapter 2

Literature Review

The content of a document may be represented as a collection of terms: words, stems, phrases, or other units derived or inferred from the text of the document. In order to reduce the complexity of a document and make it easier to handle, the document should be transformed from the full text version to a document representation R which describes the contents of the document, that is to say, for any text items D_1 and D_2, $R(D_1) = R(D_2)$ iff $meaning(D_1) = meaning(D_2)$. The effectiveness of any such representation is directly related to the accuracy with which a set of terms represents the content of a document, as well as how well it discriminates a given document with respect to other documents.

Building a high-quality document representation model is a challenging task due to the complexity of natural language. It is expected that the representation of documents should reflect the knowledge that is meant to be conveyed by the documents. From traditional word-based document representation, to today's widely-accepted document semantic representation, a number of methods have been developed to exploit the semantic similarity and relatedness between terms for the purposes of enhancing the efficiency of document representation.

This chapter reviews the related work, taking into account the definition of semantics, typical semantic representations, including the Bag-of-Words model, the vector space model families, the topic models, and lexical resource-based models. The definition of semantics and its associated families are introduced in Section 1, and then various semantic representation models are reviewed in Section 2.

2.1 Semantics

In linguistics, *semantics* is devoted to the study of *meaning*, inherent at the levels of words, phrases, sentences, and larger units of discourse (termed texts, or narratives).

One of the basic studies of semantics concerns the examination of the representation of semantics, and the study of semantic relatedness between different linguistic units and compounds: homonymy, synonymy, antonymy, hypernymy, hyponymy, meronymy, metonymy, holonymy, paronyms.

Semantic relatedness, which is different from *semantic similarity* or *semantic distance*, is a more general concept that subsumes many different kind of specific relations, including meronymy, antonymy, functional association, and others (Budanitsky & Hirst 2006). Prior work on the semantic relatedness of words pursued two main directions, using purely statistical techniques or using repositories of human knowledge (Halawi, Dror, Gabrilovich & Koren 2012).

Machine learning methods learn word semantic relatedness from text corpora. The semantic analysis of a corpus is the task of using statistical techniques to build structures that approximate concepts from a large set of documents. It generally does not involve a prior semantic understanding of the documents. While approaches measuring semantic relatedness based on lexical resources, regard the resources as networks or graphs, and then apply various measures of relatedness to the properties of paths in the networks or graphs.

2.2 Corpus-based Semantic Representation

The lexical semantic system is an important component of human language and natural language processing. Recent efforts in terms of measuring semantics can be roughly categorised as: corpus-based measures and lexical resource-based measures.

More specifically, semantic relations estimated by corpus-based statistical means seek high-dimensional vectors to model term meanings. These are derived from the co-occurrence of terms in the text corpus. Corresponding approaches such as the vector space models and its extensions compute the co-occurrence frequency patterns of terms and textual contexts across a corpus. Probabilistic models discover the distribution properties of each term over topics and the topic distribution over each document. By contrast, lexical resource-based approaches model semantic knowledge which relies on pre-existing knowledge resources, like hand-constructed networks or trees of interconnected word senses. The networks or trees do not provide a term-pair similarity metric, but various metrics based on these structures have been developed for this purpose (Rohde, Gonnerman & Plaut 2004).

A specific review of typical document representation models is provided below. This is divided into three sections, term-based models, which consist of vector space models; topic-based models, which emphasise probabilistic latent variable models; and topological models, which model semantic knowledge based on topological similarity on lexical ontology.

2.2.1 Term-based Semantic Representation

Term-based document representation models use term, which is the smallest unit of a document, as the key feature. On the basis of the development of term-based representation methods, it is important to develop algorithms that capture term correlations as completely as possible, and construct the corresponding document representation models based on terms and their pairwise correlations. In the following sections, I will chronologically present several typical term-based approaches that work on mining term correlations.

Bag-of-words Model

Early research on term-based methods usually build on bag-of-words model, which is a simplified representation used in natural language processing (NLP) and information retrieval (IR). It is an orderless document representation model, which treats all the words in a text (such as a sentence or a document) as index terms bounded with weights to reflect their importance. A document is represented as a bag (multiset) of its words, disregarding the order, structure, meaning, grammar, etc. of the words, only keeps multiplicity of words from the document. The bag-of-words model gains the limitation of the term independence assumption, ignores the semantics between terms accordingly, which leads to a great loss of text semantic information. For example, the two sentences

1. John is eating the apple, standing beside the tree.
2. The apple tree stands beside Johns house.

have the same set of content words (except house), but mean entirely different things.

On the other hand, the sentences

1. John is an intelligent boy.
2. John is a brilliant lad.

mean almost the same thing.

This is why it causes problems if we do not consider the senses of the words or their mutual semantic relations.

The bag-of-words model is summarized as:

- Orderless document representation model, keeps frequencies of words from a dictionary.

- Treat all the words in a document as index terms.
- Assign a weight to each term based on importance, in simplest case, weight is the presence or absence of words.
- Disregard order, structure, meaning, grammar, etc. of the words.
- Term occurrence is independent, document relevance is independent, totally ignore the semantic relation between them.

Vector Space Model

The Vector Space Model (Salton et al. 1975) is one of the most famous traditional models for representing text document and is regarded as the basics of various extended models. The basic vector space models are built on bag-of-words model, documents are represented as vectors, each dimension corresponds to a separate term with weight. The definition of term depends on the application. Typically terms are single words, keywords, or longer phrases. If words are chosen to be the terms, the dimensionality of the vector is the number of words in the vocabulary (the number of distinct words occurring in the corpus). The vector space models procedure can be divided in to three stages.

(1) Document Indexing

It is obvious that many of the words in a document do not describe the content, words like *the*, *is*. By using automatic document indexing those non significant words (function words) are removed from the document vector, so the document will only be represented by content bearing words (Chowdhury 2010). This indexing can be based on term frequency, where terms that have both high and low frequency within a document are considered to be function words. In practice, the use of a stop list which holds common words to remove high frequency words (stop words), are filtered out after processing of natural language data (text), which makes the indexing method language dependent.

In general, $40-50\%$ of the total number of words in a document are removed with the help of a stop list (Chowdhury 2010).

(2) Term Weighting

There are various weighting schemes to discriminate one document from the other. The term weighting for vector space models has entirely been based on single term statistics. A common weighting scheme for terms within a document is to use the frequency of occurrence, which is called *term frequency - inverse document frequency*, short for *tf-idf.* The values of the vector elements for a document are calculated as a combination of the statistics of the term frequency (*tf*) and the inverse document frequency (*idf*). In general, *tf* stands for the number of times a term occurs in document, and *idf*, indicates the ability of a term to distinguish a document.

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, this is to conduct the statistical analysis of term co-occurrence patterns by assuming that terms are regarded relational if they co-occur in the same document, and more frequently they co-occur, the stronger relations they have, and more semantics they share. The document frequency $df(t)$ is the number of documents in which t occurs at least once (Jing, Huang & Shi 2002), and the inverse document frequency *idf* can be calculated as

$$idf(t, D) = log(\frac{|D|}{df(t)}) \tag{2.1}$$

where $|D|$ is the total number of documents. The inverse document frequency, assume that the importance of a term is proportional with the number of document the term appears in. *idf* is low if t occurs in many documents and will be high if it occurs in few documents.

Then *tf-idf* is computed as:

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \tag{2.2}$$

A high weight in *tf-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-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.

Generally, this approach of document mapping can be expressed as a term sequence defined by weighting scheme,

$$\widetilde{\Phi}_{vsm} : d \to d^{'} = (tfidf_1, tfidf_2, \cdots, tfidf_n) \tag{2.3}$$

However, the *tf-idf* based methods have two main limitations. One is that they place undue emphasis on the documents where terms co-occur; the other is that *tf-idf* based on one single term may lead to synonymy and polysemy, since the semantic meaning of a term in different documents can be various.

(3) Similarity Coefficients

The similarity in vector space model is determined by using associative coefficients based on the inner product of the document vectors, where word overlap indicates similarity. The inner product is usually normalized. The most popular similarity measure is the *cosine similarity* coefficient.

Cosine similarity is widely used in information retrieval and text mining, as shown in Figure 2.1, it measures the cosine of the angle between document vectors, which is regarded as the similarity of two documents in terms of their subject matter. In information retrieval, vector space models capture the relevance of documents and queries, by comparing the cosine similarity of query vector and document vector.

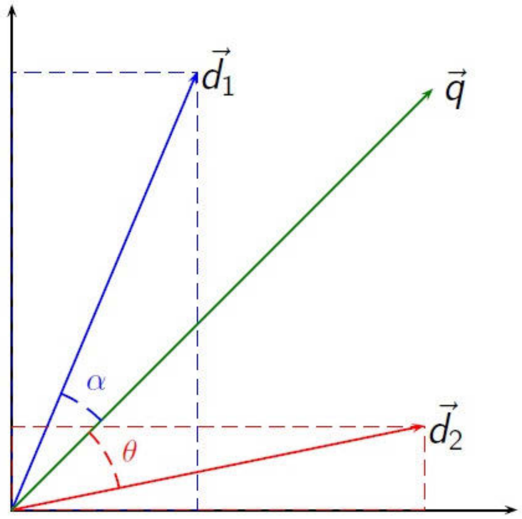


Figure 2.1: Cosine similarity in vector space model

The cosine similarity of documents d_i and d_j is represented as

$$cos\theta = \frac{\sum_{i,j=1}^{N} d_i \cdot d_j}{\sqrt{\sum_{i,j=1}^{N} {d_i}^2} \cdot \sqrt{\sum_{i,j=1}^{N} {d_j}^2}} \tag{2.4}$$

where N is the total number of documents. its outcome is neatly bounded in $[0, 1]$, due to cosine similarity is used in positive space particularly. A larger value indicates more similar distributions of d_i and d_j, it leads to a stronger document similarities.

There are also other statistical similarity or distance measures to compare the similarity of two samples, for example, *Jaccard index*, *Dice coefficient*, *Hamming distance* and so on so forth.

Generalized Vector Space Model

On the basis of vector space model, a diversity of extended models have been proposed, like the Generalized Vector Space Model (GVSM) (Wong et al. 1985).

Vector space models treat the terms as a set of orthogonal vectors, however, terms are in fact correlated, it is not possible to characterize the vector space completely by representing documents as term occurrence frequency only, without a explicit representation of term vectors that associated with term correlations.

GVSM is an analysis of the problems that the pairwise orthogonality assumption of the vector space model creates, it is not necessary in GVSM to assume that either the document or the term vectors have to be orthogonal.

Specifically, GVSM represents documents in the document similarity space, assumes that the correlation between a pair of index terms depends on the number of documents in which this pair of terms appear together, which means two terms are similar if they frequently co-occur in the same document. In GVSM, they consider a new space, where each term vector $\vec{t_i}$ was explicitly defined in a 2^n-dimensional cartesian space based on the notion of Boolean algebra, which is a linear combination 2^n *minterms*. The set of minterms $\{m\}_{2^n}$ can be represented explicitly as the set of orthonormal basic vectors,

$$\begin{aligned} \vec{m}_1 &= (1, 0, 0, \cdots, 0) \\ \vec{m}_2 &= (0, 1, 0, \cdots, 0) \\ \vec{m}_3 &= (0, 0, 1, \cdots, 0) \\ &\vdots \\ \vec{m}_{2^n} &= (0, 0, 0, \cdots, 1) \end{aligned} \tag{2.5}$$

Then each term t_i is defined by a unique disjunctive canonical representation, which is the sum of minterms,

$$\vec{t_i} = \sum_{k=1}^{r} \vec{m}_{ik} \tag{2.6}$$

To be more realistic, terms are assigned to documents as weights ($0 \leq w \leq 1$) instead of strictly Boolean algebra. In this case, the normalized term vector $\vec{t_i}$ is transformed to

$$\vec{t_i} = \frac{1}{N_i} \sum_{k=1}^{r} c(t_i) \vec{m}_{ik} \tag{2.7}$$

where

$$N_i^2 = \sum_{k=1}^{r} c^2(t_i) \tag{2.8}$$

and $c(t_i)$ denotes the sum of weights.

From now on, term vectors are explicitly defined by Equation 2.7, term correlations are also explicitly known by computing $\vec{t_i} \cdot \vec{t_j}$ between any pair of index terms. The mapping of document vector is defined as

$$\widetilde{\Phi}_{vsm} : d \rightarrow d^{'} = Dd \tag{2.9}$$

where D is the document-term matrix.

Then it uses similarity method (e.g. cosine) to capture the similarity between document vectors or document and query vectors on new space dimensions.

However, GVSM keeps the assumption that the term vectors are linearly independent, which still suffers the lack of relatedness of any pair of terms.

Context Vector Model

Context vector model (CVM) (Billhardt et al. 2002) is also developed to deal with the restrictive term independence assumption in vector space models. It incorporates *context vectors* into the vector space models to measure term dependencies, which stores terms semantic similarities to the other terms, and thus obtains semantically richer representations of documents.

To be more specific, CVM represents documents by a set of context vectors, which reflect the correlations or influences between terms, each value in the context vectors are not only determined by the occurrence frequency of the corresponding term itself, but also by other terms occurring in the document. This is the basic assumption of CVM, that terms are not independent of each other, they indicates the possibility of existence of concepts in that document, which correspond to other terms.

The basic algorithm consists three steps:

1. Build the term/document matrix,

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \end{array} \begin{array}{c} \begin{array}{cccc} d_1 & d_2 & \cdots & d_m \end{array} \\ \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{m1} \\ w_{12} & w_{22} & \cdots & w_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1n} & w_{2n} & \cdots & w_{mn} \end{pmatrix} \end{array}$$

   m is the number of documents, n is the number of terms, and each element w_{ij} in the matrix indicates the occurrence frequency of term t_j in document d_i. the document vector $\vec{d_i} = (w_{i1}, w_{i2}, \cdots, w_{in})^T$ is used for calculating document context vectors.

2. Build the term correlation matrix, represent documents as a set of 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. The term correlation matrix can be represented by an $n \times n$ matrix T as follows,

$$T = \begin{pmatrix} c_{11} & c_{21} & \cdots & c_{n1} \\ c_{12} & c_{22} & \cdots & c_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{1n} & c_{2n} & \cdots & c_{nn} \end{pmatrix}$$

   where the ith column represents the term context vector $\vec{t_i} = (c_{i1}, c_{i2}, \cdots, c_{in})^T$ for term t_i in the n-dimensional term space, each c_{ik} represents the influence of term t_k on term t_i.

   To estimate the elements in matrix T, which is the influence of term t_k on the semantic description of term t_i, term co-occurrence frequency is employed, that is, the frequency with which t_k and t_i co-occur in the same textual units across the whole collection.

3. Transform each original document vector into document context vector.

Once the term correlation matrix T has been generated, the initial document vector $\vec{d_i} = (w_{i1}, w_{i2}, \cdots, w_{in})^T$ is transformed into a context vector $\vec{d_i'} = (c_{i1}, c_{i2}, \cdots, c_{in})^T$, where

$$\vec{d_i'} = \frac{\sum\limits_{j=1}^{n} w_{ij} \frac{\vec{t_j}}{|\vec{t_j}|}}{\sum\limits_{j=1}^{n} w_{ij}} \tag{2.10}$$

$\vec{t_j}$ is the context vector of term t_j and $|\vec{t_j}|$ is the length of vector $\vec{t_j}$.

Generally, this approach of document mapping can be expressed as

$$\widetilde{\Phi}_{cvm} : d \rightarrow d' = \sum_{j=1}^{n} w_{ij} \cdot \vec{t_j} \tag{2.11}$$

The generated document context vectors $\vec{d_i'} = (c_{i1}, c_{i2}, \cdots, c_{in})^T$ correspond to the centroid of the term context vectors of all terms in the document. The value of concept c_k in the document context vector for document i is the average of the influences of term t_k on all terms occurring in d_i. If all terms are represented only by themselves, the context vectors for terms are pairwise orthogonal, CVM will behave in the same way as traditional vector space models.

After indexing process, standard cosine similarity measure also can be adapted in document similarity measurement or document retrieval. CVM uses context vector to store the similarities of one term with other terms, incorporates term dependencies based on vector space models, the document representation is further semantically enriched, so that the document similarities is based on the semantic-matching.

Global Term Context Vector Model

The CVM method considers that the term context is computed based on term co-occurrence frequency at the document level, but does not take into account the sequential nature of text and thus ignores the local distance of terms when computing term context. To address this problem, the global term context vector model (GTCV) (Kalogeratos & Likas 2012) is proposed to utilize local contextual information, in addition, extends CVM by considering term context at three level: (1) the local term context vector, (2) the document term context vector and (3) the global term context vector. The intuition of GTCV is to capture the local term context from term sequence based on the location that terms appear around the sequence, and then to build global term context representation by averaging the local contextual information at the document and corpus level.

More specifically, local term context vector (*ltcv*) is a histogram associated with the occurrence of term $d^{seq}(l)$ at location l in sequence d^{seq}, which is a modified $lowbow(d^{seq}, l)$ (Lebanon, Mao & Dillon 2007) probability vector that represents contextual information around location l. Then the document term context vector (*dtcv*) is defined as a probability vector that summarizes the context of a specific term at the document level by averaging the *ltcv* histograms corresponding to the occurrences of this term in the document. The *dtcv* of term v for document i is computed as

$$dtcv(d_i^{seq}, v) = \frac{1}{no_{v,i}} \sum_{j=1}^{nov,i} ltcv((d_i^{seq}, l_{v,i}(j)) \tag{2.12}$$

where $l_{v,i}(j)$ is an integer value in $[1, \cdots, T_i]$ denoting the location of the j-th occurrence of term v in d_i^{seq}, $no_{v,i}$ is the number of times that v appears in the term sequence d_i^{seq}.

Next, the global term context vector (*gtcv*) represents overall contextual information for all terms in the corpus of all N term sequence (documents).

$$gtcv(v) = h_{gtcv(v)} \left(\sum_{i=1}^{N} tf_{i,v} dtcv(d_i^{seq}, v) \right) \tag{2.13}$$

where $tf_{i,v}$ is the term frequency in ith document, the coefficient $h_{gtcv(v)}$ nomalizes $gtcv(v)$ with respect to the Euclidean norm.

Finally, the semantic matrix S_{gtcv} is built where each row is the $gtcv(v)$ vector of the corresponding term v.

The new document representation using the proposed GTCV approach is the cosine similarity of the global term context vector $gtcv(v)$ and the bag-of-words representation d,

$$\widetilde{\Phi}_{gtcv} : d \rightarrow d^{'} = S_{gtcv}d \tag{2.14}$$

and the product $S_{gtcv}^{T}S_{gtcv}$ calculates the pairwise term similarity based on the distribution of term weights in their respective global term context vectors.

GTCV incorporate context vectors into vector space models to measure the term dependency. This method is operated in four steps: (i) considers local contextual information for each term occurrence in the term sequences of documents based on the bag-of-words; (ii) organises the local context vectors for the occurrences of terms to define the global context vectors; (iii) constitutes the semantic matrix by using the global context of all terms; (iv) uses the semantic matrix to further map the traditional vector space document representations into a new feature space which is semantically smoothed and richer. The semantic relation for vocabulary terms can be achieved from the total contextual information across the whole document collection.

Coupled Term-Term Relation Model

An interesting effort made by Cheng et al. (Cheng et al. 2013) in coupled term-term relation model (CRM) is to capture the semantic relation of terms by considering both intra- and inter-term relations based on Non-iidness learning (Cao 2013).

Non-iidness learning deals with strong couplings and heterogeneity in complex behavioral and social applications, which cannot be abstracted or weakened to the extent of satisfying the iidness assumption. Most of existing methods are proposed on the basis of the iidness assumption that objects, attributes and values are independent and identically distributed. It works well in simple applications and abstract problems with weakened and avoidable relations and heterogeneity (Cao 2013). In text mining, it is applied as bag-of-words model which is served as the foundation of classic document representation learning theories, algorithms and models.

However, complex behavioral and social applications often exhibit strong coupling relations between attributes, which are beyond the usual dependency relation. Corresponding reflection in document representation models is that classic bag-of-words model and vector space models ignore the semantic relations between terms or only consider the explicit relations, failing to capture the complete semantics between terms.

Motivated by this, CRM is proposed to further exploits semantic relation of terms by considering both intra-relation (explicit) and inter-relation (implicit). Firstly, the intra-relation, also the explicit semantic relation between terms can be captured by their co-occurrence frequency across all documents, they adapt the popular co-occurrence measure, the *Jaccard distance* and *tf-idf* weighting scheme to compute the similarity of two terms. The intra-relation $IaR(t_i, t_j)$ between term t_i and t_j is,

$$IaR(t_i, t_j) = \frac{CoR(t_i, t_j)}{\sum\limits_{i=1, i \neq j}^{n} CoR(t_i, t_j)} \tag{2.15}$$

where

$$CoR(t_i, t_j) = \frac{1}{|H|} \sum_{x \in H} \frac{w_{xi} w_{xj}}{w_{xi} + w_{xj} - w_{xi} w_{xj}} \tag{2.16}$$

where w_{xi} and w_{xj} represent the *tf-idf* weights of t_i and t_j in document d_x, respectively; and $|H|$ denotes the number of elements in $H = \{x | (w_{xi} \neq 0) \vee (w_{xj} \neq 0)\}$.

On the other hand, the inter-relation, known as the implicit relation, calculates terms relation where a pair of terms appear in document set but do not co-occur in the same document. Cheng et al. proposed a significant improvement to computer the inter-relation in terms by integrating the intra-relation over a pair of terms with a set of link terms that connecting those two terms. For instance, in Figure 2.2, terms t_i and t_j are intra-related with

t_k in respective docu
related by the link te
captured.

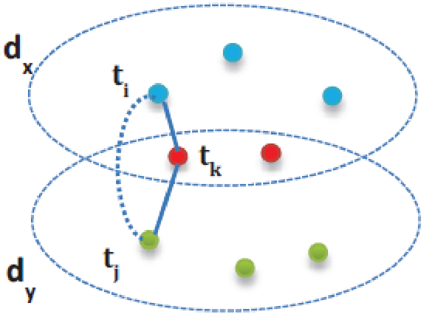


Figure 2.2: An example of the inter-relation between terms

Finally, linearly combine the intra-relation and inter-relation together to achieve a full CRM. The semantic relation matrix S_{crm} is then constructed to represent all coupled relations between each pair of terms, each document vector is transformed into new feature vector

$$\widetilde{\Phi}_{crm} : d \to d^{'} = S_{crm}d \tag{2.17}$$

Referring to results of experiments, CRM improves the performance of document clustering by building an enriched document representation to capture the complete semantic relation between terms. Especially, take inter-relation into account, which is overlooked by other measures. However, it still suffer from some limitations. CRM fails to avoid the negative effects of polysemy and synonymy of link terms, there are still exist possibilities that the link term has different meaning in different documents, which is known as polysemy. Furthermore, only one link term used is not enough to express the semantic relatedness completely.

Summary

Summarising the properties of the above-mentioned vector-based document representations, the traditional bag-of-words model holds the simplistic assumption that term features are considered to be independent of each other, which is very unrealistic in practice, since there exist semantic relations among terms which are totally ignored. To address the absence of the semantic relation, various extensions of the vector space model have been proposed which aim to capture the semantics among terms as completely as possible.

- The vector space model assumes that terms are regarded to be relational if they co-occur in the same documents and the semantics are calculated based on the similarity of the co-occurring term features.
- The generalized vector space model represents term vectors explicitly in a 2^n-dimensional vector space based on the notion of Boolean algebra and the semantic relation is captured by using the term co-occurrence pattern.
- The context vector model represents each term by introducing a term context vector that stores its similarities (semantic information) with the other terms. The similarity between terms is based on a document-wise term co-occurrence frequency.
- The global term context vector model utilizes local contextual information to construct the local term context vector, then summarises the local term context vectors of a particular term by way of the global context vector, which constitutes the semantic matrix.
- The coupled term-term relation model captures the semantic relation between terms, which considers both the co-occurrence patten of terms (intra-relation) and the dependency of terms via linkages (inter-relation).

The vector space model and its extensions can be regarded as *semantic smoothing* approaches (Kalogeratos & Likas 2012) which consider term correlations by redistributing the term weights of a vector model or mapping the features onto a new space.

The main differences among various semantic smoothing approaches is related to the fact that more and more semantics are involved. For more advanced methods, like CVM, term contextual information is introduced to store the similarities between terms, GTCV further considers term location information, and CRM captures the implicit relation between terms via the interaction with link terms.

In the next section, topic models will be reviewed which propose alternative ways to capture semantics.

2.2.2 Topic-based Semantic Representation

With the prevalence of statistical inference in recent years, more and more conventional information retrieval problems try to seek better solutions using machine learning methods. Topic models have employed machine learning methods to map documents into new feature space. The size of such document vectors is less than the size of vocabulary and these vectors are called *topic vectors*.

In topic modeling, documents are mixtures of topics. A topic, in the domain of language models, means a *probability distribution* over a vocabulary of words. This means, given a list of words, each topic has a specific value associated with that word. The list of values represents an individual topic and different topics will have different values associated with each word.

Semantic topics are concisely derived from the co-occurrence of a large number of terms from documents and are used to transform documents so that they may be located in low-dimensional topic space. These dimensionality reduction techniques improve the performance of document representation by overcoming the unavoidable negative influences of the bag-of-words model, such as sparseness, synonyms and polysemy.

Latent Semantic Analysis

Latent semantic analysis (LSA) (Deerwester, Dumais, Landauer, Furnas & Harshman 1990) is a popular technique of analysing semantic relation between terms based on the frequency of term co-occurrence patterns. It creates vector-based representations of texts which are claimed to capture their semantic content, which projects the document vectors onto a proper feature space by using Singular Value Decomposition (SVD) to reconfigure the data as a set of topics related to the documents and terms.

The cricial step of LSA algorithm is to compute the SVD of the nomalized co-occurrence matrix. A $t \times d$ matrix of terms and documents, X, can be decomposed into the product of three other matrices,

$$X(t \times d) = U(t \times r) \cdot S(r \times r) \cdot V^T(r \times d) \tag{2.18}$$

where U containing orthonormal columns known as the left singular vectors, V^T containing orthonormal rows known as the right singular vectors, and S is a diagonal matrix containing the singular values.

An SVD is similar to an eigenvalue decomposition, but can be computed for rectangular matrices (Rohde et al. 2004). The relation between SVD and eigen analysis lie in that, U is the matrix of eigenvectors of the square symmatric matrix XX^T, V is the matrix of eigenvectors of X^TX, and the singular values are akin to eigenvalues.

If the singular values in S are ordered by size, the first k largest may be kept and the remaining small values are set to 0, the terms and documents are converted into a reduced k-dimensional space.

$$\hat{X}(t \times d) = \hat{U}(t \times k) \cdot \hat{S}(k \times k) \cdot \hat{V}^T(k \times d) \tag{2.19}$$

To compare two terms, the dot product between two row vectors $\hat{X}$ reflects the extent to which two terms have a similar pattern of occurrence across the set of documents. The matrix $\hat{X}\hat{X}^T$ is a square symmetric matrix containing all term-to-term dot products,

$$\hat{X}\hat{X}^T = \hat{U}\hat{S}^2\hat{U}^T \tag{2.20}$$

To compare two documents is to calculate the dot product of two column vectors of $\hat{X}$, the matrix $\hat{X}^T\hat{X}$ contains the document-to-document dot products,

$$\hat{X}^T\hat{X} = \hat{V}\hat{S}^2\hat{V}^T \tag{2.21}$$

LSA method uses SVD to decompose the original large term by document matrix to a linear combination of orthogonal factors, deals remarkably with the synonymy problem in latent semantic space.

Probabilistic Latent Semantic Analysis

Compare to standard LSA with stems from linear algebra and performs a SVD of co-occurrence tables, Probabilistic Latent Semantic Analysis (PLSA) (Hofmann 1999) is a novel statistical technique which is a probabilistic version of LSA, based on a mixture decomposition derived from a latent class model. The PLSA approach models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of *topics*. Thus different words in a document may be generated from different topics. Each document is represented as a list of mixing proportions for these mixture components and thereby reduced to a probability distribution on a fixed set of topics. This distribution is the reduced description associated with the document (Blei 2004).

Considering observations of co-occurrences of words and documents, PLSA models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions. Given a collection of text documents $D = \{d_1, d_2, \cdots, d_M\}$ with terms from a vocabulary $W = \{w_1, w_2, \cdots, w_N\}$, PLSA introduces the *Aspect Model* as a latent variable model for co-occurrence

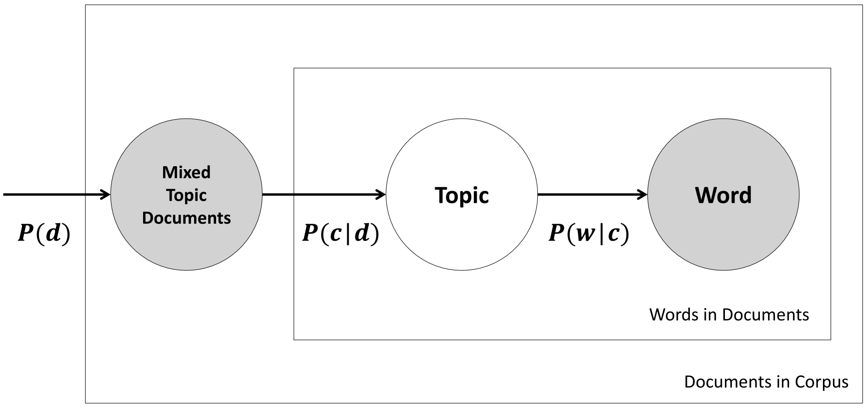


Figure 2.3: Graphical Representation of PLSA

data which associates an observed class variable $c \in C = \{c_1, c_2, \cdots, c_K\}$ with each observation. A joint probability model over $D \times W$ is defined by the mixture

$$\begin{aligned} P(d, w) &= \sum_{c \in C} P(c)P(d|c)P(w|c) \\ &= P(d) \sum_{c \in C} P(c|d)P(w|c) \end{aligned} \tag{2.22}$$

Figure 2.3 describes the model using plate notation. The outer box represents an iteration over every single document. The inner box represents an iteration over every word for each document. The grey circles represent the observed variable. The arrows indicate a dependency. Before a word is drawn from a topic, a new topic must be drawn from a distribution. Each document has its own unique distribution or mixture of topics. This allows individual documents to be composed of words drawn from multiple topics, which makes PLSA a more plausible model of a document's reality than mixture of unigram models (assume only one topic per document).

The Aspect Model introduces a conditional independence assumption, namely that d and w are independent conditioned on the state of the associated latent variable. The number of parameters is equal to $cd + wc$. The number of parameters grows linearly with the number of documents. Their parameters are learned using the EM algorithm.

In statistics, an *Expectation Maximization* (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in latent variable models. The EM iteration alternates between performing two steps:

- An expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters;

- An maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

PLSA is an advanced step toward probabilistic modelling of text which considered more principled than standard LSA, which is an identification of latent classes using Aspect Model to general co-occurrence data and a powerful fitting procedure, Expectation Maximization (EM) algorithm is adapted to avoid overfitting.

However, PLSA fails to construct probabilistic model at the level of documents. Specifically, each document in PLSA is represented by words that from a mixture model, all the mixture components can be regarded as the representations of topics, there is no probabilistic model for these topics. This may causes the number of parameters rises linearly with the number of documents in the training set, may lead to problems with overfitting. In addition, although PLSA is a generative model of the documents in the collection it is estimated on, it is not a generative model because it is not clear how to assign probability to a new document outside of the training set.

Latent Dirichlet Allocation

To proceed beyond PLSA, Latent Dirichlet Allocation (LDA) (Blei, Ng & Jordan 2003) is proposed to capture intra-document statistical structure based on mixing distribution. LDA is a generative hierarchical Bayesian probabilistic model over a corpus. The basic idea is that documents can be considered as random mixtures of various latent topics, where each topic is characterized by a distribution over words. This model provides an explicit representation of a document, where each words generation is attributable to one of the documents topics.

Given a collection of text documents $D = \{d_1, d_2, \cdots, d_M\}$ with words $d = \{w_1, w_2, \cdots, w_N\}$, words are from a vocabulary $W = \{w_1, w_2, \cdots, w_V\}$, LDA assumes the following generative process for each document d in a corpus D,

1. Choose $N|\xi \sim \text{Poisson}(\xi)$.
2. Choose proportions $\theta|\alpha \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
    - Choose a topic $c_n|\theta \sim \text{Mult}(\theta)$.
    - Choose a word $w_n|\{c_n, \beta_{1:K}\} \sim \text{Mult}(\beta_{c_n})$.

Given the parameters α and $\beta_{1:K}$, where K is the dimensionality of the topic variable c, the joint distribution of topic proportions θ, a set of N topics c, and a set of N words w is given by,

$$P(\theta, c, w|\alpha, \beta_{1:K}) = P(\theta|\alpha) \prod_{n=1}^{N} P(c_n|\theta) P(w_n|c_n, \beta_{1:K}) \tag{2.23}$$

where $P(c_n|\theta)$ is simply θ_i for the unique i such that $c_n^i = 1$. Integrating over θ and summing over latent topics, we obtain the marginal distribution of a document,

$$P(d|\alpha, \beta_{1:K}) = \int P(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{c_n} P(c_n|\theta) P(w_n|c_n, \beta_{1:K}) \right) d\theta \tag{2.24}$$

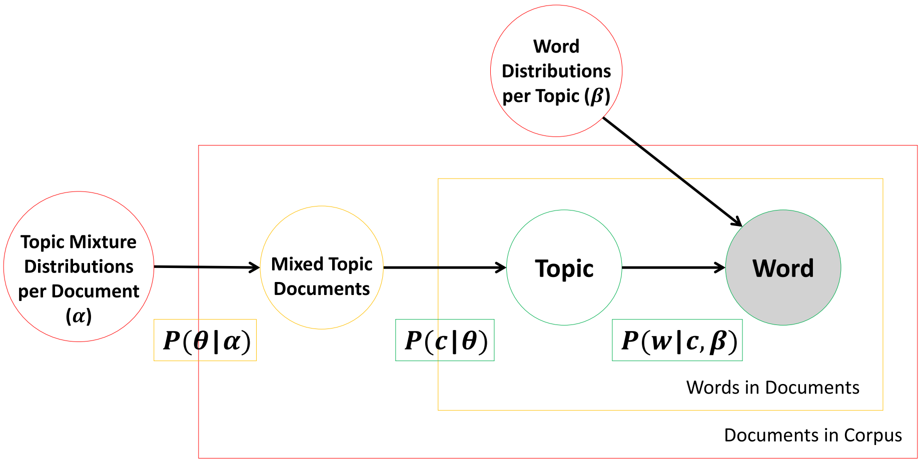


Figure 2.4: Graphical Representation of LDA

Finally, we obtain the probability of a corpus by taking the product of the marginal probabilities of single documents,

$$P(D|\alpha, \beta_{1:K}) = \prod_{m=1}^{M} \int P(\theta_m|\alpha) \left(\prod_{n=1}^{N_m} \sum_{c_{mn}} P(c_{mn}|\theta_m) P(w_{mn}|c_{mn}, \beta_{1:K}) \right) d\theta_m \tag{2.25}$$

LDA is represented as a probabilistic graphical model in Figure 2.4. As the figure makes clear, LDA is a hierarchical model with three levels represented by three colors: (1) corpus-level parameters (red) α and β are assumed to be sampled once in the process of generating a corpus; (2) document-level parameter (orange) θ is sampled once per document, finally (3) the word-level variables (green) include word variables w_{mn} and topic variables c_{mn} are sampled once for each word in each document. The generative process can be described as follows:

- For each topic, sample a distribution over words from a Dirichlet prior.
- For each document, sample a distribution over topics from a Dirichlet prior.
- For each word in the document,

- Sample a topic from the document's topic distribution.
- Sample a word from the topic's word distribution.
- Observe the word.

The parameters can be estimated using two frequently used approaches, Gibbs sampling or variational inference.

LDA fix the problems in PLSA by treating the topic mixture weights as a K-parameter hidden random variable. Then, $K + KV$ parameters in a K-topic LDA model do not grow with the size of the training corpus.

Further Research in Topic Modeling

The simple LDA model provides a powerful tool for discovering and exploiting the hidden thematic structure in texts. It has been widely used as a module in more complicated models for more complicated goals (Blei 2012). Following is a brief introduction of recent literature based on LDA extension.

1. Relaxing the Assumptions of LDA

One active area of topic modeling research is to relax and extend LDA statistical assumptions to uncover more sophisticated structure and more realistic problems in the texts.

One assumption that LDA makes is the words in the document are orderless. This is akin to the standard Bag-of-Words model assumption, and makes the individual words exchangeable. In reality, people will choose to use certain words on the basis of the words used before, it is apparently not realistic and reasonable to ignore the order of words in natural language, and may loss linguistic structure information. There have been a number of extensions to model the words nonexchangeable based on LDA. For example, Wallach (Wallach 2006) explores a hierarchical generative probabilistic model that incorporates both n-gram statistics and latent topic variables to relax this assumption. Word w_t is generated not only on the condition of the topic, but also defined by the previous word w_{t-1}. Typically this is done by computing estimators of both the marginal probability of word w_t, the conditional probability of word w_t following word w_{t-1}, and the word probability conditioned on the topic.

The syntactic topic model (STM) (Boyd-Graber & Blei 2009) also caters this assumption. It is a nonparametric Bayesian topic model that can infer both syntactically and thematically coherent topics. Rather than treating words as the exchangeable unit within a document, the words of the sentences must conform to the structure of a parse tree. In the generative process, the words arise from a distribution that has both a document-specific thematic component and a parse-tree-specific syntactic component.

LDA also assumes that the documents are unordered. This assumption may be unrealistic when analyzing long-running collections that span years or centuries (Blei 2012). To address this problem, a family of probabilistic time series models is developed to analyze the time evolution of topics in large document collections (Blei & Lafferty 2006). The themes in a document collection evolve over time, and it is of interest to explicitly model the dynamics of the underlying topics. In a dynamic topic model, the data is divided by time slice, for example by year. Documents of each slice are modelled with a K-component topic model, where the topics associated with slice t evolve from the topics associated with slice $t-1$. Rather than a single distribution over words, a topic is now a sequence of distributions over words. We can find an underlying theme of the collection and track how it has changed over time.

A third assumption of LDA is that the number of topics is assumed known and fixed. The Bayesian nonparametric topic model (Teh, Jordan, Beal & Blei 2006) provides an elegant solution: the number of topics is determined by the collection during posterior inference, and furthermore, new documents can exhibit previously unseen topics. Bayesian nonparametric topic models have been extended to hierarchies of topics, which find a tree of topics, moving from more general to more concrete, whose particular structure is inferred from the data (Blei, Griffiths & Jordan 2010).

There are still other extensions of LDA that relax various assumptions made by the model. Following are some models build on LDA to solve specific goals.

2. Focusing on Specific Tasks

Topic modeling is an emerging field in machine learning, and there are many exciting new directions for research. Recently, most work of topic modeling focused on specific tasks, such as to consider the influences of context (Chen, Zhou & Carin 2012), and apply topic modeling into sentiment analysis (Mei, Ling, Wondra, Su & Zhai 2007) (Lin & He 2009).

With the popularization of Web applications and other digital media, it raises wide interest in analyzing a large corpus, and it is desirable to place the analysis of such data within the context of other readily available associated information. As the inference of topics associated with any single document is influenced by other documents produced by the same author or published at the same or similar venues, the networks of author and venue information carry significant information. cFTM (Chen et al. 2012) is proposed to utilize the interrelationships between author names and publication venues, to allow an appropriate sharing of information from multiple documents. It automatically infers the number of topics, the number of author and venue clusters, and the probabilistic importance of the author and venue information on word assignment in a document dependent manner.

There are also great bulk of work has been focused on the problem of sentiment analysis at various levels combing topic models. Sentiment analysis or opinion mining aims to use automated tools to detect subjective information such as opinions, attitudes, and feelings expressed in text. Topic/feature detection and sentiment classification are often performed in a two-stage pipeline process, by first detecting a topic/feature and later assigning a sentiment label to that particular topic, which give us the intuition that sentiment polarities are dependent on topics. A Topic Sentiment Mixture (TSM) (Mei et al. 2007) is proposed to model and extract the multiple subtopics and sentiments in a collection of blog articles. Specifically, a blog article is assumed to be generated by sampling words from a mixture model involving a background language model, a set of topic language models, and two (positive and negative) sentiment language models. With this model, the topic/subtopics can be extracted from blog articles, reveal the correlation of these topics and different sentiments, and further model the dynamics of each topic and its associated sentiments. The unsupervised joint sentiment/topic model (JST) (Lin & He 2009) is proposed beyond TSM, which detects sentiment and topic simultaneously from text by adding a sentiment layer, the word from the distribution over words is defined not only by the topic, but also the sentiment label.

Summary

In the domain of language models, a topic is a probability distribution over the terms in a vocabulary. A topic modeling tool looks through a corpus for these clusters of words and groups them together by a process of similarity. The following points provide a brief overview of topic modeling:

- The unigram model uses a single topic in the entire corpus. Each document in the corpora is composed of words selected from a single topic distribution for the entire corpus.

- The mixture of unigrams model introduces the possibility of multiple topics and a distribution of topics from which we draw a new distribution of words for each document.

- PLSA allows individual documents to be composed of multiple topics.

- In LDA, each document is represented as a mixture of latent topics, and each topic is represented as a mixture of words. These mixture distributions are assumed to be Dirichlet-distributed random variables which must be inferred from the data.

Topic modeling proposes alternative ways to define the semantic matrix though feature mapping. Starting with the simple unigram model, to the mixture of unigrams, to probabilistic latent semantic analysis, to latent dirichlet allocation, researchers have developed more and more advanced topic models to discover and exploit the hidden thematic structure of texts, and these have been broadly extended and adapted to cover more sophisticated structure and specific tasks.

2.2.3 Discussion and Conclusion

The above-mentioned corpus-based representations offer two different ways to capture the semantics of terms.

Term-based document representations consider term semantic relations by extracting the contexts of terms from large corpora and then redistributing the terms into new feature space. Vector space models and extensions involve more and more interactions between terms to enrich their correlations. From the earliest only term frequency considered, researchers are now concerned with the terms co-occurrence frequency, contextual information, location information and even more implicit interactions between terms. However, term-based models split texts into the smallest units and then try to connect them by various information they share which reveals little in terms of document statistical structure. In addition, this may cause complexity problems because of the high-dimensional vectors, and also cannot avoid ambiguity problems because more recapitulative information is overlooked.

Intuitively, an article is easier to understand if it is represented by main topics. To address the shortcomings in term-based models, a large amount of literature has been proposed to develop a generative probabilistic model of text corpora. Topic modeling is a form of text mining, a way of identifying patterns in a corpus. You take your corpus and run it through a tool which groups words across the corpus into topics. Miriam Posner has described topic modeling as a method for finding and tracing clusters of words (called topics in shorthand) in large bodies of texts.

A basic difference between the topic modeling methods and semantic smoothing methods is related to the dimension of the new feature space. In topic models, each document in a given corpus is represented by a histogram containing the occurrence of words. The histogram is modeled by a distribution over a certain number of topics, each of which is a distribution over words in the vocabulary. By learning the distributions, a corresponding low-rank representation of the high-dimensional histogram can be obtained for each document (Hofmann 2001). As their feature dimension is less than the size of the vocabulary, such vectors are called *topic vectors* (Kalogeratos & Likas 2012) which have a distribution of weights associated with the original terms that define their contribution to the corresponding topic. In addition, topic models help handle the polysemy and synonymy that lie in Bag-of-Words models, since the count for a topic in a document can be much more informative than the count of individual words belonging to that topic.

In the next section, lexical resource-based approaches will be reviewed which capture semantics which rely on pre-existing knowledge resources.

2.3 Lexical Resource-based Semantic Representation

All of the above models compare terms as Bag-of-words in the vector space of the corpus. However, terms are different from other features in statistical learning, because the sequence of terms is not arbitrary. Indeed, it must follows a certain grammar structure from the natural language perspective. Moreover, terms often represent some concepts which can always be organised by the relation of the hyponym or hypernym. For many large-scale data mining tasks, including classifying and clustering documents, it is sufficient to use a simple representation that loses all information about structure. Lexical resource-based approaches are raised for the purpose of integrating heterogeneous databases, and relevant tools refer to establishing relations among ontological resources as well as assessing the similarities and semantics.

Considering the contents of an information item and its intended meaning, semantic conflicts are mainly caused by confounding, homonyms and synonyms. The use of ontology for the explication of implicit and hidden knowledge is a possible approach to overcome these problems. Initially, ontology is defined as an explicit specification of a conceptualisation (Gruber 1993) and the common components of ontology include individuals, classes, attributes, relations, rules, etc., which can be used to identify and associate semantically corresponding information concepts.

2.3.1 Edge-based Semantic Representation

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 the path length between the concepts.

In early research the main assumption that capture the similarity between two concepts is to find the shortest-path linking the two concept nodes in a taxonomy graph (Rada, Mili, Bicknell & Blettner 1989). This approach is taken on MeSH (Medical Subject Headings)[1], a semantic hierarchy of terms used for indexing articles in the bibliographic retrieval system Medline. The network has 15000 terms form a nine-level hierarchy based on the broader-than relationship, which can be as formal as a set of meronymic and hyper/hyponymic relationships. The principal assumption of (Rada et al. 1989) is that the number of edges between terms in the MeSH hierarchy is a measure of conceptual distance between terms. The shorter the distance, the more similar the concepts are semantically. Despite the simplicity of this distance function, the authors were able to obtain surprisingly good results in their information retrieval task (Budanitsky & Hirst 2006).

[1] https://www.nlm.nih.gov/mesh/

However, solely counting links between nodes is not sufficient. The example given in (Richardson, Smeaton & Murphy 1994) illustrates this problem. The distance between "plant" and "animal" is 2 in WordNet[2] since their common parent, is "living thing". The distance between "zebra" and "horse" is also 2 since their common parent is equine. Intuitively, one would judge "zebra" and "horse" more closely related than "plant" and "animal".

To overcome the limitation of simple edge counting, many other approaches suggest considering more characteristics when calculating the weight between two concepts. The edges are then weighted to reflect the difference in edge distances by using different information about the edge in determining its weight: including the concept depth, the density of edges at that depth, the type and the strength of the relation that connect two concepts (Cross 2004).

2.3.2 Information-theoretic Semantic Representation

More advanced methods consider the semantic similarity of concepts based on the *information content* (IC) they share on taxonomy or more sophisticated structure. Information content is an important dimension of word knowledge when assessing the similarity of two terms or word senses (Seco, Veale & Hayes 2004). The information-theoretic models of semantic similarity add to the information already present in the network by using a qualitatively different, knowledge source. The groundwork for much of this research is founded on the insight that conceptual similarity between two concepts may be judged by the degree to which they share information (Cross 2004). The more information they share then the more similar they are. Different methods have been used to approximate that information content.

[2] https://wordnet.princeton.edu/

Corpora-based IC Models

The conventional way of measuring the IC of word senses is to combine knowledge of their hierarchical structure from an ontology like WordNet with statistics on their actual usage in text as derived from a large corpus. The probability of each concept in the taxonomy is based on term occurrences frequency in a given corpus.

Resnik (Resnik et al. 1999) proposes a way to edge counting methods based on the information-theoretic hypothesis, which is regarded as the foundation of IC learning. Based on standard information theory, the IC of a concept c is obtained by computing the inverse of its appearance probability in a corpus,

$$IC(c) = -\log P(c) \tag{2.26}$$

where $P(c)$ is probability of encountering an instance of c in a specific corpus.

$$P(c) = \frac{\sum\limits_{w \in W(c)} count(w)}{N} \tag{2.27}$$

where $W(c)$ is the set of terms in the corpus whose senses are subsumed by c, and N is the total number of corpus terms that are contained in the taxonomy.

The information shared by two concepts c_1 and c_2 is approximated by the information content of the lowest super-concept c_3 that subsumes them in the hierarchy. The similarity between c_1 and c_2 is given as

$$sim(c_1, c_2) = \max_{c \in S(c_1, c_2)} \big(-\log P(c) \big), \tag{2.28}$$

This quantitative characterization of information provides a new way to measure semantic similarity. The information shared by two concepts is indicated by the information content of the concepts that subsume them in the taxonomy. The higher position of the super-concept, the more abstract thay are, and therefore, the lower semantic similarity between c_1 and c_2.

A step forward, the semantic similarity between two concepts proposed in (Lin 1998) takes both *commonality* and *difference* into account. It is argued that the similarity between two concepts is not about the concepts but about the instances of the concepts. For example, when we say "horse and zebra are similar", we are not comparing the set of horse with the set of zebra. Instead, we are comparing a generic horse and a generic zebra. Then the amount of information contained in $x_1 \in c_1$ and $x_2 \in c_2$ is

$$-\log P(c_1) - \log P(c_2) \tag{2.29}$$

where $P(c_1)$ is the probability that a randomly selected object x_1 would belong to c_1. If concept c_3 is the most specific concept that subsumes both c_1 and c_2, then

$$Sim(c_1, c_2) = \frac{2\log P(c_3)}{\log P(c_1) + \log P(c_2)} \tag{2.30}$$

Lin's measure of similarity between two concepts in a taxonomy is a corollary of similarity theorem, which defines that the similarity between A and B is measured by the ratio between the amount of information needed to state their commonality and the information needed to fully describe what they are (Budanitsky & Hirst 2006).

A combined model (Jiang & Conrath 1997) is proposed that derived from the edge-based notion by adding the information content as a decision factor. The distance of the edge connecting a child-concept c to its parent-concept $par(c)$ is formatted as

$$dist\big(c, par(c)\big) = logP\big(c|par(c)\big) \tag{2.31}$$

where

$$P\big(c|par(c)\big) = \frac{P\big(c \cap par(c)\big)}{P\big(par(c)\big)} = \frac{P(c)}{P\big(par(c)\big)} \tag{2.32}$$

Following the standard argument of information theory,

$$dist\big(c, par(c)\big) = IC(c) - IC\big(par(c)\big) \tag{2.33}$$

The overall distance between two nodes would thus be the summation of edge weights along the shortest path linking two nodes.

$$\begin{aligned} dist(c_1, c_2) &= IC(c_1) + IC(c_2) - 2 \times IC(c_3) \\ &= 2(c_3) - \big((c_1) + (c_2)\big) \end{aligned} \tag{2.34}$$

where c_3 denotes the lowest super-ordinate of c_1 and c_2. The final similarity is calculated as the deference between the total sum and the information content of the most specific common super-concept.

Previous information-theoretic approaches obtain the needed IC values by statistically analyzing corpora, word senses should be identified in the corpus in order to accurately compute concept appearance probabilities. It causes some problems: (1) it takes time to analyze text, and resulting probabilities will depend on the size and nature of input corpora; (2) there are high and strict request of the corpus, contents of the corpus should be adequate with respect to the ontology scope and big enough to avoid data sparseness; moreover, (3)the background taxonomy should be complete enough to include most of the specializations of each concept covered in the corpus, in order to provide reliable results at a conceptual level (Sánchez et al. 2011).

Ontology-based IC Approaches

Based the limitations of corpus-based IC computation models, a number of literature is produced on the basis of the assumption that previously established ontologies can also be used as a statistical resource and produce the IC values needed for semantic similarity calculations with no need for external ones.

Seco et al. (Seco et al. 2004) were the first to base information content calculations on the number of concept hyponyms. They assumed that the taxonomic structure like WordNet is organized in a meaningful and structured way, where concepts with many hyponyms (i.e., specialisations) contain less information than concepts that are leaves, and concepts that are leaf nodes are the most specified in the taxonomy so the information they express is maximal, as they are not further differentiated.

Being $hypo(c)$ the number of hyponyms in the taxonomical tree below the concept c and max_nodes the maximum number of concepts in the taxonomy, they compute information content of a concept in the following way:

$$IC(c) = \frac{\log\left(\frac{hypo(c)+1}{max\_nodes}\right)}{\log\left(\frac{1}{max\_nodes}\right)} = 1 - \frac{\log\big(hypo(c)+1\big)}{\log(max\_nodes)} \tag{2.35}$$

Obviously, this metric only take into account of the hyponyms of a given concept, but assigns the same score to all leaf nodes in the taxonomy regardless of their overall depth. Concepts that have equal number of hyponyms but different degrees of generality will be given the equal similarity. To address this limitation, Zhou et al. (Zhou, Wang & Gu 2008) proposed a hyponym-depth-combined IC computation model,

$$IC(c) = k\left(1 - \frac{\log\big(hypo(c)+1\big)}{\log(max\_nodes)}\right) + (1-k)\left(\frac{\log\big(depth(c)\big)}{\log(max\_depth)}\right) \tag{2.36}$$

where $depth(c)$ returns the depth of concept c in the taxonomy and max_depth is the max depth of the taxonomy. k is a tuning factor so as to control the weight of the two items of equation. Based on this model, the concepts with same hyponyms and different depth can be discriminated.

In WordNet, around 21% of the total amount of concepts correspond to inner taxonomical nodes (Devitt & Vogel 2004). For different ontology, the level of inner taxonomical detail may vary. This influences the coherency of intrinsic IC computations relying on the total size of the hyponym tree. To avoid depending on the inner-detail of the hierarchy, Sanchez et al. (Sánchez et al. 2011) proposed a improved IC-based model to better capture the semantic similarity in an ontology for the particular concept, they computes the information content of one term by considering both

1. the number of its descendants (i.e., *leaves*) in a taxonomy, as leaves present maximum IC as they are completely differentiated (i.e., not further specialized) from any other concept in the taxonomy;

2. the depth of a concept in a taxonomy, i.e., its number of taxonomical subsumers.

The IC of a concept is defined as

$$IC(c) = -\log\left(\frac{\frac{|leaves(c)|}{|subsumers(c)|}+1}{max\_leaves+1}\right) \tag{2.37}$$

where $leaves(c)$ and $subsumers(c)$ define the set of leaves and subsumers of a concept c, respectively. max_leaves represents the number of leaves corresponding to the root node of the hierarchy.

It has been proved that Equation 2.37 monotonically increase as one moves down in the taxonomy, which is according to the basic assumption that concept specializations gain more information in the taxonomy. The method performs to better capture the generality/concreteness of a concept, in addition, it avoids problem in Equation 2.36 depending on tuning parameters.

Since taxonomy (tree) -based semantic similarity measures has been well-studied from either edge-based or information-theoretic perspective, the design of similarity measures for objects stored in the nodes of arbitrary graphs is an open problem.

Graph-based Measures

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 (is-a links) and non-hierarchical (cross links) concepts in an ontology (graph).

Maguitman et al. (Maguitman, Menczer, Roinestad & Vespignani 2005) defined a graph-based semantic similarity measure that generalized from the information-theoretic tree-based similarity on the Open Directory Project (ODP)[3]. The ODP ontology is more complex than a simple tree. Categories can be classified by multiple criteria, a node may have multiple parent nodes, graph edges can have diverse types (e.g., is-a, symbolic, related). Maguitman et al. uses MaxProduct fuzzy composition to define fuzzy membership value

[3] http://www.dmoz.org

for each concept, then the semantic similarity of two concepts $sim(c_1, c_2)$ can be calculated from the fuzzy membership matrix W.

$$sim(c_1, c_2) = \max_k \frac{2\min(W_{k1}, W_{k2}) \cdot \Pr(c_k)}{\log\big(\Pr(c_1|c_k)\cot\Pr(c_k)\big) + \log\big(\Pr(c_2|c_k)\cot\Pr(c_k)\big)} \tag{2.38}$$

where the probability $\Pr(c_k)$ represents the prior probability that any document is classified under topic c_k and is calculated as

$$\Pr(c_k) = \frac{\sum\limits_{c_j \in V} (W_{kj} \cdot |c_j|)}{|N|} \tag{2.39}$$

N is the total number of documents in the ontology.

It is the first information-theoretic measure of similarity that is applicable to objects stored in the nodes of arbitrary graphs, in particular conceptual ontologies and Web directories that combine hierarchical and non-hierarchical components. However, this measure doesn't consider some important properties like the depth of the ontology and the dense of concepts.

Song et al. (Song, Ma, Liu, Lian & Zhang 2007) proposed a fuzzy semantic similarity measure based on information theory that exploits both the hierarchical and non-hierarchical structure in ontology. This method also utilize ontology graphs as adjacency matrices, but caters some limitations in Maguitman's work, concerning the depth, the density and various relation types in the Open Knowledge Base Connectivity (OKBC)[4]. The semantic similarity between two concepts is determined by computing two semantic extended fuzzy sets, which includes all ancestral concepts in hierarchical structure and all concepts that have a layer non-hierarchical semantic relation with these ancestral concepts, formatted by

$$c^+ = \left\{ \frac{1}{c}, \frac{sim(c, c_p)}{c_p}, \cdots, \frac{sim(c, root)}{root} \right\} \tag{2.40}$$

where c_p denotes the parents concepts of c and the parents of these parents, $sim(c, c_p)$ reflects the similarity degree of c and c_p, it can be used directly from the semantic relation matrix W.

[4] http://www.ai.sri.com/ okbc/

Then the fuzzy similarity measure is computed based on shared information content, which measures the degree of similarity proportional that the pairwise concepts share. Based on (Song et al. 2007), it could reflect latent semantic relation of concepts better than ever.

Another graph-based approach (Hawalah & Fasli 2011) use arbitrary ontology GBSRO to propose a six-stage strategy to deal with different type of semantic relatedness between both hierarchical and non-hierarchical concepts, where each stage deals with a particular aspect of relatedness, and each kind of relatedness is presented by an adjacency matrix, then all matrices are integrated into one to represent the final semantic relatedness across all concepts.

This approach measures four different type of semantic relation: directed-related, transitive-related, sibling-related, parent-related concepts, four adjacency matrices are aggregated using a composition function $\otimes$ as follows

$$\begin{aligned} M &= M_{dr} \otimes M_{tr} \otimes M_{sr} \otimes M_{pr} \\ &= \max\big(M_{dr}, M_{tr}, M_{sr}, M_{pr}\big) \end{aligned} \tag{2.41}$$

where M is the aggregated adjacency matrices that represents the different type of semantic relatedness of pairwise concepts, M_{dr}, M_{tr}, M_{sr} and M_{pr} represents directed-related, transitive-related, sibling-related and parent-related matrix, respectively.

The graph-based measures focus on ontologies with more complex structure as well as various relation types including both hierarchical and non-hierarchical ones. Compared to taxonomic tree-based methods, graph-based methods are more general to measure the relation between concepts in arbitrary ontologies.

2.3.3 Wikipedia-based Semantic Representation

It has been recognized that hand-crafted lexical databases like WordNet or Roget's Thesaurus[5] provide limited knowledge of the language lexicon,

[5] http://www.thesaurus.com/

from both scope and scalability perspectives. With the rapid development of World Wide Web, much larger ontologies which constantly evolving huge amount of text are required to capture the semantics of terms. Approaches based on the largest encyclopedia in existence, the *Wikipedia*[6] have been widely developed for this purpose. With its extensive network of cross-references, portals and categories it also contains a wealth of explicitly defined semantics (Witten & Milne 2008).

Wikirelate

Strube and Ponzetto (Strube & Ponzetto 2006) are the first to compute semantic relation using Wikipedia. They propose the *Wikirelate* which take the familiar path-length techniques that has been previously applied in WordNet. Given a term pair, Wikirelate searches out the corresponding articles that contain each term, and adapt various distance measures in Wikipedia hierarchy to compute the semantic relation.

Explicit Semantic Analysis

Gabrilovich and Markovitch (Gabrilovich & Markovitch 2007) propose *Explicit Semantic Analysis* (ESA) to explicitly represent any unrestricted natural language texts in terms of Wikipedia-based concepts. ESA is somewhat reminiscent of vector space model, Wikipedia concepts are all turned into "bags of words", i.e., inverted index that store the term frequency. The output of the inverted index for a text fragment is a list of indexed documents (Wikipedia concepts), each given a score depending on how often the text occurred in them (weighted by the total number of words in the document).

Concretely, given a text fragment, we first represent it as a vector using tf-idf scheme. The semantic interpreter iterates over the text words, retrieves corresponding entries from the inverted index, and merges them into a weighted vector of concepts that represents the given text. Let $T = \{w_i\}$ be input text, and let $\langle vi \rangle$ be its tf-idf vector, where v_i is the weight of

[6] https://www.wikipedia.org/

word w_i. Let $\langle kj \rangle$ be an inverted index entry for word w_i, where k_j quantifies the strength of association of word w_i with Wikipedia concept c_j, $\{c_j \in c_1, \cdots, c_N\}$, where N is the total number of Wikipedia concepts. Then, the semantic interpretation vector V for text T is a vector of length N, in which the weight of each concept c_j is defined as

$$\sum_{w_i \in T} v_i \cdot k_j$$

Mathematically, it is an N-dimensional vector of word-document scores, entries of this vector reflect the relevance of the corresponding concepts to text T. To compute semantic relatedness of a pair of text fragments we compare their vectors by computing the cosine similarity.

Compared to Wikirelate that only compute the semantic relation of single words occurring in Wikipedia titles, ESA is free to compare texts of any length that appear within the texts of Wikipedia articles. In addition, ESA is explicit in the sense that the concepts are manipulated manifestly in human cognition, rather than latent concepts used by Latent Semantic Analysis (LSA).

Wikipedia Link-based Measure

Rather than measures based on Wikipedia category hierarchy or textual content, Milne and Witten (Witten & Milne 2008) propose a hyperlink-based measure which calculates semantics between terms using the links found within their corresponding Wikipedia articles.

At first they use an anchor-based approach to identify the Wikipedia articles that discuss the term pair, as well as avoid word ambiguity. Then they offer two different ways, tf-idf inspired approach and Google distance inspired approach to measure similarity between articles. In specific, the first measure is almost identical to tf-idf weighting scheme, the weight of a link is the inverse probability of any link being made to the target,

$$w(s \to t) = \log\left(\frac{|W|}{|T|}\right) \tag{2.42}$$

where T is the set of all articles that link to t, $s \in T$,and W is the set of all articles in Wikipedia. The other measure is based on Wikipedia's links inspiring by Google distance approach,

$$sr(a,b) = \frac{\log\big(\max(|A|,|B|)\big) - \log\big(|A \bigcap B|\big)}{\log\big(|W|\big) - \log\big(\min(|A|,|B|)\big)} \tag{2.43}$$

where $sr(a,b)$ is the semantic relation of two articles a and b, A and B are the sets of all articles that link to a and b respectively. They also compare five options on measuring relatedness between terms.

Temporal Semantic Analysis

Later, Temporal Semantic Analysis (TSA) (Radinsky, Agichtein, Gabrilovich & Markovitch 2011) was proposed to further improved the performance of ESA by extending it with a temporal dimension. This way, the computation of semantics was augmented with patterns of word occurrence over time (e.g., in an archive of a 100 years worth of New York Times articles).

In TSA, hypothesis is that concepts that behave similarly over time, are semantically related. Each concept is no longer scalar, but is instead represented as time series (i.e., dynamics) over a corpus of temporally-ordered documents. Let $t_1, \cdots, t_n$ be a sequence of consecutive discrete time points (e.g., days), $H = D_1, \cdots, D_n$ be a history represented by a set of document collections, where D_i is a collection of documents associated with time t_i. Let c be a concept represented by a sequence of words $w_{c1}, \cdots, w_{ck}$, d is a document, the dynamics of a concept c is the time series of its frequency of appearance in H,

$$Dynamics(c) = \left\{ \frac{|\{d \in D_1 | appears(c,d)\}|}{|D_1|}, \cdots, \frac{|\{d \in D_n | appears(c,d)\}|}{|D_n|} \right\} \tag{2.44}$$

To compute semantic relatedness of a pair of words is to compare their vectors using measurements of weighted distance between multiple time series, combined with the static semantic similarity measure of the concepts. The similarity between individual time series is based on Dynamic Time

Warping (DTW) algorithm, in addition, consider the recent concepts have stronger correlation than past concepts, the DTW is combined with a temporal weighting function as

$$\|t_{s1}(i) - t_{s2}(j)\| \cdot f(i, j)$$

where $\|t_{s1}(i) - t_{s2}(j)\|$ is the DTW metric between two time series t_{s1} and t_{s2}, $f(i, j)$ is such a temporal weighting function that supports various linear or non-linear functions.

TSA is the first approach to compute semantic relatedness with the aid of a large scale temporal corpus, and yields statistically significant improvements in correlation of computed relatedness scores.

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.

2.3.4 Discussion and Conclusion

Ontologies provide a highly expressive ground for describing keywords or concepts and a rich variety of interrelations among them (Hawalah & Fasli 2011). Unlike simple methods of representing information such as weighted keywords and semantic networks, an ontology provides a more powerful, deeper and broader concept hierarchy representation. Some of the most important advantages of developing and using ontologies are summarised below: (Hawalah & Fasli 2011):

1. sharing of common understanding of the information;
2. the ability to reuse a domain ontology;
3. support for context reasoning;
4. powerful information and context management and inference mechanisms;

5. reduction of ambiguity and solving the inconsistency of the information.

Traditional lexical resource-based semantic measurements compute the relatedness of concepts by counting the shortest path length. Furthermore, new trend considers the information content which is shared between two concepts. This takes into account not only taxonomic relations, but also different types of relations including hierarchical and non-hierarchical concepts. Additionally, larger and more comprehensive knowledge repositories are broadly adapted rather than relying on curated lexical databases which provide a limited language lexicon.

2.4 Summary

Quantifying the semantic relatedness of terms or texts underlies many fundamental tasks in natural language processing, including information retrieval, word sense disambiguation, and document clustering (Radinsky et al. 2011). To compute semantic relatedness, we must consult external sources of knowledge. Existing methods employ various linguistic resources, such as large-scale text corpora or hand-crafted lexical structures.

In Chapter 2, a number of existing semantic measurements are reviewed and evaluated from corpus-based and lexical resource-based categories, which offer two different ways to capture the semantic similarity and relation between terms. For corpus-based models, researchers try to model lexical semantic information in high-dimensional vectors, by considering term occurrence patterns, contexts, locations, etc. While lexical resource-based approaches estimate semantics by defining a topological similarity and by using lexical ontologies to measure the distance between terms or concepts. These approaches rely on hand-crafted resources such as thesauri, taxonomies, semantic networks or encyclopedias, as the context for comparison (Li et al. 2013). The databases do not provide a term-similarity metric but various metrics based on its structure have nonetheless been developed.

The table 2.1 below summarises and evaluates all of the methods mentioned in Chapter 2, and compares with my proposed approaches in terms of the various semantic representation merits.

Table 2.1: The Summary of Semantic Representations

Semantic Representation Type	A	B	C	D	E
Vector Space Models			√		
Topic Models	√	√	√		
Edge-based Models			√		
Information-therotic Models	√		√		
Wikipedia-based Models	√		√		
Term Pair Semantic Coupling Model	√	√	√	√	
Hierarchical Tree Learning Model	√	√		√	√

where A,B,C,D and E denote five different merits, respectively, A is *Ambiguity*, B is *Implicit Relation*, C is *High accuracy*, D is *Semantic Couplings*, and E is *High-order Semantics.*

In summary, different efforts have been made to address semantic relatedness issues. Due to the intrinsic complexities of natural language, there is more work to do on deeply exploring term semantic relationships and representing semantic similarity. In Chapter 3, SCS is built to solve natural language ambiguity, and non-iidness theory (Cao 2013) is adapted to handle unstructured textual data and the complex relationship of concepts. SCS attempts to capture the semantic relatedness in a coupled thought, by combining the graph-based method and statistical-based method. This helps to mine in detail the explicit and implicit relatedness of terms pairs. In Chapter 4, a hierarchical tree is constructed to simultaneously handle high order semantics and address the time-consuming nature of SCS by proposing a hierarchical feature extraction algorithm to map the document into a much smaller feature space.

Chapter 3

Semantic Representation: Capturing Explicit and Implicit Content Couplings

3.1 Introduction

Document similarity analysis is increasingly critical since roughly 80% of big data is unstructured. The way that terms, words and phrases in a document are organized reflects certain explicit and implicit coupling relationships embedded in its contents, syntactic/linguistic or even subjective perspectives (Gabrilovich & Markovitch 2007). It is challenging to extract the complete latent relation over terms (words or phrases) according to the difficulty of exploiting the explicit and implicit relation between terms.

Coupling refers to any relationships (for instance, co-occurrence, neighborhood, dependency, linkage, correlation, or causality) between two or more aspects (Cao 2013). Accordingly, the effective capturing of such content couplings is thus crucial for a genuine understanding of document similarity, which has emerged as a promising and important topic recently, such as in semantic relatedness (Strube & Ponzetto 2006, Gabrilovich & Markovitch 2007), content coverage (Holloway et al. 2007), word networking (Budanitsky

& Hirst 2006, Agirre et al. 2009), term-term couplings (Cheng et al. 2013), as well as general problems including information retrieval (Billhardt et al. 2002, Hliaoutakis et al. 2006), ontological engineering (Hawalah & Fasli 2011, Li et al. 2013), and document clustering and classification (Farahat & Kamel 2011, Kalogeratos & Likas 2012).

The problem of document similarity can be further decomposed to explore the coupling relationships and similarity between terms (words or phrases) which forms a document. This is to build a feature space that consists of all necessary terms with their couplings 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 (Cao 2013).

In documents, content couplings may be caused by various reasons and in different forms, term couplings could be very complicated. Challenges are hidden in the couplings between terms and documents, for instance, meronymy, antonymy, functional association, and others (Budanitsky & Hirst 2006). Existing methods focus on occurrence patterns (Bullinaria & Levy 2007), word networking (Castillo 2011, Wang, Yu, Li, Zhai & Han 2013), topic distribution (Blei et al. 2003, Teh et al. 2006), ontological distribution (Sánchez et al. 2011), latent semantic analysis (Arora & Ravindran 2008, Miao, Guan, Moser, Yan, Tao, Anerousis & Sun 2012), or on a natural language processing perspective (Jackson & Moulinier 2007) (which is a very different topic).

Those recent efforts can be roughly characterized into two categories: corpus-based statistical and topological measures. More specifically, term relation estimated by corpus-based statistical means such as vector space models and its extensions, compute the co-occurrence frequency patterns of terms and textual contexts across corpus; probabilistic models are developed to discover the distribution properties of each term over topics and the topic distribution over each document. Instead, topological approaches capture the relation between terms or concepts by using ontologies to define the distance between them; most of such methods rely on pre-existing knowledge resources that are represented by a directed or undirected graph consisting of vertices, for example, semantic networks and taxonomies.

Significant gaps remain in the literature that are to effectively capture the sophisticated couplings not only between explicitly linked terms but also implicitly related terms for various reasons and in different forms, including topological and statistical aspects. These involve significant challenges hidden in the couplings between terms and documents, for instance, meronymy, antonymy, functional association, and others.

Our method addresses the above issues from topological and statistical perspectives, by proposing a graphical representation of both explicit and implicit content (term) couplings, addressing synonymy (many words per sense) and polysemy (many senses per word), which are overlooked by previous models. We propose the **semantic coupling similarity (SCS)** measure, consisting of

- the intra-term pair couplings, reflecting the explicit couplings within term pairs that is represented by the relation strength over probabilistic distribution of terms across document collection;
- the inter-term pair couplings, capturing the implicit couplings 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;
- coupled semantic couplings, effectively combining the intra- and inter-couplings. The corresponding term semantic similarity measures are then defined to capture such relation for analyzing term and thus document similarity.

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

- (i) A statistical measure to capture the intra-term couplings within term pairs by adapting a relation strength function to calculate the similarity between a pair of terms as per their probabilistic distributions. The measure counts the term pair occurrence frequency - inverse document frequency (tpf-idf) across the document set.

- (ii) A graph-based measure to capture the inter-term couplings between term pairs by measuring the relation strength of every term pair distribution. Inter-coupling is measured by the term pair occurrence frequency - inverse path frequency (tpf-ipf) weighting scheme on all possibly indirectly connected paths when term couplings are presented by a graph.

- (iii) 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. SCS is then incorporated into hierarchical clustering to cluster documents, showing impressive performance compared to the typical algorithms on multiple real textual data sets.

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 statistically significant and exceed the performance of benchmark methods consistently on all data sets.

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, and non-iidness theory (Cao 2013) is adapted here to handle unstructured textual data and complex relationships of concepts. In the next section, the proposed research methodology is discussed, which attempts to capture the semantic relatedness in a coupled thought, by combining the graph-based method and the statistical-based method together, to deeply mine the explicit and implicit relatedness of terms pairs.

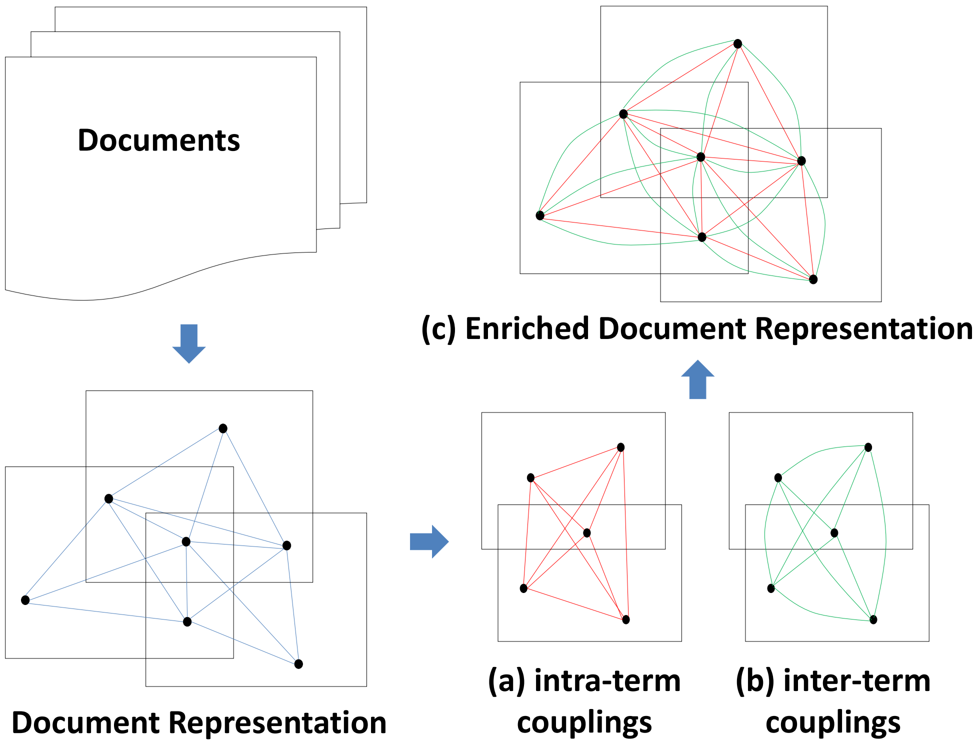


Figure 3.1: An overview of term pair semantic coupling analysis

The remainder of this chapter is organized as follows: Section 2 reviews and evaluates the related work of document similarity representations from topological and statistical measures. Section 3 proposes the coupled term pair similarity and its application in document analysis. Section 4 demonstrates the experimental results of clustering analysis on real document sets, and the comparison with prevalent existing approaches. Finally, conclusion and future work are described in Section 5.

3.2 Term Pair Semantic Coupling Analysis

In this section, a semantic coupling similarity (SCS) measure is proposed to comprehensively capture the couplings within and between term pairs from two aspects: the semantic intra-term coupling, capturing the similarity of probabilistic distribution of every term pair based on its occurrence frequency pattern across the data set; and the semantic inter-term coupling, considering the underlying semantic relatedness by capturing the similarities of term pair distributions on indirectly connected paths. Figure 3.1 illustrates the process: (a) SCS calculates the intra-couplings within term pairs by considering their co-occurrence frequency across a document set; (b) it further constructs bridges formed by linked term pairs to compute the inter-coupling between term pairs; and (c) it integrates the intra- and inter-couplings to obtain the complete semantic similarity.

3.2.1 Semantic Intra-couplings within Term Pairs

The intra-couplings of term pairs explore the explicit relatedness between terms. This is to conduct the statistical analysis of term co-occurrence patterns by assuming that terms are regarded coupled if they co-occur in the same document. The more frequently they co-occur, the stronger couplings they have. Accordingly, the intra-couplings between terms are estimated based on the term co-occurrence frequency across all documents.

To this end, the weighting scheme *term frequency - inverse document frequency*, short for *tf-idf*, *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 (Jing et al. 2002), 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-idf* is computed as:

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \tag{3.1}$$

A high weight in *tf-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-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, the *tf-idf* based methods have two main limitations. One is that they place undue emphasis on the documents where terms co-occur; the other is that *tf-idf* based on one single term may lead to synonymy and polysemy, since the semantic meaning of a term in different documents can be various. To solve these problems, we propose the *term pair occurrence frequency - inverse document frequency* (*tpf-idf*) for term pairs, defined as follows:

Definition 3.1 ***tpf-idf*** *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 tpfidf is formatted as:*

$$tpfidf((t_i, t_j), d, D) = tpf((t_i, t_j), d) \times idf((t_i, t_j), D) \tag{3.2}$$

where (t_i, t_j) stands for a term pair, d is a single document in a document collection D, $tpf((t_i, t_j), d)$ means the frequency of term pair t_i and t_j in d, and $idf((t_i, t_j), D)$ indicates the inverse document frequency that contains the term pair t_i and t_j.

The term pair occurrence frequency matrix M_{tpf} represents the occurrence frequency of every term pair in D, which 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-idf* scheme, terms appear as pairs, the meaning of a single term is more semantically complete compared with *tf-idf*. It is used to depict the real explicit relatedness of term pairs by adapting statistical distance measures for a solid statistical significance.

Accordingly, the probability of the term pair (t_k, t_i) in document set D, i.e. $P^{Ia}(t_k|t_i)$, and the probability over all term pairs, $P^{Ia}(t_i)$, for a given t_i, are defined as follows:

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

$$P^{Ia}(t_k|t_i) = \frac{tpfidf_{(t_k,t_i)}}{\sum_{k=1}^{K} tpfidf_{(t_k,t_i)}} \tag{3.3}$$

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

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

$$\begin{aligned} P^{Ia}(t_i) &= \left\{P^{Ia}(t_1|t_i), P^{Ia}(t_2|t_i), \cdots, P^{Ia}(t_k|t_i)\right\} \\ &= \left\{P^{Ia}(t_k|t_i)\right\}_{k=1}^{K} \end{aligned} \tag{3.4}$$

Then, we adapt *Relation Strength Function (RS)* (Chen, Gou, Zhang & Giles 2011) to represent the Intra-coupling similarity within term pairs. RS 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 3.2 *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:*

$$IaR(t_i, t_j) = RS(P^{Ia}(t_i), P^{Ia}(t_j)) \tag{3.5}$$

In this paper, $RS(P^{Ia}(t_i), P^{Ia}(t_j))$ is instantiated to *cosine similarity* to quantify the similarity between $P^{Ia}(t_i)$ and $P^{Ia}(t_j)$, the probability over all term pairs given t_i and t_j respectively. Cosine similarity is widely used in information retrieval and text mining, measuring the cosine of the angle between document vectors, which is regarded as the similarity of two documents in terms of their subject matter. The intra-coupling of term pair (t_i, t_j) is represented as:

$$RS(P^{Ia}(t_i), P^{Ia}(t_j)) = \frac{\sum_{i,j=1}^{K} P^{Ia}(t_i) P^{Ia}(t_j)}{\sqrt{\sum_{i,j=1}^{K} P^{Ia}(t_i)^2} \cdot \sqrt{\sum_{i,j=1}^{K} P^{Ia}(t_j)^2}} \tag{3.6}$$

where its outcome is neatly bounded in $[0, 1]$, due to cosine similarity is used in positive space particularly.

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.

Thus, the procedure of computing intra-term coupling of term pair (t_i, t_j) is summarized in Algorithm 1.

```
Algorithm 1: Intra-coupling Similarity
Input: Document-Term matrix $D$
Output: $IaR(t_i, t_j)$
Construct $M_{tpf}$ (Equation 3.2);
for term $t_i$ in $M_{tpf}$ do
    for Term $t_j$ ($t_j \neq t_i$) in $M_{tpf}$ do
        Calculate $P^{Ia}(t_j|t_i)$ (Equation 3.3);
    end
    Calculate $P^{Ia}(t_i)$ (Equation 3.4);
end
for Term pair $(t_i, t_j)$ ($t_i \neq t_j$) do
    Calculate $IaR(t_i, t_j)$ (Equation 3.5);
end
```

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

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

3.2.2 Inter-couplings between Term Pairs

This section further measures the implicit inter-couplings between term pairs based on the graph theory. A document set is represented as a graph with nodes and edges to reflect the terms and their couplings respectively. The intra-couplings between terms introduced above only captures the explicit couplings of two adjacent nodes in the graph, but fails to consider the couplings of term pairs in a global view, for the reason that it does not take the indirect (implicit) interactions with other terms in the document set into consideration. In this section, we propose an approach to capture this kind of implicit couplings based on graph theory.

Term Pair Frequency Graph

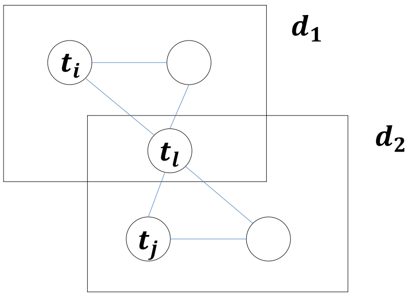


Figure 3.2: The Term Pair Frequency Graph

For example, to cluster or classify a sport article and a document talking about human healthy, we can draw a graph of each document based on their term frequency, then combine the two graphs, there are always some terms used both in the two articles, such as some words of human body function.

As shown in Figure 3.2, the term pair frequency graph G_{tpf} is an ordered pair:

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

compriseing 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, 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 graphs are precisely described as undirected.

Intra-coupling Graph

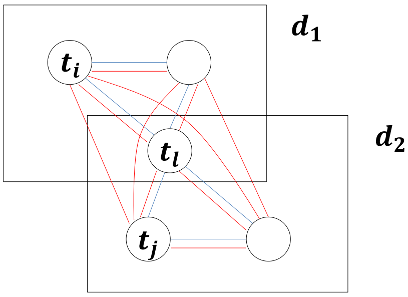


Figure 3.3: The Intra-coupling Graph

Based on the intra-term couplings, in the Figure 3.3, G_{IaR} is constructed to represent terms and their intra-couplings of all term pairs across document collection, 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.

The intra-couplings are represented as an unordered pair of vertexes with respect to a particular edge, and G_{IaR} is a complete graph of G_{tpf}, every two vertexes are related, which means the intra-couplings capture the explicit interactions of all term pairs, including the terms from different documents.

Then the intra-coupling graph of the sport article and the human healthy article is able to represent more relations between the terms that cannot be find in both articles based on the human function words, such as the relation between some professional words of motor skills and human diseases.

However, to reflect the semantic relatedness of term pairs completely, G_{IaR} is fail to provide a reasonable way to consider the influence of all other term pairs. Then how to draw a special "line" to connect them, which means how to find a sensible approach to capture the implicit couplings of them is the main problem solved in following sections.

Inter-coupling Graph

Due to G_{IaR} fails to provide a reasonable way to consider the influence of all other term pairs, thus cannot reflect the couplings of term pairs completely. This triggers the question of how to draw a special “line” to connect them, namely to capture the implicit couplings of term pairs.

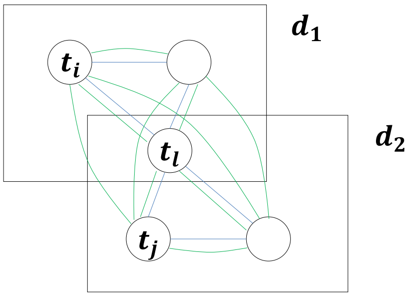


Figure 3.4: The Inter-coupling Graph

Firstly, for a term pair in G_{tpf}, no matter it is connected or not, intuitively, it can be related through other terms, in Figure 3.4, there exist paths starting at t_i and ending at t_j, $t_i \to t_l \to t_j$ for instance. To capture the inter-couplings of term pairs across a document set, no matter how two terms appear separately in different documents or they co-occur in some documents, their interactions with other terms play a major role, i.e the human function words. In other words, we discover *paths* containing other terms to connect every term pair in G_{tpf}. The definition of *path* is given as:

Definition 3.3 *A* **path** *is a subgraph of G_{tpf}, containing a finite sequence of edges which connect a sequence of vertexes,*

$$\begin{aligned} Path(t_i, t_j) = & \big\{(T^P_{i;j}, E^P_{i;j}) \mid t_i, t_{l_1}, \cdots, t_{l_n}, t_j \in T^P_{i;j}, \\ & e_{il_1}, e_{l_1 l_2}, \cdots, e_{l_n j} \in E^P_{i;j},\ t_i \neq t_j, \\ & T^P_{i;j} \subset T,\ E^P_{i;j} \subset E_{tpf},\ n \in [1, \theta]\big\} \end{aligned} \tag{3.7}$$

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.

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.

Secondly, vertexes t_{l_n} between t_i and t_j build a bridge to link them. We further define the n vertexes between two terms t_i and t_j on path $Path(t_i, t_j)$ as **link term set** T_{link}:

$$T_{link} = \big\{t_l \mid t_l \in T^P \backslash (t_i, t_j),\ T^P \in Path(t_i, t_j)\big\} \tag{3.8}$$

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

Furthermore, it is understandable that every term pair in G_{tpf} 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-couplings of every two terms, which is calculated by the couplings of all term pairs on all possible paths between them on G_{tpf}. The detailed algorithm of inter-term coupling is concluded in the following section.

Inter-coupling Similarity

To calculate the inter-coupling similarity between term pairs, we need to concern the RS of every term pair on all possible paths. For this, the *tpf-idf* scheme is adjusted to *term pair occurrence frequency - inverse path frequency* (*tpf-ipf*) to represent the impact of a term pair on paths, defined as:

Definition 3.4 ***tpf-ipf*** *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.*

$$tpfipf((t_i, t_j), d, m) = tpf((t_i, t_j), d) \times log(\frac{m}{pf(t_i, t_j)}) \tag{3.9}$$

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 $\forall (t_k, t_i) \in D(k, i \in [1, K], k \neq i)$,

$$W(t_k|t_i) = \frac{tpfipf_{(t_k,t_i)}}{\sum_{k=1}^{K} tpfipf_{(t_k,t_i)}} \tag{3.10}$$

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:

$$W_{t_{l_1},\cdots,t_{l_n}}(t_k|t_i) = W(t_{l_1}|t_i) \cdot \prod_{p=1}^{n-1} W(t_{l_{p+1}}|t_{l_p}) \cdot W(t_k|t_{l_n}) \tag{3.11}$$

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, longer path goes through more edges, the value of product is smaller, and the weight of 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

$$W_m(t_k|t_i) = \sum_{q=1}^{m} W_q(t_k|t_i) \tag{3.12}$$

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:

$$P^{Ie}(t_k|t_i) = \frac{W_m(t_k|t_i)}{\sum W_m(t_k|t_i)} \tag{3.13}$$

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:

$$\begin{aligned} P^{Ie}(t_i) &= \{P^{Ie}(t_1|t_i), P^{Ie}(t_2|t_i), \cdots, P^{Ie}(t_k|t_i)\} \\ &= \{P^{Ie}(t_k|t_i)\}_{k=1}^{K} \end{aligned} \tag{3.14}$$

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

Definition 3.5 *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, $n \in [1, \theta]$.*

$$IeR_n(t_i, t_j) = RS_n(P^{Ie}(t_i), P^{Ie}(t_j)) \tag{3.15}$$

where $IeR_n(t_i, t_j)$ is the nth order inter-coupling which stands for the RS 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) with the coefficient *exponential decay*,

$$IeR(t_i, t_j) = \frac{\sum\limits_{n=1} exp(1-n) IeR_n(t_i, t_j)}{\sum\limits_{n=1} exp(1-n)} \tag{3.16}$$

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 inter-relate.

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

Algorithm 2: Inter-coupling Similarity

```
Input: Document-Term matrix $D$, User-defined link term quantity $n$
       and threshold $\theta$
Output: $IeR(t_i, t_j)$
Construct $M_{tpf}$;
for term $t_i$ in $M_{tpf}$ do
    for term $t_j (t_j \neq t_i)$ in $M_{tpf}$ do
        Search all possible paths $Path(t_i, t_j)$ with $n$ link terms,
        $n \in [1, \theta]$;
        Compute $P^{Ie}(t_j|t_i)$ (Equation 3.13);
    end
    Compute $P^{Ie}(t_i)$ (Equation 3.14);
end
for term pair $(t_i, t_j)(t_i \neq t_j)$ do
    Compute $IeR(t_i, t_j)$ (Equations 3.15 & 3.16);
end
```

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

3.2.3 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 3.6 *Given a document set D, the* ***Semantic Coupling Similarity (SCS)*** *of a term pair (t_i, t_j) in D is:*

$$SCS(t_i, t_j) = (1 - \alpha) \cdot IaR(t_i, t_j) + \alpha \cdot IeR(t_i, t_j) \tag{3.17}$$

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 coupled similarity of term pairs reaches the highest value 1 when the terms have identical meaning, which means the distance between them is zero.

Property 2: Symmetrical Property

On the undirected graphs G_{IaR} and G_{IeR}, there is only one type of relatedness for term pairs on each graph, then the order is disregarded, so that the coupled similarity for term pairs is symmetrical.

Property 3: Positive Property

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

Property 4: Minimal Distance Property

Early edge-based model of semantic relatedness assumes that the semantic distance is based on the number of edges between terms (Rada et al. 1989), 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}, 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-coupling 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.

3.3 Coupled Document Analysis

We here apply our SCS measure: the semantic coupling similarity of term pairs, to analyze documents by capturing the semantic related documents.

After an optimal combination of intra- and inter-term coupling, a new coupled similarity graph G_{SCS} is drawn as the integration of G_{IaR} and G_{IeR}, it can be transferred into a $K \times K$ coupled similarity matrix M_{cou}, whose elements reflect the couplings of each term pair as follows:

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

Firstly, each document is defined as the mapping:

$$\phi : d \rightarrow \phi(d) = \big\{P(t_1, d), P(t_2, d), \cdots, P(t_k, d)\big\} \tag{3.18}$$

where $P(t_k, d) = \frac{tf(t_k,d)}{\sum_{k=1}^{K} tf(t_k,d)}$ is the probability of term t_k in document d.

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

$$\widetilde{\phi}(d) = \phi(d)M_{cou} \tag{3.19}$$

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

$$Sim(d_i, d_j) = \phi(d_i)M_{cou}M_{cou}^T\phi(d_j)^T \tag{3.20}$$

Thus, the new document representation $\widetilde{\phi}(d)$ is computed efficiently directly from the original data using Equation 3.19, documents are represented in new coupled semantic feature space based on the term occurrence frequency pattern and comprehensive term pair couplings.

$\widetilde{\phi}(d)$ 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 complete and average term linkages, which measure the *cosine similarity* between two clusters based on the average and minimum of their document similarities, respectively.

3.4 Experiments and Evaluation

In this section, SCS is incorporated into HAC as CHAC with both complete and average linkages, evaluating the CHAC performance in terms of the impact of inter-coupling, by considering three scenarios: "no link term", "one link term", "two link terms" and "three link terms". Then, 5-fold cross-validation is employed to present parameter tuning and automatically estimate the optimal value of parameter α in Equation 3.17 on various data sets. Finally, we compare our best performance with similar and typical document representations.

3.4.1 Experimental Settings

Data Sets

Three most popular text data sets are chosen: Reuters-21578[1], TDT2[1] and WebKB[2].

- The documents in the Reuters-21578 collection appeared on the Reuters newswire in 1987. It contains 21578 documents in 135 categories. After preprocessing, this corpus contains 18933 distinct terms with 65 classes.

- The Nist Topic Detection and Tracking corpus (TDT2) consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI) and 2 television programs (CNN, ABC). It consists of 11201 on-topic documents which are classified into 96 semantic categories. In this subset, those documents appearing in two or more categories were removed, and only the largest 30 categories were kept, thus leaving us with 9,394 documents in total.

- WebKB data set contains WWW-pages collected from computer science departments of various universities in January 1997 by the World Wide Knowledge Base project of the CMU text learning group. The 8,282 pages were manually classified into the following 7 categories:

[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/

Table 3.1: Characteristics of Data Sets

Data Sets	n	m	m_{doc}	k	n_{class}
re1	7085	8933	42	8	886
re2	6387	4312	37	4	1597
re3	6632	4563	38	4	1658
td1	5476	8000	118	7	782
td2	5476	8000	138	5	1095
td3	5476	8000	118	7	782
td4	5476	8000	119	7	780
w1	4087	7770	79	4	1022
w2	3268	7770	78	4	817
w3	3268	7770	80	4	817

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

student (1641), faculty (1124), staff (137), department (182), course (930), project (504), other (3764).

In our experiments, *re1*, *re2* and *re3* are subsets of Reuters-21578, *td1*, *td2*, *td3* and *td4* are subsets of TDT2, *w1*, *w2* and *w3* are subsets of WebKB data. Detailed information of 10 data sets are summarized in Table 3.1.

Evaluation Metrics

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

- *Purity* is an external evaluation criterion for cluster quality, each cluster is assigned to the most frequent class, then the accuracy of the assignment is measured by:

$$purity(\Omega, C) = \frac{1}{N}\sum_{k} max_j|\omega_k \cap c_j|$$

where $\Omega = \{\omega_1, \omega_2, \cdots, \omega_k\}$ is the set of clusters and $C = \{c_1, c_2, \cdots, c_j\}$ is the set of classes, k is the number of clusters and N is the number of documents.

- *Rand Index* measures the percentage of decisions that are correct, it penalizes both false positive and false negative decisions during clustering.
$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$
where TP, TN, FP, FN stand for true positive, true negative, false positive and false negative, respectively.

- F_1 *measure* considers precision and recall in evaluation of clustering, supports different weighting of false positive and false negative errors.
$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$
where $precision = TP/(TP + FP)$, $recall = TP/(TP + FN)$.

- *Normalized Mutual Information* is a popular information-theoretically interpreted metric for evaluating clustering quality, it trades off the quality of the clustering against the number of clusters, due to high Purity is easy to achieve when the number of clusters is large.
$$NMI(\Omega, C) = \frac{I(\Omega; C)}{[H(\Omega) + H(C)]/2}$$
where I is mutual information, $H(\Omega)$ and $H(C)$ is the entropy of Ω and C.

3.4.2 Inter-coupling Ordering

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

The inter-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, 2nd and 3rd order inter-coupling on ten data sets.

In Table 3.2 & 3.3, we compare the impact of inter-coupling ordering in terms of four clustering evaluation metrics on the selected data sets, the results of best performance are bold. Overall, for every evaluation metric, all ten data sets are sensitive to the inter-coupling ordering, and achieve remarkable improvements compared to the performance based on intra-coupling only. In addition, in 20 experiments, there are 17 experiments show that we achieve better performance with the import of inter-coupling no matter how many link terms are involved.

The performance of CHAC on a particular order of inter-coupling has been greatly improved compared to the 0 order inter-coupling, the reason is, when no link term exists, couplings 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. However, the trends on the higher order of inter-coupling are not so remarkable, or even start to descend, but still better than 0 ordering, which reflect that more link terms and longer path will result in weaker indirect influence of term couplings.

Consider there are 15 experiments prove that the inter-coupling with 1st or 2nd ordering achieve best performance, and time complexity, we recommend that SCS on as far as 2nd order of Inter-coupling is likely acceptable to our need.

3.4.3 Tuning Parameter α

As the parameter α controls the effect of intra- and inter-couplings, it is essential to optimize α to achieve the best possible performance. We ran the experiments using different value of α, and the values that achieve the best performance are chosen as the optimal values. By exploiting 5-fold cross-validation, the value of α is automatically estimated. The Purity scores calculated from each fold are averaged to reflect the performance of clustering on testing sets.

5-fold cross-validation is employed in our experiments to estimate the optimal value of α, the original data set is randomly partitioned into 5 equal sized subsets. Of the 5 subsets, a single subset is retained as the validation data for testing the model, and the remaining 4 subsets are used as training data. The cross-validation process is then repeated 5 times (the folds), with each of the 5 subsets used exactly once as the validation data. The 5 results from the folds are then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. The 5-fold cross-validation is helpful to estimate how accurately a predictive model will perform in practice and limit the problems like overfitting.

For each data set, the automatic selection of α equals 0.40, 0.05, 0.20, 0.45, 0.20, 0.25, 0.15, 0.10, 0.10 and 0.25 for complete-link CHAC, and the corresponding Purity scores are 0.8051, 0.8876, 0.8859, 0.8425, 0.8898, 0.8202, 0.8112, 0.6511, 0.6561 and 0.6365. For average-link CHAC, α equals 0.25, 0.50, 0.15, 0.35, 0.15, 0.25, 0.30, 0.05, 0.10 and 0.40, the corresponding RI scores are 0.8264, 0.8863, 0.8926, 0.8699, 0.9126, 0.8646, 0.8651, 0.5806, 0.6334 and 0.5624 respectively.

The growth trends of Purity 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 3.5 & 3.6. In particular, for every data set, the trend keeps growing from the beginning, then starts to descend after it reaches the peak, it shows that the Purity results achieve the best performance at a peak point with respect to a certain value of α, and for different value of α, there are considerable difference between the validation performance. In addition, the optimal value of α is always achieved between 0 and 1 for all data sets, much better than both ends of the diagrams. This proves that the combination of intra- and Inter-coupling similarity achieves better performance than using Intra-coupling only (when $\alpha = 0$) or Inter-coupling only (when $\alpha = 1$).

3.4.4 Experimental Results

CHAC is compared with Bag of Words (BOW) , LSA (Deerwester et al. 1990), LDA (Blei et al. 2003) and CRM (Cheng et al. 2013). 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. 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. The best performance of CHAC demonstrated in Table 3.2 & 3.3 with an optimal value of α are selected to compare with other typical document representation models with respect to Purity, RI, F_1 measure and NMI scores on ten data sets.

The technical performance for different document representation models on testing data is evaluated and concluded in Table 3.4 & Table 3.5. Specifically, for each model, each cell illustrates the practical HAC with complete or average linkage results considering various evaluation metrics. For each evaluation metric, a larger value indicates a more accurate and reliable model. We present CHAC scores with bold face when it achieves best performance over all models.

Obviously, CHAC on both average and complete linkages achieves great improvement and outperforms majority models by considering the given clustering evaluation criteria on various data sets. CHAC significantly improve the performance over the traditional model, like BOW and LSA; it also achieve considerable improvement comparing with more advanced and complicated document representation models, such as GVSM and LDA; considering CRM, a model which calculate term relation through link terms as well, CHAC is better than it on all data sets with respect to all evaluation metrics.

In specific, for the results of different model using HAC with complete linkage in Table 3.4, CHAC is the best model which achieve highest score on both four clustering metrics over 8 data sets; for the results of HAC with average linkage in Table 3.5, CHAC is competitive over 6 data sets.

The reason lies in that SCS offers a deeper way to capture the semantic relations of term pairs. Unlike BOW, LSA and LDA methods which overlook the internal interactions between terms, SCS accomplishes a comprehensive consideration of not only the intra- (explicit) couplings which is captured via term co-occurrence frequency patterns, but also the effect of inter- (implicit) 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.

3.5 Conclusions

We have proposed a semantic coupling similarity (SCS) measure to comprehensively capture the coupling relationships both within and between term pairs in a document through representing term couplings as a term linkage graph and considering probabilistic distributions of terms and term couplings. A document set is then represented as a term coupling vector for document analysis.

SCS achieves this in terms of a four-step procedure:

1. captures the semantic intra-coupling of term pairs based on its occurrence frequency information across a document set;

2. capture the semantic inter-coupling 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 full coupled semantic similarity of term pairs is achieved;

4. the original document set can then be represented by a coupled semantic 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, inter-coupling of term pairs plays an important role in comprehensively capturing semantic couplings. For this, it is essential to tune the weight between intra- and inter-coupling of term pairs across documents. In addition, experimental performance illustrates that SCS is a path-length-sensitive model, although a path showing term linkage could be quite long, our comprehensive test shows that we may only need as far as two steps of term linkage for most of cases for an acceptable level of running time.

This research opens new opportunities to deeply explore semantic similarity, our further research efforts include: first, all relatedness is built on term pairs in our approach to avoid the polysemy of every single term; however, document representation constructed by term pairs will expand the feature space, which will result in low efficiency. A sensible way needs to be identified to project documents to a new and smaller space. Second, we are also introducing the coupled idea into the calculation of document pair relatedness. Finally, the time complexity brought by the increase of link terms also needs further improvement.

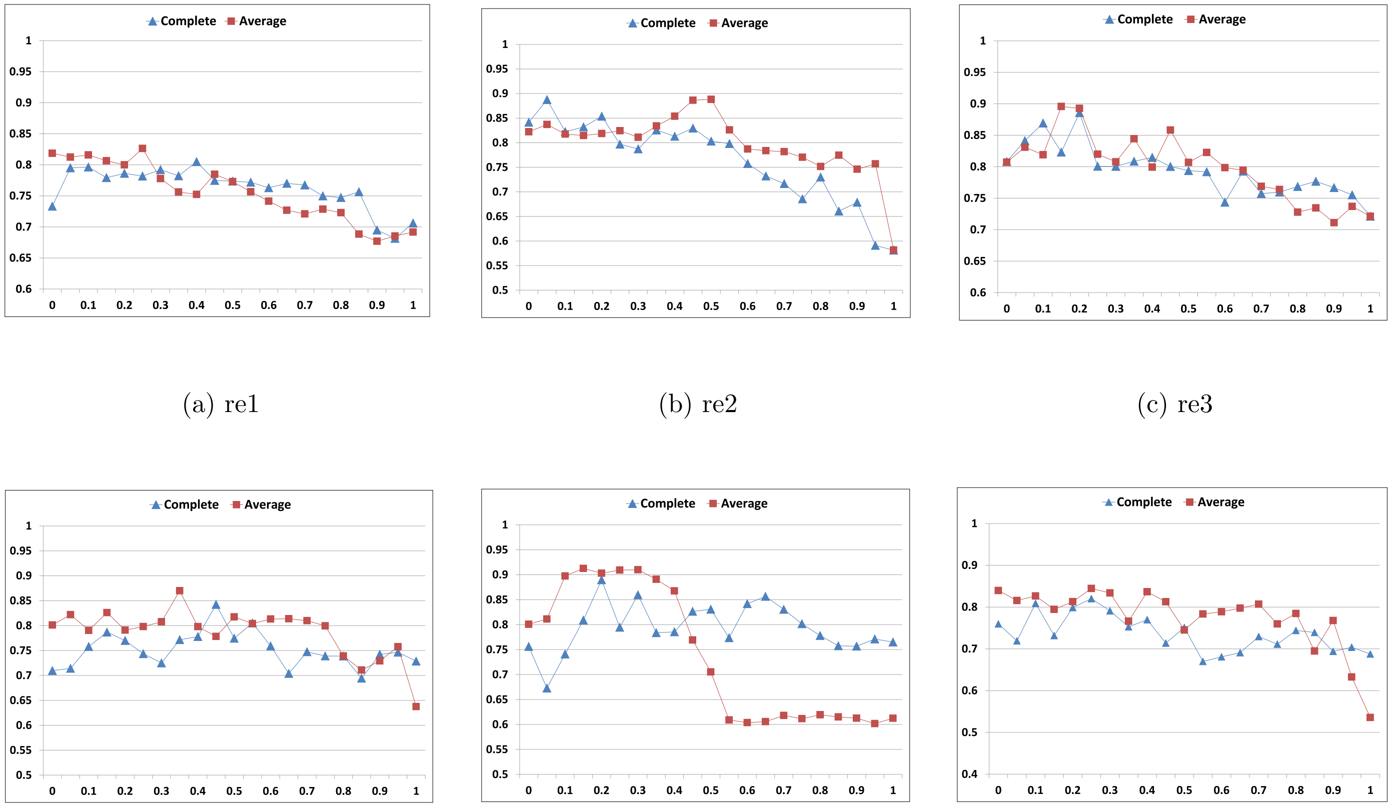
We are working on theoretical analysis of the effect of the number of link terms, and comparing SCS with the most recent machine learning methods for latent semantic analysis and document classification.

Table 3.2: The Impact of Inter-coupling Ordering Using HAC with Complete Linkage

Data Sets	Ordering	Purity	RI	F_1 measure	NMI
re1	0	0.7332	0.6930	0.4433	0.4128
	1	0.7927	0.7401	0.5163	0.4929
	2	**0.8051**	**0.8437**	**0.7395**	**0.5946**
	3	0.7910	0.7345	0.5284	0.4855
re2	0	0.8414	0.7410	0.6835	0.4902
	1	**0.8876**	**0.8089**	**0.7516**	**0.5885**
	2	0.8868	0.7833	0.7341	0.5741
	3	0.8751	0.7739	0.7061	0.5540
re3	0	0.8081	0.7279	0.6268	0.4997
	1	0.8691	0.8254	0.7568	0.5910
	2	**0.8859**	**0.8261**	**0.7591**	**0.5923**
	3	0.8593	0.8122	0.7397	0.5669
td1	0	0.7096	0.8137	0.5440	0.5858
	1	**0.8425**	**0.8957**	**0.7395**	**0.7326**
	2	0.8095	0.8745	0.6899	0.6999
	3	0.7922	0.8593	0.6777	0.6859
td2	0	0.7563	0.7620	0.6026	0.6265
	1	**0.8898**	**0.9305**	**0.8531**	**0.7730**
	2	0.8858	0.8927	0.7916	0.7728
	3	0.8791	0.9004	0.8138	0.7703
td3	0	0.7597	0.8320	0.5913	0.5940
	1	**0.8202**	**0.9003**	**0.7594**	**0.7188**
	2	0.7980	0.8694	0.6789	0.6830
	3	0.8041	0.8494	0.6701	0.6620
td4	0	0.7456	0.8049	0.5688	0.5901
	1	**0.8112**	**0.8997**	**0.7496**	**0.6936**
	2	0.8072	0.8684	0.6798	0.6755
	3	0.8015	0.8386	0.6283	0.6498
w1	0	0.5224	0.5768	0.4928	0.3037
	1	0.6462	0.6760	0.5063	0.3550
	2	0.6007	0.6477	0.4638	0.3315
	3	**0.6511**	**0.6946**	**0.5019**	**0.3667**
w2	0	0.5615	0.5525	0.5264	0.3557
	1	0.6197	0.6570	0.4906	0.3200
	2	0.6365	0.6761	0.5065	0.3422
	3	**0.6561**	**0.6985**	**0.5221**	**0.3615**
w3	0	0.5557	0.6003	0.4694	0.3266
	1	0.6098	0.6311	0.4798	0.3193
	2	0.6102	0.6424	0.4830	0.3265
	3	**0.6365**	**0.6910**	**0.5082**	**0.3436**

Table 3.3: The Impact of Inter-coupling Ordering Using HAC with Average Linkage

Data Sets	Ordering	Purity	RI	F_1 measure	NMI
re1	0	0.8186	0.7633	0.5794	0.5172
	1	0.8136	0.7613	0.5729	0.5215
	2	**0.8264**	**0.7699**	**0.7826**	**0.5408**
	3	0.8127	0.7641	0.5751	0.5161
re2	0	0.8221	0.7551	0.7160	0.5749
	1	0.8863	0.7973	0.7474	0.5775
	2	**0.8863**	**0.8038**	**0.7504**	**0.5813**
	3	0.8812	0.7804	0.7320	0.5697
re3	0	0.8872	0.8245	0.7534	0.6016
	1	0.8393	0.7915	0.7283	0.5888
	2	0.8862	0.8261	0.7586	0.5995
	3	**0.8926**	**0.8263**	**0.7588**	**0.6137**
td1	0	0.8011	0.9034	0.7458	0.7412
	1	0.8629	0.9105	0.7964	0.7739
	2	**0.8699**	**0.9197**	**0.8209**	**0.7936**
	3	0.8548	0.9167	0.8161	0.7825
td2	0	0.9008	0.9268	0.8618	0.8213
	1	0.9104	0.9392	0.8840	0.8317
	2	**0.9126**	**0.9408**	**0.8866**	**0.8499**
	3	0.9071	0.9324	0.8712	0.8353
td3	0	0.8396	0.9185	0.8102	0.7570
	1	0.8521	0.9040	0.7915	0.7659
	2	**0.8646**	**0.9345**	**0.8430**	**0.7730**
	3	0.8506	0.9057	0.7951	0.7621
td4	0	0.8420	0.9063	0.7961	0.7618
	1	0.8526	0.9006	0.7747	0.7490
	2	**0.8651**	**0.9225**	**0.8280**	**0.8018**
	3	0.8383	0.9318	0.8354	0.7685
w1	0	0.5596	0.5812	0.4437	0.2919
	1	0.5789	0.5871	0.5633	0.4010
	2	0.5804	0.5986	0.5670	0.3895
	3	**0.5806**	**0.5992**	**0.5671**	**0.3942**
w2	0	0.4801	0.4125	0.4662	0.1858
	1	0.5719	0.5738	0.5506	0.3706
	2	**0.6334**	**0.6942**	**0.5582**	**0.3800**
	3	0.5765	0.5930	0.5586	0.3780
w3	0	0.5318	0.5003	0.5061	0.2910
	1	0.5588	0.5553	0.5364	0.3459
	2	**0.5624**	**0.5540**	**0.5402**	**0.3650**
	3	0.5569	0.5482	0.5355	0.3492



(a) re1

(b) re2

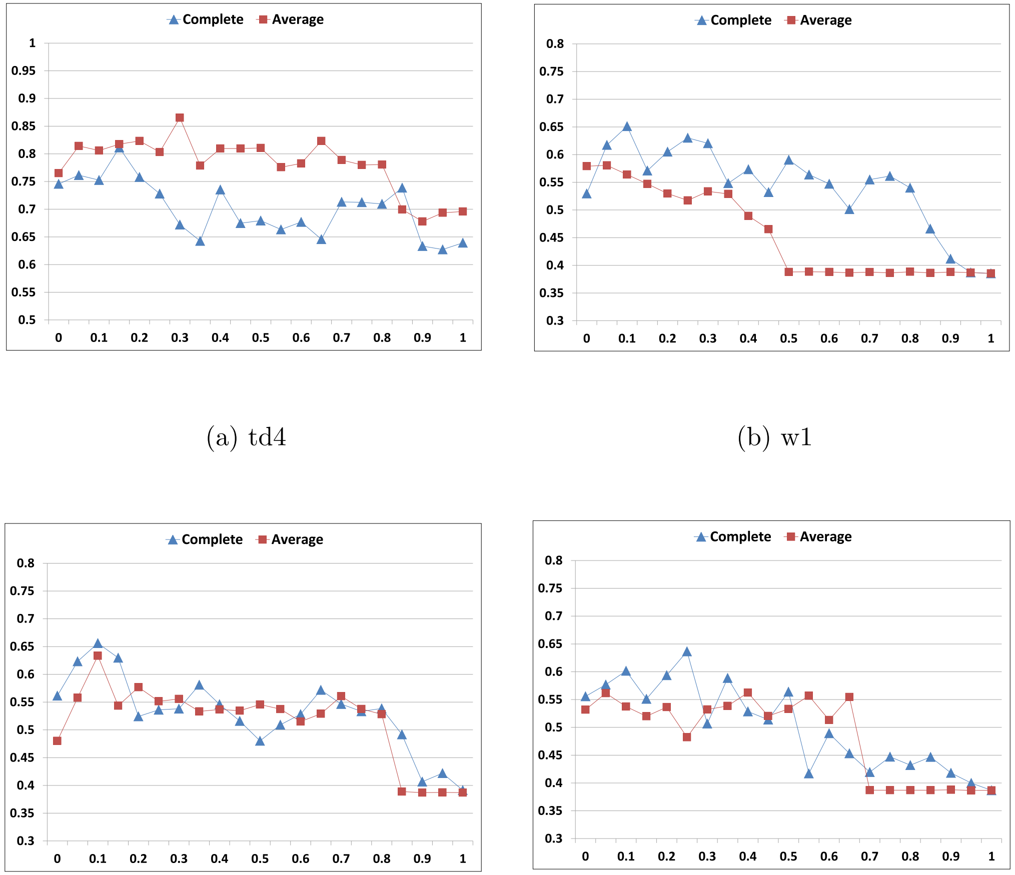
(c) re3

(d) td1

(e) td2

(f) td3

Figure 3.5: Tuning of α (1)



(a) td4

(b) w1

(c) w2

(d) w3

Figure 3.6: Tuning of α (2)

Table 3.4: Results of Different Model Using HAC with Complete Linkage

Model \ Data sets	re1	re2	re3	td1	td2	td3	td4	w1	w2	w3
Purity										
BOW	0.5668	0.6133	0.6175	0.2922	0.3314	0.3010	0.2887	0.3969	0.3856	0.4079
GVSM	0.7403	0.8052	0.8239	0.8263	0.8825	0.8790	0.8460	0.4125	0.4905	0.4535
LSA	0.5268	0.6103	0.5749	0.6090	0.5487	0.5180	0.5262	0.3854	0.3856	0.3917
LDA	0.8038	0.7115	0.8332	0.8362	0.8521	0.8888	0.8402	0.5055	0.5499	0.5756
CRM	0.6100	0.6219	0.6642	0.4670	0.6015	0.4353	0.4272	0.3881	0.3911	0.4168
CHAC	**0.8051**	**0.8876**	**0.8859**	**0.8425**	**0.8898**	0.8202	0.8112	**0.6511**	**0.6561**	**0.6365**
RI										
BOW	0.4470	0.5086	0.5395	0.2362	0.3013	0.2450	0.2346	0.3967	0.3153	0.3492
GVSM	0.7767	0.7511	0.7916	0.8788	0.9111	0.9277	0.9047	0.3322	0.4341	0.3924
LSA	0.4640	0.5412	0.5928	0.7330	0.6623	0.6546	0.6535	0.5091	0.4905	0.4641
LDA	0.8435	0.6361	0.8100	0.8850	0.8977	0.9367	0.8836	0.5589	0.5702	0.6121
CRM	0.6282	0.5820	0.6364	0.6482	0.7131	0.6641	0.6462	0.6012	0.6067	0.5921
CHAC	**0.8437**	**0.8089**	**0.8261**	**0.8957**	**0.9305**	0.9003	0.8997	**0.6946**	**0.6985**	**0.6910**
F1-measure										
BOW	0.5146	0.6420	0.6347	0.3342	0.4009	0.3269	0.3241	0.4125	0.4330	0.4433
GVSM	0.6772	0.6856	0.7079	0.7379	0.8294	0.8127	0.7036	0.4391	0.4726	0.4497
LSA	0.4906	0.5654	0.4913	0.4671	0.4351	0.4556	0.4396	0.3529	0.3721	0.3846
LDA	0.7323	0.6373	0.7546	0.6792	0.7945	0.8338	0.6868	0.4519	0.5174	0.4725
CRM	0.3231	0.5812	0.5471	0.3563	0.4619	0.3447	0.3147	0.3051	0.3447	0.3036
CHAC	**0.7395**	**0.7516**	**0.7591**	**0.7395**	**0.8531**	0.7594	0.7496	**0.5019**	**0.5221**	**0.5082**
NMI										
BOW	0.1795	0.1813	0.2477	0.2551	0.2096	0.2522	0.2381	0.1212	0.1099	0.1348
GVSM	0.5301	0.4778	0.5717	0.7043	0.7724	0.7690	0.7218	0.1514	0.2014	0.1187
LSA	0.1371	0.1561	0.1410	0.3377	0.2658	0.3600	0.3560	0.1031	0.1034	0.1062
LDA	0.5744	0.3304	0.5745	0.7040	0.7316	0.7883	0.7367	0.1891	0.2680	0.2260
CRM	0.4910	0.4881	0.4839	0.5659	0.5279	0.5524	0.5311	0.2149	0.2062	0.2235
CHAC	**0.5946**	**0.5885**	**0.5923**	**0.7326**	**0.7730**	0.7188	0.6936	**0.3667**	**0.3615**	**0.3436**

Table 3.5: Results of Different Model Using HAC With Average Linkage

Model \ Data sets	re1	re2	re3	td1	td2	td3	td4	w1	w2	w3
Purity										
BOW	0.5303	0.5831	0.5620	0.3859	0.3004	0.3862	0.3867	0.3859	0.3865	0.3862
GVSM	0.7715	0.8580	0.7886	0.7646	0.6033	0.7674	0.7663	0.3859	0.3871	0.3862
LSA	0.5265	0.5818	0.5639	0.2712	0.4008	0.2716	0.2711	0.3876	0.3856	0.3856
LDA	0.7140	0.8589	0.6439	0.9242	0.9571	0.9415	0.9159	0.5366	0.5474	0.5404
CRM	0.6237	0.7136	0.6968	0.4408	0.5875	0.3532	0.4494	0.3876	0.3859	0.3886
CHAC	**0.8264**	**0.8863**	**0.8926**	0.8699	0.9126	0.8646	0.8651	**0.5806**	**0.6334**	**0.5624**
RI										
BOW	0.3729	0.4473	0.4167	0.4097	0.2412	0.4073	0.4095	0.2871	0.2873	0.2873
GVSM	0.7356	0.7507	0.7898	0.8591	0.7090	0.8606	0.8603	0.2868	0.2879	0.2870
LSA	0.4325	0.4588	0.4277	0.2051	0.4234	0.2032	0.2031	0.3001	0.2994	0.3139
LDA	0.7231	0.7951	0.6301	0.9518	0.9577	0.9593	0.9381	0.5211	0.5397	0.5172
CRM	0.6125	0.6434	0.6495	0.5961	0.6653	0.4015	0.6033	0.3130	0.2875	0.2925
CHAC	**0.7699**	**0.8038**	**0.8263**	0.9197	0.9408	0.9345	0.9225	**0.5992**	**0.6942**	**0.5540**
F1-measure										
BOW	0.5364	0.6169	0.5866	0.3996	0.3874	0.4005	0.4007	0.4448	0.4450	0.4447
GVSM	0.7815	0.7318	0.7455	0.7036	0.5961	0.7062	0.7041	0.4448	0.4449	0.4448
LSA	0.5132	0.6160	0.5849	0.3330	0.4396	0.3337	0.3336	0.4415	0.4424	0.4340
LDA	0.5818	0.6634	0.6751	0.8749	0.9118	0.8969	0.8427	0.5126	0.5112	0.4930
CRM	0.5358	0.6226	0.6105	0.4094	0.4996	0.3671	0.4140	0.4385	0.4444	0.4446
CHAC	**0.7826**	**0.7504**	**0.7588**	0.8209	0.8866	0.8430	0.8280	**0.5671**	**0.5582**	**0.5402**
NMI										
BOW	0.1197	0.1040	0.1045	0.3182	0.3016	0.3216	0.3222	0.1894	0.2571	0.2079
GVSM	0.5298	0.5671	0.4846	0.6403	0.5504	0.6507	0.6414	0.1696	0.3753	0.2119
LSA	0.3557	0.2018	0.1479	0.2051	0.2502	0.2035	0.2546	0.1441	0.1700	0.1627
LDA	0.5140	0.4932	0.4832	0.8376	0.8628	0.8441	0.8038	0.2820	0.2769	0.2410
CRM	0.1747	0.2556	0.2058	0.2283	0.3370	0.1531	0.2426	0.2427	0.2070	0.2444
CHAC	**0.5408**	**0.5813**	**0.6137**	0.7936	0.8499	0.7730	0.8018	**0.3942**	**0.3800**	**0.3650**

Chapter 4

Semantic Representation Using Hierarchical Tree Augmented Naive Bayes

4.1 Introduction

Classification is one of the basic problems of data mining. Currently, there is a diverse range of classification methods, including decision tree classification, the support vector machine, neural network classifiers, etc. Among them, Bayesian classifiers based on the probability theory have received considerable attention in recent years.

According to the Bayes school of thoughts, in the absence of any observations, our knowledge is represented by a prior distribution. We then update the prior distribution based on an observation of attributes in terms of the posterior probability, which is the probability that the attribute belongs to a class. The Bayesian classifiers, select the class having the largest posterior probability as the class that the attribute belongs to.

In the Bayesian classifiers family, the naive Bayes classifier attracts a lot of attention because of its simple implementation and good performance, but its independence requirement between attributes nevertheless limits its scope. This limitation leads to more evident flaws when it is applied to text classification. Document representation based on naive Bayes treats attributes independently given the class label, that is to say, that it treats the terms *document* and *representation* in the text mining articles and library management articles equally. However, empirically, they rarely co-occur in library management articles and there is a certain dependence which exists between them in the text mining articles.

Compared with general classification problems, text classification faces the unavoidable problem that there might be a certain dependence among attributes. A great amount of extended structure based on pure naive Bayes have been proposed to relax the strong independence assumption. This will allow extra dependencies between attributes (Friedman, Geiger & Goldszmidt 1997) (Rubio & Gámez 2011) , embedding them with other classification models (Kohavi 1996) (Frank, Hall & Pfahringer 2002) (Jiang, Zhang & Su 2005), or using local data learning models(Webb, Boughton & Wang 2005) (Zhang, Jiang & Su 2005), which will further improve classification performance and maintain computation simplicity. However, as text classification is a special kind of classification, representing documents as terms and pairwise correlations is still not enough to fully capture the meaning of documents.

The reasons for this lie in (1) the term independence assumption is not necessarily correct in practice, but relaxing the independence assumption or even considering each pairwise term dependence as a complete graph may be problematic owing to the high computational cost; (2) representing documents as terms and pairwise correlations is still lacking in terms of the dependence of term pairs. Furthermore, high order semantics are overlooked.

For example, term pairs *document representation* and *machine learning* co-occur frequently in text mining articles. *Bio informatics* and *machine learning* may be easily found in the articles about biological sciences, however, single-layer Bayesian networks cannot find the dependence between term pairs.

The proposed method addresses the above issues by constructing a hierarchical tree-like structure to extract highly correlated terms in a layerwise fashion while pruning weak correlations to keep efficiency. We propose a **Hierarchical Tree Learning** method. There are three main contributions that our work makes to the field:

- A hierarchical tree structure with hierarchical feature extraction and a correlation computation procedure. Highly correlated terms are merged into sets and this is associated with more complete semantic information.

- Through the means of a hierarchical tree structure, features are turned into more comprehensively-connected term sets and these carry more semantic information than that of a single term. Moreover, it is able to avoid word sense ambiguity.

- Each layer is a maximal weighted spanning tree to prune weak feature correlations, which, in turn, leads to improvements in the efficiency of the implementation.

- It can be applied to wide range of applications, including both supervised and unsupervised learning approaches. In this chapter, we associate the tree with TAN as Hierarchical Tree Augmented Naive Bayes (HTAN).

The hierarchical tree is able to capture the dependence between terms, term pairs, or even term sets. The higher the order of the tree, the more semantics it can carry. The hierarchical tree is much closer to a human understanding of texts, grouping them into different classes by comprehending the topics and contents of the texts.

The remainder of this chapter is organised as follows: Section 2 reviews and evaluates the related work of Bayesian text classifier measures. Section 3 proposes the hierarchical tree learning procedure and its application in

HTAN. Finally, a conclusion and future work recommendations are provided in Section 4.

4.2 Bayes Classification Methods

Text Classification (alternatively *Text Categorization*) concerns the task of labelling natural language texts $D = \{d_1, d_2, \cdots, d_{|D|}\}$ with thematic categories $|C|$ from a predefined set $C = \{c_1, c_2, \cdots, c_{|C|}\}$. Generally, text categorization approaches have two steps,

- The learning step, where a text classifier is trained, it produces a classification function $F : D \to C$ that maps labeled training documents to categories;
- The classification step, where the classifier is used to predict class labels for testing data.

Many standard machine learning techniques have been applied to automated text categorization problems, such as Bayes classifiers, decision tree classifiers, support vector machines, neural networks, regression methods, on-line methods and so on so forth(Peng, Schuurmans & Wang 2004). Among them, Bayes classifier and its extensions have shown the surprising performance in text classification domain due to their simple and efficient implements(Rennie, Shih, Teevan, Karger et al. 2003).

Normally, for the training set $U = \{D, C\}$, $D = \{d_1, d_2, \cdots, d_{|D|}\}$, $C = \{c_1, c_2, \cdots, c_{|C|}\}$, $c_j \in C$ is the category label of document $d_i \in D$, the inductive construction of Bayes learning methods is defined in terms of the posterior probability as an application of Bayes' theorem:

$$P(c_j|d_i) = \frac{P(c_j)P(d_i|c_j)}{P(d_i)} \tag{4.1}$$

where the posterior probability $P(c_j|d_i)$ is the probability that document d_i belongs to category c_j.

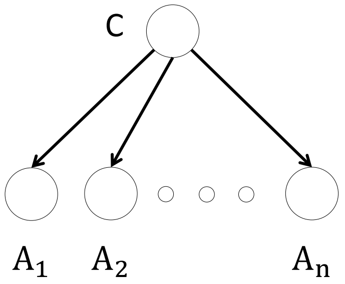


Figure 4.1: The structure of the naive Bayes network

Posterior probability is a composition of a likelihood function $P(d_i|c_j)$ and a prior probability $P(c_j)$, and $P(d_i)$ is constant for all classes, then only $P(c_j)P(d_i|c_j)$ needs to be estimated, Equation (1) is rewritten as

$$P(c_j|d_i) = P(c_j)P(d_i|c_j) \tag{4.2}$$

Based on *Maximum a Posterior* (MAP) hypothesis, the classifier will predict that d_i belongs to c_j which has the highest posterior probability conditioned on d_i

$$\begin{aligned} c^* &= \arg\max_{c \in C}\{P(c_j|d_i)\} \\ &= \arg\max_{c \in C}\{P(c_j)P(d_i|c_j)\} \end{aligned} \tag{4.3}$$

Then the classier is constructed by seeking the optimal category which maximize the posterior probability.

4.2.1 Naive Bayes Classifier

In text categorization, documents are normally represented based on vector space models in Bayes classifier, every term is a feature and every document is represented as a vector $d_i = \{t_{1i}, t_{2i}, \cdots, t_{\tau i}\}$ of weighted terms, which gives data with many attributes, it is expensive to calculate $P(d_i|c_j)$ since the space of possible documents d_i is vast. To simplify calculation, it is common to make the naive assumption of "*class-conditional independence*", which means all attributes t_{ki} are statistically independent of each other given the value of the category label c_j. This independence assumption is encoded as

$$P(d_i|c_j) = \prod_{k=1}^{\tau} P(t_{ki}|c_j) \tag{4.4}$$

Probabilistic classifiers that adjust this assumption are called *Naive Bayes* classifiers, which has the simple structure depicted in Figure 4.1, that predict the category with the highest posterior probability, simplify computation by reducing Equation 4.3 to

$$c^* = \arg\max_{c \in C} \left\{ P(c_j) \prod_{k=1}^{\tau} P(t_{ki}|c_j) \right\} \tag{4.5}$$

Text categorization approaches under such assumption, the learning step is to estimate the conditional probability of each feature given a category; and then the classification step is to determine the category which testing document belongs according to these conditional probabilities(Hong-Bo, Zhi-Hai, Hou-Kuan & Li-Ping 2002).

Naive Bayes classifiers are still frequently used in some literature due to its low complexity and easy implementation (Rennie et al. 2003), in the meantime, they perform surprisingly well in some classification problems(Frank et al. 2002). However, it is unrealistic since there might be certain dependence among terms across various documents, the "naive assumption" is obviously not verified in practice.

There are a great amount of approaches had been proposed to augment the performance of pure naive Bayes classifiers that account for relaxing the strong independence assumption and still keep the computational advantages of efficiency, e.g. Semi-naive Bayes (Kononenko 1991), which increases dependencies by using clusters of variables instead of single ones; Tree Augmented Naive Bayes (TAN) (Friedman et al. 1997) and k-Dependence Bayesian Network (KDB) (Rubio & Gámez 2011) allow extra dependencies between predictive attributes by adding augmenting edges; also, it has been observed that the naive Bayes especially fit to be a local model embedded into another model, such as a decision tree or a k-nearest neighbor, corresponding methods are Naive Bayes Tree (NBTree) (Kohavi 1996), Locally Weighted Naive Bayes (LWNB) (Frank et al. 2002) and Instance Cloning Local Naive Bayes (ICLNB) (Jiang et al. 2005). There are also approaches consider the influence of attributes which avoid problem of structure learning, like Averaged One-Dependence Estimators (AODE) (Webb et al. 2005) and Hidden Naive Bayes (Zhang et al. 2005).

4.2.2 Tree Augmented Naive Bayes Classifier

A popular extension of naive Bayes classifier that relaxing the independence assumption is based on a tree-like structure Bayesian network, called *Tree Augmented Naive Bayes Classifier* (TAN) (Friedman et al. 1997). It considers the correlations among attributes by allowing additional augmenting edges, for each attribute, it has as parents the class variable and at most one other attribute.

The Construct-TAN Procedure

More precisely, let $U = \{D, C\}$, $D = \{d_1, d_2, \cdots, d_{|D|}\}$, $C = \{c_1, c_2, \cdots, c_{|C|}\}$, $c_j \in C$ is the category label of document $d_i \in D$, each document is represent as a vector $d_i = \{t_{1i}, t_{2i}, \cdots, t_{\tau i}\}$. In both Naive Bayes and TAN classifiers, the class variable is the root, i.e. $\Pi_{c_j} = \emptyset$, where Π_{c_j} denotes the set of parents of c_j. For Naive Bayes, each attribute has the class variable as its unique parent, namely, $\Pi_{t_{ki}} = \{c_j\}$, while in TAN, the class variable is a parent of each attribute, i.e. $c_j \in \Pi_{t_{ki}}$; besides c_j, each attribute has at most one other attribute as a parent, i.e. $|\Pi_{t_{ki}}| \leq 2$.

TAN models are formed by adding directional augmenting edges between attributes, see Figure 4.2 as an example, in this augmented structure, an edge from A_2 to A_1 implies the influence of A_2 on the assessment of the class variable also depends on the value of A_1. While in Figure 4.1, the influence of each attribute on the class variable is independent of other attributes.

The Construct-TAN learning procedure consists of seven main steps:

1. Input training data $U = \{D, C\}$, $D = \{d_1, d_2, \cdots, d_{|D|}\}$, $C = \{c_1, c_2, \cdots, c_{|C|}\}$, $c_j \in C$ is the category label of document $d_i \in D$, $d_i = \{t_{1i}, t_{2i}, \cdots, t_{\tau i}\}$.

2. Compute *Conditional Mutual Information* (CMI) between each pair of attributes given the class variable, it is formatted as

   $$I_{\hat{P}_U}(X;Y|C) = \sum_{x,y,c} \hat{P}_U(x,y,c) log \frac{\hat{P}_U(x,y|c)}{\hat{P}_U(x|c)\hat{P}_U(y|c)} \tag{4.6}$$

   where X and Y denote pair of attributes (terms), their values are x and y, $x \neq y$.

   CMI is based on *empirical distribution* $\hat{P}_U$ on the training data, which is defined by frequencies of observations.

3. Build a complete undirected graph in which the nodes are the attributes $t_{1i}, t_{2i}, \cdots, t_{\tau i}$, and the weight of edges that connect each node pair are annotate by $I_{\hat{P}_U}(t_i; t_j|c)$.

4. Select a subset of arcs from the graph to construct a *maximal weighted spanning tree* in which the sum of weights is maximized.

5. Choose a root variable and transform the undirected tree to a directed one by setting the direction of all edges to be outward from it.

6. Choose a class variable and set the directions from it to all attributes.

7. Learn the parameters and output the TAN.

TAN for Text Classification

Based on the augmented tree structure above, a TAN classifier model can be applied to text classification in a similar manner to a naive Bayes model by computing the highest posterior probability,

$$c^* = \arg\max_{c \in C} \left\{ P(c_j) \prod_{k=1}^{\tau} P(t_{ki}|\Pi_{t_{ki}}) \right\} \tag{4.7}$$

where $\Pi_{t_{ki}}$ is learned based on TAN model, which has two forms:

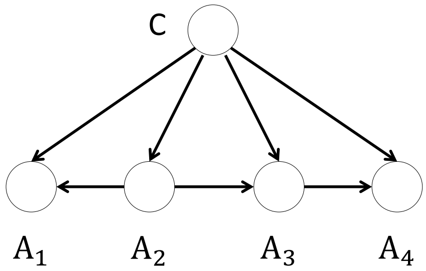


Figure 4.2: A simple tree augmented naive Bayes structure

- $\Pi_{t_{ki}} = \{c_j\}$, t_{ki} has no non-category parent.

- $\Pi_{t_{ki}} = \{c_j, t_{\pi(ki)}\}$, t_{ki} has one non-category parent.

TAN approaches relax the strong assumption of independence in naive Bayes by exploring correlations among attributes, and experimentally outperform it while maintaining computational simplicity on learning (Friedman et al. 1997). A number of TAN algorithms have been proposed to enhance classification accuracy and efficiency, such as SuperParent TAN (Keogh & Pazzani 1999) and StumpNetwork (Zhang & Ling 2001). Some methods further relax the conditional independence assumption of TAN by taking more attributes correlation into account, e.g. KDB (Rubio & Gámez 2011), assuming that every attribute has k parents at most.

However, TAN and its extensions have rarely been used in text classification applications, the reason is that these methods only concern the dependence of terms, but overlook the high order semantics capturing, underlying information is ignored. To address this problem, we proposed a *Hierarchical structured TAN* (HTAN) to represent documents as hierarchical trees that extract implicit relation of attributes layer by layer, high order semantics is fully captured.

4.3 Hierarchical Tree Learning

Documents should be expressed in a representation before classification algorithms are performed. This assists in building a feature space that consists of all the necessary terms with their correlations captured and embedded in a similarity learning model. In this section, we propose a novel tree augmented model containing a hierarchical structure to select features and extract feature correlations. Based on our concrete analysis, the hierarchical tree is a term-set level semantic representation and it is able to capture high order semantics.

4.3.1 Feature Extraction

In order to enhance classification efficiency, the feature is identified before the feature space construction. When traditional TAN is applied to text classification, the feature-vector document representation is used. This takes one document as a set of term occurrence frequency sequence. Term frequencies are regarded as features, and then applied to a weighted scheme CMI, which is used as a weighting factor to reflect the correlation strength of a pairwise terms to a document in a corpus. This representation proves to be efficient and powerful, however, it is not enough to match the main task of text and natural language learning, that is, to learn the *semantics* of words without prior linguistic knowledge. The reasons behind this lie in the fact that it is a *term-level* document representation with the feature selected individually. In addition, it considers the meaning of documents only in terms of the mutual information of each pairwise term share, regardless of the real contents and themes that the documents express.

Probabilistic topic modelling is a reasonable way to represent documents with thematic information. Through this approach, the selected features are not original terms in a corpus, but semantic topics that are probability distributions over the terms in a corpus (Blei 2012).

Our hierarchical tree model borrows this feature extraction idea and con-

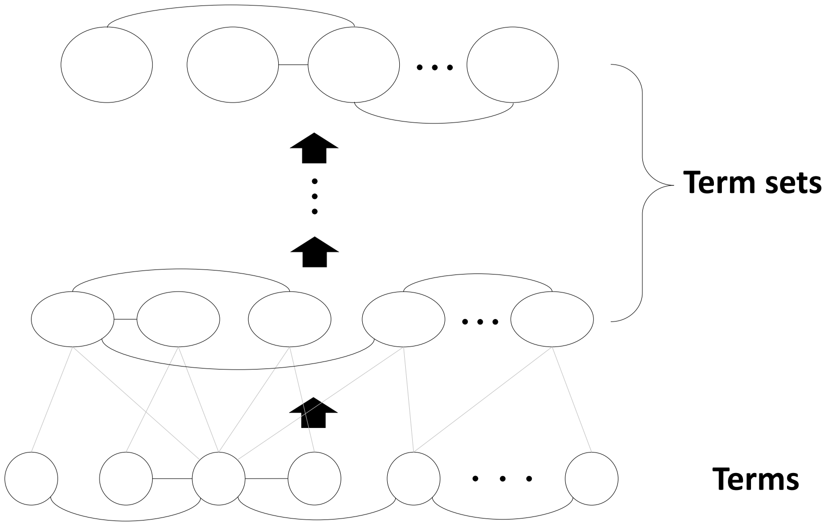


Figure 4.3: Schematic diagram of feature extraction procedure in HTAN

structs a tree-like hierarchical structure. The features of each layer are extracted from the lower layer by building a *maximal weighted spanning tree* in which the sum of weights is maximised. The terms that are close in meaning are merged as term sets and then treated as new features for the purposes of the next level. By this hierarchical feature extraction method, features can carry more information to characterise a document, and at the higher layer structure, more semantics may be contained.

Specifically, the **Hierarchical Feature Extraction** procedure consists of the following steps:

- For the first hierarchical tree layer, the elements in the selected feature set are the original terms in a corpus;

- From the second hierarchical tree layer, based on the results of the maximal weighted spanning tree of the former order, the features that have high correlations are aggregated as a new feature;

- If we repeat the computation of each order until the last layer, features that are extracted from the former order are used to construct matrices of classification or clustering.

As the illustration in Figure 4.3 demonstrates, each node denotes a feature, and the edges denotes correlations that connect each pair, while every layer is a maximal weighted spanning tree. From the bottom to top, only strong correlations are kept and terms are aggregated hierarchically. Furthermore, each feature carries more and more terms with their correlations, i.e. semantics.

This hierarchical feature extraction structure guarantees that with the higher layer hierarchical tree, the features are reflected as term sets with more comprehensive semantic information involved. It is not a simple term-level document representation, but a layer-increased *term set-level* document representation. This contains the correlations and similarity between terms and term sets which form a document. Moreover, this efficient layer-wise learning results in two distinct advantages:

- High order and implicit semantics are mined. We no longer represent the semantics in a document by cutting it into the smallest pieces but merge the terms into larger and more comprehensive features based on their semantic relatedness.

- Word sense ambiguity is avoided by higher order features.

Intuitively, if we imagine a document is just like a photo, there is no doubt that the bigger pieces carry more information than the pixel data, so it is easier to understand ”what actually has been expressed” and ”what was intended” of that photo based on the representation of pieces and their correlations.

In the following section we outline detailed approaches for the purposes of more deeply exploring feature correlations.

4.3.2 Feature Correlation

It is very challenging to analyse semantic relation due to it being driven by both intrinsic textual/linguistic complexity (such as natural language ambiguity) and various relationships (such as co-occurrence) between terms and documents.

In traditional TAN, *conditional mutual information* (CMI) $I_{\hat{P}_U}(X;Y|C)$ is employed to reflect the correlation strength of variables. This measures the information that Y provides about X when the value of class C is known. Empirical distribution, which is defined by frequencies of observations, is embedded in so that we can compute the joint probability between features.

Here, we propose a more general concept to summarise the feature correlation strength from single-level to multi-level structure, the **Hierarchical Correlation Ratio** (HCR).

Definition 4.1 *For the training data set* $U = \{D, C\}$*,* $D = \{d_1, d_2, \cdots, d_{|D|}\}$*,* $C = \{c_1, c_2, \cdots, c_{|C|}\}$*,* $c_j \in C$ *is the category label of document* $d_i \in D$*, and* d_i *is represented as terms or term sets sequence for different layer of a hierarchical tree,* $d_i = \{t_{1i}, t_{2i}, \cdots, t_{\tau i}\}$*. Assume that there are* n *layers, for* $\forall t_\alpha, t_\beta \in d_i$*,*

- *on the* $[1, n-1]$ *layers, HCR of features is calculated by an unsupervised dependency function,* $Corr(t_\alpha, t_\beta)$*;*
- *on the* n*th layer, HCR of features is calculated by a supervised dependency function with class variable known,* $Corr(t_\alpha, t_\beta|c_j)$*.*

HCR is able to detect almost any functional second-moment (pairwise or quadratic) dependency, e.g. the entropy-based mutual information, total correlation, dual total correlation and even more general dependencies. We have chosen mutual information (MI) in this instance.

- On the $[1, n-1]$ layers, HCR is denoted as MI of each term (term set) pair

$$\begin{aligned} Corr(t_\alpha, t_\beta) &= MI(t_\alpha, t_\beta) \\ &= \sum_{t_\alpha, t_\beta} P(t_\alpha, t_\beta) \log \frac{P(t_\alpha, t_\beta)}{P(t_\alpha)P(t_\beta)} \end{aligned} \tag{4.8}$$

- On the nth layer, HCR is denoted as CMI of each term (term set) pair with class variable c_j is given

$$\begin{aligned} Corr(t_\alpha, t_\beta | c_j) &= CMI(t_\alpha; t_\beta | c_j) \\ &= \sum_{t_\alpha, t_\beta, c_j} P(t_\alpha, t_\beta, c_j) \log \frac{P(t_\alpha, t_\beta | c_j)}{P(t_\alpha | c_j) P(t_\beta | c_j)} \end{aligned} \tag{4.9}$$

Except some correlation methods which captures dependency directly from data characteristics (e.g. Pearson's correlation) or distance (e.g. distance correlation), various correlation measures in use may be undefined for certain joint distributions. That is to say, HCR holds a sensitivity to data distribution. In the following section we illustrate several typical approaches to compute joint distribution with a different data structure hypothesis.

Joint Probability Estimator

Since undefined feature distribution leads to a difference of HCR results, the crucial part of feature correlation is to estimate the joint probability of terms and term sets.

Let P be a joint probability distribution over the discrete feature variables in D, for $\forall$ term (term set) pair $t_\alpha, t_\beta \in d_i$, various methods to compute $P(t_\alpha, t_\beta)$ are listed as follows.

1. Empirical Distribution

(1) Binary Scheme

Traditional TAN uses empirical distribution to measure joint probability.

Let $\hat{P}_T(\cdot)$ be the empirical distribution, which is defined by frequencies of observations $T = \{t_1, t_2, \cdots, t_\tau\}$, is given by

$$\hat{P}_T(A) = \frac{1}{\tau} \sum_i \delta_A(t_i) \tag{4.10}$$

where the *delta function* returns $\delta_A(t_i) = 1$ if $t_i \in A$, $\delta_A(t_i) = 0$ otherwise. Similarly,

$$P(t_\alpha, t_\beta) = \hat{P}_T(t_\alpha, t_\beta) = \frac{1}{\tau} \sum_i \delta_{t_\alpha, t_\beta}(t_i) \tag{4.11}$$

where the joint probability is defined as the probability of documents that contain corresponding terms or term sets.

This simple counting method is under the assumption that documents are *independent* and *identically distributed* (iid), any documents with term (term set) t_α, t_β are counted as 1, or 0 otherwise.

(2) Raw Frequency Scheme

In vector space models, it is common to calculate joint probabilities considering term frequencies. *tpf*, short for *term pair occurrence frequency*, reflects the importance of a term pair to a document in a corpus. $tpf\big((t_\alpha, t_\beta), d_i\big)$ counts the number of times a term pair t_α, t_β occurs in a document d_i. The joint probability based on tpf scheme is formatted as:

$$P(t_\alpha, t_\beta) = \frac{\sum_{d_i} tpf\big((t_\alpha, t_\beta), d_i\big)}{\sum_{t_\alpha, t_\beta} \sum_{d_i} tpf\big((t_\alpha, t_\beta), d_i\big)} \tag{4.12}$$

where (t_α, t_β) stands for a term pair, and d_i is a single document in a document collection D, the joint probability of a term pair is the probability of the term pair in document set D.

Term occurrence frequency method treats terms bounded with weights to reflect their importance and only keeps multiplicity, but disregards the order, structure, meaning, grammar, etc. of the terms. It assumes that terms are regarded highly relational if they co-occur frequently in the same documents.

2. Multinomial Distribution

LDA (Blei et al. 2003) is a generative probabilistic model which represent documents as random mixtures over latent topics, where each topic is characterized by a distribution over words.

Joint distribution in LDA is calculated as

$$\begin{aligned} P_\theta(t_\alpha, t_\beta) &= \sum_z P(t_\alpha, t_\beta, z, \theta) \\ &= \sum_z P(t_\alpha|z)P(t_\beta|z)P(z|\theta) \end{aligned} \tag{4.13}$$

where θ is a Dirichlet distributed random vector; a topic z follows multinomial distribution on the condition of θ, $P(z_i|\theta) = \theta_i$; a term (term set) is a multinomial probability conditioned on the topic, denoted as $P(t_\alpha|z)$.

LDA assumes that terms are conditional independent given the latent topics.

Various correlation methods can be employed here to compute the feature-wise dependencies in hierarchical trees, with corresponding joint probability estimator. Empirically, we recommend joint probability estimator which consider more term couplings and relax the strong iidness assumption.

Algorithm below summarize the hierarchical tree learning procedure.

Algorithm 3: Hierarchical Tree Learning

```
Input: Terms $T_1$, joint probability estimator $j$, $n$ layers.
Output: $Tree_n(T_n, Corr_j)$
for $i = 1, \cdots, n$ do
    for $t_\alpha, t_\beta \in T_i$, $(t_\alpha \neq t_\beta)$ do
        Compute $Corr_j(t_\alpha, t_\beta)$;
    end
    Construct a maximal weighted spanning tree $Tree_i(T_i, Corr_j)$;
    $T_{i+1}$ is a set of aggregation of linked nodes in $Tree_i$;
end
```

After the construction of a hierarchical tree, the top layer of the tree is a network which contains dependencies between extracted term sets, it has a broad application. In the next section, we demonstrate an example of its application in a TAN text classifies.

4.4 Hierarchical Structured TAN (HTAN)

In this section, we describe a procedure for constructing a hierarchical tree augmented naive Bayes network (HTAN), which is a combination of the hierarchical tree and the tree augmented naive Bayes model, containing as it does, a hierarchical structure to select features and extract feature correlations.

4.4.1 The Construct-HTAN Procedure

Algorithm 3 generalise a tree-like hierarchical structure to extract features and calculate feature correlations. It is represented as a complete undirected graph with the features and correlations associated. This means that, it has a wide application, not only in terms of classifications but also unsupervised learning measures. HTAN is one of the applications in the text classification area.

The **Construct-HTAN Procedure** consists of the following steps:

1. Input training data $U = \{D, C\}$, $D = \{d_1, d_2, \cdots, d_{|D|}\}$, $C = \{c_1, c_2, \cdots, c_{|C|}\}$, $c_j \in C$ is the category label of document $d_i \in D$, and d_i is represented as terms from corpus originally, $d_i = \{t_{1i}, t_{2i}, \cdots, t_{\tau i}\}$. Assume there are n layers.

2. For each layer from the 1st to $(n-1)$th, compute *Hierarchical Correlation Ratio* (HCR) between each pair of features (terms or term sets), it is formatted as $Corr(t_\alpha, t_\beta)$.

3. Build a complete undirected graph in which the nodes are the features, and the weight of edges that connect each node pair are annotate by HCR.

4. Select a subset of arcs from the graph to construct a *maximal weighted spanning tree* in which the sum of weights is maximized.

5. As features with high HCR are kept, those features are aggregated as new features to compute HCR for the next layer.

6. Repeat step 2 to 5 until the nth layer, compute $Corr(t_\alpha, t_\beta|c_j)$ with class variable known and construct the maximal weighted spanning tree.

7. Choose a root variable and transform the undirected tree to a directed one by setting the direction of all edges to be outward from it.

8. Choose a class variable and set the directions from it to all attributes.

9. Output the HTAN.

Figure 4.4 is a simple instance of the HTAN structure. From the bottom to the top, each node denotes a feature, and the edges denote the HCR that connects each pair of them so that every layer is a maximal weighted spanning tree. It is unsupervised learning from the 1st to $(n-1)$th ordering, whereby the features are original terms from the corpus at the first order (the bottom layer), and then they are merged hierarchically in order to maximise the HCR. In the meantime, the features are presented as bigger and bigger term sets that can contain more information and correlations hierarchically. Until the last order (the top layer), it is through supervised learning that we consider the HCR of features in relation to the assessment of the class variable. Similar to the TAN, the undirected maximal weighting spanning tree is transformed to a directed one by choosing a start node and setting the direction so that all edges point outward from it. There are arcs from C to each feature, ensuring that the class variable is the parent of each feature,

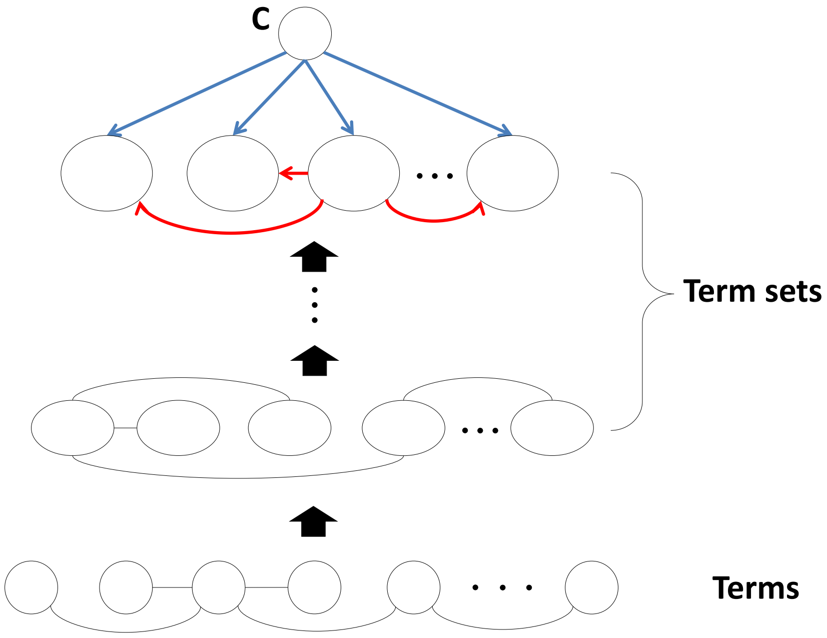


Figure 4.4: The structure of the hierarchical tree augmented naive Bayes network

i.e. $c_j \in \Pi_{t_\alpha}$; besides c_j, and each feature has, at most, one other feature as a parent, i.e. $|\Pi_{t_\alpha}| \leq 2$.

Theorem 4.1 *Let U be a collection of N instances C, D. The procedure Construct-HTAN builds a HTAN network that maximizes $\sum_{t_\alpha, t_\beta} Corr(t_\alpha, t_\beta)$ from the first to the $(n-1)$th ordering, and maximize $\sum_{\Pi_{t_\alpha}} Corr(t_\alpha, \Pi_{t_\alpha})$ for the nth ordering. It has the time complexity $O(n \cdot \tau^2 \cdot N)$.*

Π_{t_α} is learned from HTAN model, which has two forms:

- *$\Pi_{t_\alpha} = \{c_j\}$, t_α has no non-category parent.*
- *$\Pi_{t_\alpha} = \{c_j, t_{\pi(\alpha)}\}$, t_α has one non-category parent.*

The next section we demonstrate how to apply HTAN to distinguish documents from categories as a text classifier.

4.4.2 Using HTAN as Text Classifiers

After hierarchical tree learning procedure, the tree at the top layer is a pruned tree consist of nodes which are features containing more semantics, and edges which are feature correlations. This tree-like structure supports wide applications, both supervised and unsupervised learning algorithms. When applied to classification, various approaches can be employed, e.g. naive Bayes, TAN, KDB and more advanced models. We here also borrow TAN to continue the follow-on contents.

As a HTAN classifier is trained, the second step of text classification approaches is the classification step, where the classifier is used to predict class labels for testing data. A HTAN model can be applied to text classification in a similar manner to a TAN model. In this case, *maximum a posterior* (MAP) classifier can be constructed by seeking the optimal category which maximizes the posterior $P(c_j|d_i)$,

$$\begin{aligned} c^* &= \arg\max_{c\in C}\{P(c_j|d_i)\} \\ &= \arg\max_{c\in C}\{P(c_j)P(d_i|c_j)\} \\ &= \arg\max_{c\in C}\left\{P(c_j)\prod_{k=1}^{\tau}P(t_\alpha|\Pi_{t_\alpha})\right\} \end{aligned} \tag{4.14}$$

HTAN encodes conditional independence statements that each feature is independent of its non-descendants given the state of its parent(s), a MAP classifier based on Equation 4.14 is optimal.

Then, classifying a new document from testing set is to estimate two groups of probabilities from the training set, $P(c_j)$, the prior probability of class variables; and $P(t_\alpha|\Pi_{t_\alpha})$, the likelihood function which is the probability of each feature (term or term set) given its parent(s).

In Bayesian probability theory, if the posterior distributions are in the same family as the prior probability distribution, the prior is called a *conjugate prior* for the likelihood function. It has an algebraic convenience, giving a closed-form expression for the posterior; otherwise a difficult numerical integration may be necessary. Further, conjugate priors may give intuition, by more transparently showing how a likelihood function updates a prior distribution (Heckerman 2008). The Dirichlet distribution is a conjugate prior for the multinomial distribution. This means that if the prior distribution of the multinomial parameters is Dirichlet then the posterior distribution is also a Dirichlet distribution (with parameters different from those of the prior).

It can certainly be discussed whether these are good reasons to choose a particular prior, as these criteria are unrelated to actual prior beliefs. Nevertheless, conjugate priors are widely accepted, as they often are reasonably flexible and convenient to use for the reasons stated above.

When Dirichlet distribution is used as the prior, HTAN assumes the following generative process,

1. $Pr\big(C = c_1, \cdots, c_{|C|}\big) \sim \text{Multinomial}(\theta)$
   $\theta \sim \text{Dir}(\theta^0)$
2. $Pr\big(T_1 = t_1, \cdots, T_\tau = t_\tau \big| \Pi_T\big) = \text{Multinomial}\ \big(P(T_1 = t_1, \cdots, T_\tau = t_\tau \big| \Pi_T)\big) \sim \text{Multinomial}(\lambda)$
   $\lambda \sim \text{Dir}(\lambda^0)$

In Bayesian learning of a Dirichlet distributed prior $P(C = c_j)$ for $j = 1, \cdots, |C|$, we set the parameters $\Theta = \{\theta_j : j = 1, \cdots, |C|\}$, where $\theta_j = P(C = c_j)$,

$$P(C = c_j) = \frac{\sum_{d_i} tf\big((t_i, d_i), c_j\big)}{\sum_{d_i} tf(t_i, d_i)} \tag{4.15}$$

which indicate the probability of t_i labelled by c_j, $tf(t_i, d_i)$ is the term t_i occurrence frequency in document d_i.

Parameters $\Lambda = \{\lambda_i : i = 1, \cdots, \tau\}$ of likelihood function also follows Dirichlet distribution, $\lambda_i = P(T = t_i|\Pi_{t_i})$. For each value of T, the distribution given a particular value of its parents is

$$P(T = t_i|\Pi_{t_i}) = \frac{\tau \cdot \hat{P}_T(t_i, \Pi_{t_i}) + N^0 \cdot \lambda^0}{\tau \cdot \hat{P}_T(\Pi_{t_i}) + N^0} \tag{4.16}$$

where λ^0 is the prior estimate of $P(T = t_i|\Pi_{t_i})$ and N^0 is the confidence associated with that prior. We set marginal probability of T as the prior probability, $\lambda^0 = \hat{P}(T = t_i)$. In (Friedman et al. 1997), N^0 is chosen as 5 in experiments and performed slightly better than other values. Going back to Equation 4.16, $\hat{P}_T(t_i, \Pi_{t_i})$ and $\hat{P}_T(\Pi_{t_i})$ are calculated as

$$\begin{aligned} \hat{P}_T(t_i, \Pi_{t_i}) = \hat{P}_T(t_i|\Pi_{t_i}) \cdot \hat{P}_T(\Pi_{t_i}) \\ = \frac{\sum_{d_i} tf\big((t_i, d_i), \Pi_{t_i}\big)}{\sum_{d_i} tf(t_i, d_i) - \sum_{d_i} tf(\Pi_{t_i}, d_i)} \cdot \frac{\sum_{d_i} tf\big((\Pi_{t_i}, d_i), c_j\big)}{\sum_{d_i} tf(t_i, d_i)} \end{aligned} \tag{4.17}$$

which are also probabilities by counting term frequencies that satisfy different conditions.

Equation 4.16 contains smoothing factors to avoid zero probability estimates. A zero estimate may happen in $\hat{P}_T(t_i|\Pi_{t_i})$ when the attribute values do not actually occur in documents, i.e., $tf(\cdot) = 0$.

Eventually, a graphical model of hierarchical tree augmented naive Bayes text classifier is given in Figure 4.4, where the root node is the class label, and each leaf node is a term sequentially occurring in the documents that are belonged to the category.

4.5 Conclusions

In this chapter, we have proposed a hierarchical tree learning measure to capture the high order semantics of terms, which have been overlooked by existing measures. Additionally, the hierarchical tree is able to address the traditional natural language processing problem, avoiding ambiguity by layer-wise larger and more comprehensive features.

The hierarchical tree achieves this through the means of a three-step procedure:

1. For the first hierarchical tree layer, the elements in the selected feature set are original terms in a corpus;

2. From the second hierarchical tree layer, based on the results of the maximal weighted spanning tree of the former order, features that have high correlations are aggregated as a new feature;
3. The computation of each order is repeated until the last layer and the features that are extracted from the former order are used for applications.

We have also introduced an example of the application, which combines the hierarchical tree with tree augmented naive Bayes network as a feature extraction procedure for TAN. The top layer of the tree is represented as a matrix whereby the features are the term sets that have been merged from the former layer. The matrix is then employed for text classification.

This research opens new opportunities to deeply explore high order semantics. Further research efforts could include: experimentally proving the advantages of the hierarchical tree, comparing HTAN with the naive Bayes classifier, training TAN and LDA on the same term set features, and comparing HTAN with the traditional TAN that is trained on the original terms. Each layer of the hierarchical tree is still a bag-of-words model which indicates that each layer is a undirected graph that ignores the semantic structure information. A sensible way to consider the appropriate direction is needed. We are working on training the hierarchical tree based on lexical resources, i.e. human hand-crafted language databases which contain complete semantic structure information.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In conclusion, this thesis presents several techniques to address document representation modeling based on semantic relatedness.

In Chapter 1, a detailed introduction of the background of my research topic is presented and the research issues and contributions are outlined.

In Chapter 2, a number of existing semantic measurements are reviewed and evaluated, including corpus-based document representations and lexical resource-based models, which offer two different ways to capture the semantic similarity and relations between terms. For corpus-based models, researchers try to model lexical semantic information in high-dimensional vectors e.g. by considering term occurrence patterns, contexts, locations, etc. While lexical resource-based models are built upon lexical databases, such as WordNet, this does not provide a term-similarity metric. However, various metrics based on its structure have been developed.

In Chapter 3, the semantic coupling similarity (SCS) measure is presented to completely and comprehensively capture the coupling relationships both within and between the term pairs in a document. This measure represents term couplings as a term linkage graph and considers the probabilistic distributions of terms and term couplings. A document set is then represented as a term coupling vector for document analysis. SCS is a four-step procedure, containing (1) captures the semantic intra-term couplings based on its pairwise term occurrence frequency pattern across a document set; (2) capture the semantic inter-term couplings based on the interactions with the link terms on all possible paths of term connections; (3) achieves a full coupled semantic similarity of term pairs via an optimal combination; and (4) represents the original document set by a coupled semantic similarity matrix, which can be broadly applied to document clustering and classification tasks.

The proposed measure is compared with typical document representations on various benchmark data sets. Our model produce outcomes that are great significant and consistently exceeds the performance of benchmark methods on most data sets.

In Chapter 4, the hierarchical tree learning algorithm is proposed to extract high order semantics between correlated terms and prune weak correlations to maintain efficiency. The hierarchy is built in a three-step procedure: (1) a hierarchical feature extraction and correlation computation procedure is employed whereby highly correlated terms are merged into sets and are associated with more complete semantic information; (2) each layer constitutes a maximal weighted spanning tree to prune weak feature correlations; (3) the top level of the tree is a high order semantic matrix of terms and can be applied to both supervised and unsupervised learning algorithms.

Chapter 3 of this thesis is supported by a published conference papers[1] listed in the **List of Publications**. Accordingly, we have sought through the means of this thesis to add considerable value to the document representation research and its specific application to the text mining area.

5.2 Future Work

This research opens new opportunities to deeply explore semantic similarity. Further research efforts could be directed towards some of the issues and

[1] The paper of Chapter 3 is published, the paper of Chapter 4 is still under modification.

challenges stated below:

1. A theoretical analysis of the effect of the number of link terms in SCS should be undertaken.

2. More experiments should be developed regarding the proposed HTAN algorithm in future works.

3. Semantic measures built on term pairs and term sets are helpful to avoid the natural language ambiguities of every single term, but such document representations are experimentally expensive due to the high dimensional feature space. A sensible way forward needs to be identified in order to project documents into a new and smaller space.

4. Experimental results should be presented to support the main claims of the papers in order to enhance in the modelling of polysemy and synonmy.

5. We have finished the work in relation to coupled term pair similarity but there is a need to introduce the coupled idea into the calculation of the document pair relation.

6. Adapting the hierarchical tree with lexical resources would serve to capture more semantic structure information.

Appendix A

List of Symbols

The following list is neither exhaustive nor exclusive, but may be helpful.

$tfidf(t, d, D)$	The importance of a term t to a document d in a corpus D
$tf(t, d)$	The number of times term t occurs in document d
$tpfidf((t_i, t_j), d, D)$	The importance of a term pair (t_i, t_j) to a document d in a corpus D
tpf	counts the number of times a term pair occurs in a document
M_{tpf}	The term pair occurrence frequency matrix
$P^{Ia}(t_k\|t_i)$	The probability of the term pair (t_k, t_i) in corpus D
$P^{Ia}(t_i)$	The probabilities over all term pairs given t_i
RS	The relation strength function
$IaR(t_i, t_j)$	The intra-term coupling similarity
G_{tpf}	The term pair frequency graph
G_{IaR}	The intra-term coupling graph

G_{IeR}	The inter-term coupling graph
$Path(t_i, t_j)$	A path with the initial vertex t_i and the terminal vertex t_j
T_{link}	The link term set
$tpfipf((t_i, t_j), d, m)$	The importance of a term pair (t_i, t_j) to all possible m paths
$W(t_k \vert t_i)$	The weight of a term pair (t_k, t_i) in graph G_{IeR}
$W_{t_{l_1} \cdots t_{l_n}}(t_k \vert t_i)$	The weight of one path through $t_{l_1} \cdots t_{l_n}$ between term pair (t_k, t_i) in G_{tpf}
$W_m(t_k \vert t_i)$	The weight of m paths between term pair (t_k, t_i)
$P^{Ie}(t_k \vert t_i)$	The probability of the term pair (t_k, t_i) on all possible m paths
$P^{Ie}(t_i)$	The probabilities over all term pairs on possible m paths given t_i
$IeR(t_i, t_j)$	The inter-term coupling similarity
$CTPS(t_i, t_j)$	The coupled similarity of term pair (t_k, t_i)
M_{cou}	The coupled similarity matrix
$Sim(d_i, d_j)$	The document similarity of document pair (d_i, d_j)
$\hat{phi}(d)$	The new document representation based on CTPS
$P(c_j \vert d_i)$	The posterior probability
$I(X; Y \vert C)$	The conditional mutual information between each pair of attributes given the class variable
Π_{t_k}	The parent set of t_k in a Bayesian network

$Corr(t_\alpha, t_\beta)$	The hierarchical correlation ratio of term pair (t_α, t_β)
$P(t_\alpha, t_\beta)$	The joint probability estimator
$Tree_i(T_i, Corr)$	The maximal weighted spanning tree with nodes T_i

Bibliography

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M. & Soroa, A. (2009), A study on similarity and relatedness using distributional and wordnet-based approaches, *in* ‘NAACL’, pp. 19–27.

Arora, R. & Ravindran, B. (2008), Latent dirichlet allocation based multi-document summarization, *in* ‘Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data’, pp. 91–97.

Billhardt, H., Borrajo, D. & Maojo, V. (2002), ‘A context vector model for information retrieval’, *JASIST* **53**, 236–249.

Blei, D. M. (2004), Probabilistic models of text and images, PhD thesis, University of California, Berkeley.

Blei, D. M. (2012), ‘Probabilistic topic models’, *Communications of the ACM* **55**(4), 77–84.

Blei, D. M., Griffiths, T. L. & Jordan, M. I. (2010), ‘The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies’, *Journal of the ACM (JACM)* **57**(2), 7.

Blei, D. M. & Lafferty, J. D. (2006), Dynamic topic models, *in* ‘Proceedings of the 23rd international conference on Machine learning’, pp. 113–120.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), ‘Latent dirichlet allocation’, *Journal of Machine Learning Research* **3**, 993–1022.

Boyd-Graber, J. L. & Blei, D. M. (2009), Syntactic topic models, *in* 'Advances in neural information processing systems', pp. 185–192.

Budanitsky, A. & Hirst, G. (2006), 'Evaluating wordnet-based measures of lexical semantic relatedness', *Computational Linguistics* pp. 13–47.

Bullinaria, J. A. & Levy, J. P. (2007), 'Extracting semantic representations from word co-occurrence statistics: A computational study', *Behavior research methods* **39**, 510–526.

Cao, L. (2013), 'Non-iidness learning in behavioral and social data', *The Computer Journal* p. bxt084.

Cao, L. (2014), 'Coupling learning of complex interactions', *Information Processing & Management* .

Cao, L., Ou, Y. & Yu, P. S. (2012), 'Coupled behavior analysis with applications', *Knowledge and Data Engineering, IEEE Transactions on* **24**(8), 1378–1392.

Castillo, J. J. (2011), 'A wordnet-based semantic approach to textual entailment and cross-lingual textual entailment', *International Journal of Machine Learning and Cybernetics* **2**, 177–189.

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

Chen, X., Zhou, M. & Carin, L. (2012), The contextual focused topic model, *in* 'Proceedings of the 18th ACM SIGKDD', pp. 96–104.

Cheng, X., Miao, D., Wang, C. & Cao, L. (2013), Coupled term-term relation analysis for document clustering, *in* 'IJCNN', pp. 1–8.

Chowdhury, G. (2010), *Introduction to modern information retrieval*, Facet publishing.

Cross, V. (2004), Fuzzy semantic distance measures between ontological concepts, *in* 'Fuzzy Information, 2004. Processing NAFIPS'04. IEEE Annual Meeting of the', Vol. 2, IEEE, pp. 635–640.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W. & Harshman, R. A. (1990), 'Indexing by latent semantic analysis', *JASIS* **41**, 391–407.

Devitt, A. & Vogel, C. (2004), The topology of wordnet: Some metrics, *in* 'Proc. of the 2nd International WordNet Conference (GWC)', pp. 106–111.

Farahat, A. K. & Kamel, M. S. (2011), 'Statistical semantics for enhancing document clustering', *Knowledge and information systems* **28**(2), 365–393.

Frank, E., Hall, M. & Pfahringer, B. (2002), Locally weighted naive bayes, *in* 'Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence', pp. 249–256.

Friedman, N., Geiger, D. & Goldszmidt, M. (1997), 'Bayesian network classifiers', *Machine learning* **29**(2-3), 131–163.

Gabrilovich, E. & Markovitch, S. (2007), Computing semantic relatedness using wikipedia-based explicit semantic analysis., *in* 'IJCAI', Vol. 7, pp. 1606–1611.

Gruber, T. R. (1993), 'A translation approach to portable ontology specifications', *Knowledge acquisition* **5**(2), 199–220.

Halawi, G., Dror, G., Gabrilovich, E. & Koren, Y. (2012), Large-scale learning of word relatedness with constraints, *in* 'the 18th ACM SIGKDD', pp. 1406–1414.

Hawalah, A. & Fasli, M. (2011), A graph-based approach to measuring semantic relatedness in ontologies, *in* 'WIMS', p. 29.

Heckerman, D. (2008), A tutorial on learning with bayesian networks, *in* 'Innovations in Bayesian Networks', Springer, pp. 33–82.

Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E. G. & Milios, E. (2006), 'Information retrieval by semantic similarity', *IJSWIS* **2**(3), 55–73.

Hofmann, T. (1999), Probabilistic latent semantic indexing, *in* 'SIGIR', pp. 50–57.

Hofmann, T. (2001), 'Unsupervised learning by probabilistic latent semantic analysis', *Machine learning* **42**(1-2), 177–196.

Holloway, T., Bozicevic, M. & Börner, K. (2007), 'Analyzing and visualizing the semantic coverage of wikipedia and its authors', *Complexity* **12**(3), 30–40.

Hong-Bo, S., Zhi-Hai, W., Hou-Kuan, H. & Li-Ping, J. (2002), Text classification based on the tan model, *in* 'TENCON'02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering', Vol. 1, IEEE, pp. 43–46.

Jackson, P. & Moulinier, I. (2007), *Natural language processing for online applications: Text retrieval, extraction and categorization*, Vol. 5, John Benjamins Publishing.

Jiang, J. J. & Conrath, D. W. (1997), 'Semantic similarity based on corpus statistics and lexical taxonomy', *arXiv preprint cmp-lg/9709008* .

Jiang, L., Zhang, H. & Su, J. (2005), Instance cloning local naive bayes, *in* 'Advances in Artificial Intelligence', Springer, pp. 280–291.

Jing, L.-P., Huang, H.-K. & Shi, H.-B. (2002), Improved feature selection approach tfidf in text mining, *in* 'Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on', Vol. 2, IEEE, pp. 944–946.

Kalogeratos, A. & Likas, A. (2012), 'Text document clustering using global term context vectors', *Knowl. Inf. Syst.* **31**, 455–474.

Keogh, E. & Pazzani, M. (1999), Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches, *in* 'Proceedings of the seventh international workshop on artificial intelligence and statistics', Citeseer, pp. 225–230.

Kohavi, R. (1996), Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid., *in* 'KDD', pp. 202–207.

Kononenko, I. (1991), Semi-naive bayesian classifier, *in* 'Machine LearningEWSL-91', pp. 206–219.

Lebanon, G., Mao, Y. & Dillon, J. V. (2007), 'The locally weighted bag of words framework for document representation.', *Journal of Machine Learning Research* **8**(10), 2405–2441.

Li, P., Wang, H., Zhu, K. Q., Wang, Z. & Wu, X. (2013), Computing term similarity by large probabilistic isa knowledge, *in* 'CIKM', pp. 1401–1410.

Lin, C. & He, Y. (2009), Joint sentiment/topic model for sentiment analysis, *in* 'Proceedings of the 18th ACM CIKM', pp. 375–384.

Lin, D. (1998), An information-theoretic definition of similarity., *in* 'ICML', Vol. 98, pp. 296–304.

Maguitman, A. G., Menczer, F., Roinestad, H. & Vespignani, A. (2005), Algorithmic detection of semantic similarity, *in* 'Proceedings of the 14th international conference on WWW', pp. 107–116.

Mei, Q., Ling, X., Wondra, M., Su, H. & Zhai, C. (2007), Topic sentiment mixture: modeling facets and opinions in weblogs, *in* 'Proceedings of the 16th international conference on World Wide Web', ACM, pp. 171–180.

Miao, G., Guan, Z., Moser, L. E., Yan, X., Tao, S., Anerousis, N. & Sun, J. (2012), Latent association analysis of document pairs, *in* 'the 18th ACM SIGKDD', ACM, pp. 1415–1423.

Peng, F., Schuurmans, D. & Wang, S. (2004), 'Augmenting naive bayes classifiers with statistical language models', *Information Retrieval* **7**(3-4), 317–345.

Rada, R., Mili, H., Bicknell, E. & Blettner, M. (1989), 'Development and application of a metric on semantic nets', *Systems, Man and Cybernetics, IEEE Transactions on* **19**, 17–30.

Radinsky, K., Agichtein, E., Gabrilovich, E. & Markovitch, S. (2011), A word at a time: Computing word relatedness using temporal semantic analysis, *in* 'WWW', pp. 337–346.

Rennie, J. D., Shih, L., Teevan, J., Karger, D. R. et al. (2003), Tackling the poor assumptions of naive bayes text classifiers, *in* 'ICML', Vol. 3, pp. 616–623.

Resnik, P. et al. (1999), 'Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language', *J. Artif. Intell. Res.(JAIR)* **11**, 95–130.

Richardson, R., Smeaton, A. & Murphy, J. (1994), 'Using wordnet as a knowledge base for measuring semantic similarity between words'.

Rohde, D. L., Gonnerman, L. M. & Plaut, D. C. (2004), 'An improved method for deriving word meaning from lexical co-occurrence', *Cognitive Psychology* **7**, 573–605.

Rubio, A. & Gámez, J. A. (2011), Flexible learning of k-dependence bayesian network classifiers, *in* 'Proceedings of the 13th annual conference on Genetic and evolutionary computation', pp. 1219–1226.

Salton, G., Wong, A. & Yang, C. S. (1975), 'A vector space model for automatic indexing', *Commun. ACM* **18**, 613–620.

Sánchez, D., Batet, M. & Isern, D. (2011), 'Ontology-based information content computation', *Knowledge-Based Systems* **24**, 297–303.

Seco, N., Veale, T. & Hayes, J. (2004), An intrinsic information content metric for semantic similarity in wordnet, *in* 'ECAI', Vol. 16, p. 1089.

Song, L., Ma, J., Liu, H., Lian, L. & Zhang, D. (2007), Fuzzy semantic similarity between ontological concepts, *in* 'Advances and Innovations in systems, computing sciences and software engineering', Springer, pp. 275–280.

Strube, M. & Ponzetto, S. P. (2006), Wikirelate! computing semantic relatedness using wikipedia, *in* 'AAAI', Vol. 6, pp. 1419–1424.

Strzalkowski, T. (1994), Document representation in natural language text retrieval, *in* 'HLT'.

Teh, Y. W., Jordan, M. I., Beal, M. J. & Blei, D. M. (2006), 'Hierarchical dirichlet processes', *Journal of the american statistical association* **101**(476).

Wallach, H. M. (2006), Topic modeling: Beyond bag-of-words, *in* 'Proceedings of the 23rd International Conference on Machine Learning', ICML '06, pp. 977–984.

Wang, C., Cao, L., Wang, M., Li, J., Wei, W. & Ou, Y. (2011), Coupled nominal similarity in unsupervised learning, *in* 'Proceedings of the 20th ACM international conference on Information and knowledge management', ACM, pp. 973–978.

Wang, C., Yu, X., Li, Y., Zhai, C. & Han, J. (2013), Content coverage maximization on word networks for hierarchical topic summarization, *in* 'CIKM', ACM, pp. 249–258.

Webb, G. I., Boughton, J. R. & Wang, Z. (2005), 'Not so naive bayes: aggregating one-dependence estimators', *Machine learning* **58**(1), 5–24.

Witten, I. & Milne, D. (2008), 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', pp. 25–30.

Wong, S. K. M., Ziarko, W. & Wong, P. C. N. (1985), Generalized vector spaces model in information retrieval, *in* 'Proceedings of the 8th Annual International ACM SIGIR Conference', pp. 18–25.

Yu, Y., Wang, C., Gao, Y., Cao, L. & Chen, X. (2013), A coupled clustering approach for items recommendation, *in* 'Advances in Knowledge Discovery and Data Mining', Springer, pp. 365–376.

Zhang, H., Jiang, L. & Su, J. (2005), Hidden naive bayes, *in* 'Proceedings of the National Conference on Artificial Intelligence', Vol. 20, p. 919.

Zhang, H. & Ling, C. X. (2001), An improved learning algorithm for augmented naive bayes, *in* 'Advances in Knowledge Discovery and Data Mining', Springer, pp. 581–586.

Zhou, Z., Wang, Y. & Gu, J. (2008), A new model of information content for semantic similarity in wordnet, *in* 'Future Generation Communication and Networking Symposia, 2008. FGCNS'08. Second International Conference on', Vol. 3, IEEE, pp. 85–89.