Information Sciences 179 (2009) 2249–2262

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Exploiting noun phrases and semantic relationships for text document clustering

Hai-Tao Zheng, Bo-Yeong Kang, Hong-Gee Kim *

Biomedical Knowledge Engineering Laboratory, BK21 College of Dentistry, Seoul National University, 28 Yeongeon-dong, Jongro-gu, Seoul 110-810, Republic of Korea

ARTICLE INFO

Article history:
Received 30 July 2008
Received in revised form 27 February 2009
Accepted 27 February 2009

Keywords:
Ontology
WordNet
Text document clustering
Noun phrase
Hypernymy
Hyponymy
Holonymy
Meronymy

ABSTRACT

Text document clustering plays an important role in providing better document retrieval, document browsing, and text mining. Traditionally, clustering techniques do not consider the semantic relationships between words, such as synonymy and hypernymy. To exploit semantic relationships, ontologies such as WordNet have been used to improve clustering results. However, WordNet-based clustering methods mostly rely on single-term analysis of text; they do not perform any phrase-based analysis. In addition, these methods utilize synonymy to identify concepts and only explore hypernymy to calculate concept frequencies, without considering other semantic relationships such as hyponymy. To address these issues, we combine detection of noun phrases with the use of WordNet as background knowledge to explore better ways of representing documents semantically for clustering. First, based on noun phrases as well as single-term analysis, we exploit different document representation methods to analyze the effectiveness of hypernymy, hyponymy, holonymy, and meronymy. Second, we choose the most effective method and compare it with the WordNet-based clustering method proposed by others. The experimental results show the effectiveness of semantic relationships for clustering are (from highest to lowest): hypernymy, hyponymy, meronymy, and holonymy. Moreover, we found that noun phrase analysis improves the WordNet-based clustering method.

1. Introduction

Text document clustering techniques were initially developed to improve precision and recall of information retrieval systems by effectively partitioning texts into previously unseen categories [63,69]. More recently, driven by the ever increasing amount of text documents available in document management repositories and on the Internet, the focus has shifted towards providing ways to efficiently browse large collections of documents and reorganize search results for display in a structured, usually hierarchical manner [44].

In recent work, since an ontology is a formal, explicit specification of a shared conceptualization for a domain of interest, it has been applied to various clustering methods for improving their performance [3,4,7,9,11,14,24,48,57,68]. As a comprehensive semantic lexical ontology for the English language, WordNet [39] has been widely used for document clustering [13,16,17,22,23,25,26,40,50,54,55]. However, most existing WordNet-based clustering methods only make use of single-term analysis; they do not perform any phrase-based analysis. A noun phrase, a phrase whose head is a noun optionally accompanied by a set of modifiers used to identify it in detail, can be utilized as a more explicit term for analysis. Additionally, the above WordNet-based methods utilize synonymy to identify concepts and only explore hypernymy to calculate con-

* Corresponding author. Tel.: +82 2 740 8796; fax: +82 2 743 8706.
E-mail address: hgkim@snu.ac.kr (H.-G. Kim).

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.02.019

cept frequencies, without considering other semantic relationships between terms. To clarify some of these semantic relationships, we employ the definitions from WordNet [39]: hypernymy, the semantic relation of being superordinate or belonging to a higher rank or class; hyponymy, the semantic relation of being subordinate or belonging to a lower rank or class; holonymy, the semantic relation that holds between a whole and its parts; meronymy, the semantic relation that holds between a part and the whole.

To address the aforementioned issues, we propose a document representation methodology that takes into account both noun phrases and various semantic relationships. There are a number of semantic relationships that can relate a pair of words. Morris and Hirst [41,42] used various kinds of syntactic categories to compose lexical chains between words. Hirst and St-Onge [20] employed WordNet to study lexical chains for the detection and correction of malapropisms. Kang et al. [29] exploited semantic relationships between words to construct concept clusters for indexing. The previous research shows that hypernymy, hyponymy, holonymy, and meronymy are the most common relationships used to relate two different concepts. Therefore, in this study, we utilize identity and synonymy to identify concepts and consider four other kinds of relations – hypernymy, hyponymy, holonymy, and meronymy – for clustering. Based on noun phrases as well as single-term analysis, we compare the effectiveness of hypernymy, hyponymy, holonymy, and meronymy. Then, the most effective method is compared with the well-known WordNet-based clustering method proposed by Hotho et al. [22]. The main contribution of this paper is to investigate the effectiveness of the different semantic relationships and to determine whether or not noun phrase analysis can improve WordNet-based clustering methods.

The rest of the paper is organized as follows: Section 2 reviews the related work. In Section 3, we describe our document representation methodology based on noun phrases and semantic relationships. Section 4 presents our experimental results. Section 5 discusses the conclusion and future work.

2. Related work

Most of the document clustering in use today is based on the vector space model [1,18,51,52], which is a widely used data model for text classification and clustering. The vector space model represents documents as a feature vector of the terms (words) that occur in the entire document set. Each feature vector contains term weights (usually term frequencies) of the terms occurring in that document. The similarity between documents is measured using one of several similarity measures that are based on such a feature vector, e.g., the cosine measure or the EMD-based (Earth Mover's Distance) measure [67]. Numerous clustering methods have been targeted at single-term analysis, such as GAKREM (Genetic Algorithm K-means Logarithmic Regression Expectation Maximization) [43], graph-based optimization [36], fuzzy clustering [6], LSI (Latent Semantic Indexing) [5,12,15], PLSI (Probabilistic Latent Semantic Indexing) [21], PCA (Principal Component Analysis) [28], SOFM (Kohonen Self-Organizing Map) [34], MDS (Multi-Dimensional Scaling) [27] and so on.

Recently, several phrase-based document clustering methods have been proposed [19,53,70,71]. Zamir and Etzioni [70,71] proposed a phrase-based document clustering approach based on suffix tree clustering. This method basically involves the use of a trie (a compact tree) structure to represent shared suffixes between documents. Based on these shared suffixes they identify base clusters of documents, which are then combined into final clusters based on a connected-component graph algorithm. SanJuan and Ibekwe-SanJuan [53] applied different clustering algorithms to the task of clustering multi-word terms in order to reflect a human-built ontology. Hammouda and Kamel [19] proposed a phrase indexing graph model to calculate document similarity and applied clustering algorithms based on this similarity. In their method, the similarity between documents is based on both single term weights and matching phrase weights. The empirical experimental results indicate that phrase-based methods outperform single term-based clustering methods.

Most of the above clustering methods do not consider applying background knowledge such as ontologies for clustering. Since an ontology is a description of the concepts and relationships that can well represent background knowledge, it has been applied to various clustering methods for improving their performance [3,4,7,9,11,14,24,48,57,68]. Cheng et al. [9] used the graph-based structure of Gene Ontology (GO) [4] to infer the degree of similarity between genes. This knowledge-based similarity metric was applied to detect sets of related genes with biological classifications. Rosaci [48] presented a technique for automatically clustering customer agents by employing ontologies to represent the interests and behaviors of customer agents. Adryan and Schuh [3] proposed GO-Cluster, which uses the structure of the Gene Ontology as a framework for numerical clustering, allowing a simple visualization of gene expression data at various levels of the ontology tree. Wen et al. [68] proposed a framework for genomic information retrieval, which utilized the UMLS ontology to index documents and clustered documents using the fuzzy c-means method. De Luca and Nrnberger [14] used the MultiWordNet ontology to cluster search results by mapping them to semantic classes that are defined by the senses of a query term. Hotho et al. [24] proposed two preprocessing strategies, TES (TErm Selection) and COSA (COncept Selection and Aggregation), which apply ontologies as background knowledge to improve clustering results. Smirnov et al. [57] presented a method for clustering users and their requests, which uses ontologies to improve the semantic inter-operability between companies and customers. Corsi et al. [11] developed the BioPrompt-box, which is an ontology-based clustering tool for searching in biological databases. Breaux and Reed [7] proposed a hierarchical clustering system, which used semantic relationships encoded in the ontology to analyze and visualize documents at conceptually richer levels.

As a large well-constructed lexical database, the WordNet ontology [39] has been widely used for document clustering [13,16,17,22,23,25,26,50,54,55]. In order to calculate the relatedness of two documents, Green [16,17] used WordNet to con-

struct chains of related synsets (lexical chains) from the occurrence of terms in a document. De Buenaga et al. [13] integrated WordNet to complement training information for automatic document categorization. Through word sense disambiguation, Urena-Lopez et al. [65] integrated WordNet and a training collection for text categorization. Rosso et al. [50] used WordNet to study the influence of semantics in the Text Categorization (TC) and Information Retrieval (IR) tasks. Seo et al. [55] described a sense disambiguation method for a polysemous target noun using the context words surrounding the target noun and its WordNet relatives, such as synonyms, hypernyms, and hyponyms. Torra et al. [62] presented a system, the Gambal system, for clustering and visualization of documents based on fuzzy clustering techniques. This system can compute the similarities between documents based on syntactic analysis as well as WordNet and latent semantic analysis. Hung and Wermter [26] proposed three text vector representation models, ESVM (Extended Significance Vector Model), HSVM (Hypernym Significance Vector Model), and HyM (Hybrid Vector Space Model), for neural network based document clustering. In particular, HSVM exploits a semantic relationship from the WordNet ontology and further improves the performance of clustering. Hung and Wermter [25] also proposed a hybrid neural document clustering method using guided self-organization and WordNet, which shows that the hypernym semantic relationship in WordNet complements the neural model, improving classification accuracy and clustering analysis. One of the most prevalent WordNet-based text document clustering methods was proposed by Hotho et al. [22]. They used WordNet as background knowledge to obtain concepts, which is the set of words that have semantic relationships. In their method, synonymy and hypernymy were considered to identify the concepts. Then, the weights of concepts as well as terms were used to represent documents. The experimental results show that their method can improve clustering performance.

However, most existing WordNet-based clustering methods do not explicitly make use of phrase analysis to enhance the quality of clustering. Since phrases, especially noun phrases, have more expressive power than single terms, they can be utilized to represent documents more semantically. Moreover, these WordNet-based methods utilize synonymy to identify concepts and only explore hypernymy to compute concept frequencies; other semantic relationships such as hyponymy, holonymy, and meronymy have not been studied. In this study, we identify noun phrases and explore various semantic relationships to represent documents for clustering. To the best of our knowledge, the idea of exploiting both noun phrases and semantic relationships for clustering has not be researched in detail until now.

3. Exploiting noun phrases and semantic relationships for clustering

3.1. Intuition and overview

In general, documents contain various concepts, and to represent the same concepts, different terms that have the same or related meanings are often used. Suppose there are two documents belonging to the same category. If they only include different terms for the same concept, there can be a problem of clustering them into the same category. To solve this problem, we can exploit semantic relationships between these different terms using a lexical ontology, such as WordNet. First, we identify the concepts that contain different terms having the same or related meanings. Second, the identified concepts themselves can be used to represent documents for clustering. Additionally, since noun phrases are able to provide more detailed information than single term nouns, concepts that consist of noun phrases thus carry more explicit meanings. Taking these factors into account, we explore noun phrases and semantic relationships in order to determine what beneficial effects can be achieved when they are used to represent documents for clustering.

Fig. 1 shows a schematic overview of the proposed methodology. When applied to a given document, the proposed method first calculates the weights of terms *t*. Second, noun phrases *np* are identified and their weights are calculated. Third, the WordNet ontology is used as background knowledge to identify the concepts *c* in the document, and the weights of those concepts are computed. In this study, the *tfidf* (term frequency-inverted document frequency) is applied to calculate the weights of terms, noun phrases, and concepts. Finally, documents are represented as feature vectors using the weights of terms, noun phrases, and concepts, which are then utilized for clustering.

Note that the term 'concept' used here is a set of terms or noun phrases that have identity, or synonym relationships. To study the effectiveness of hypernymy, hyponymy, holonymy, and meronymy, they will be, respectively, included with their

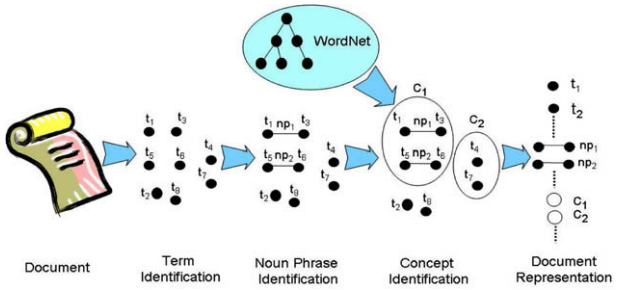

Fig. 1. An overview of the proposed methodology.

related concepts for text document clustering. To understand the process of our methodology, we use a text example as follows.

Example 1. When you feel your *body* in a *bad condition* or you are suffering some *health problems*, you need *exercise* to eliminate the *unfitness* and maintain your *health*. *Yoga* combines the *physical exercises* that stretch and tone your *body* with the *nurture* and *development* of your *emotional health* and *well-being*. This simple *practice* which works your *back*, *hips*, *neck* and *shoulders*, is ideal for relaxing your *mind* and *body* and easing your *unhealthiness*.

In this sample text, the identified nouns or noun phrases are italicized to distinguish them from other words. Fig. 2 illustrates the identified results from the sample text. The terms and the noun phrases, which are linked by the identity or synonym relationships, make up the concepts. With respect to the hypernyms, hyponyms, holonyms, and meronyms, they are used to calculate concept frequencies, respectively. We found that the text includes four noun phrases, five concepts, and a number of single terms. The concepts contained in the sample text are listed as follows:

Concept 1: body, body.

Concept 2: bad condition, unfitness.

Concept 3: health problem, unhealthiness.

Concept 4: exercise, physical exercise, practice.

Concept 5: health, well-being.

Before representing documents as feature vectors, we need to perform preprocessing. There are several standard preprocessing steps, i.e., stopword removal, POS tagging, and word stemming. Stopwords are words considered not to convey any meaning. We use a standard list of 571 stopwords [60] and remove them from the documents. QTagger [47] is used to recognize the part of speech of words in the documents. Words with the same meaning appear in various morphological forms. To capture their similarity, they are normalized into a common root-form, which is called word stemming. We utilize the Porter stemmer [46] to conduct the word stemming in this study.

3.2. Noun phrase identification

To understand the noun phrases in this study, we employ the definition of a noun phrase verbatim from Routledge Dictionary of Language and Linguistics [8]: Noun phrase, grammatical category (or phrase) which normally contains a noun (fruit, happiness, Phil) as its head and which can be modified (specified) in many ways. For identifying noun phrases in the documents, we need to carry out syntactic analysis. Most of the structures of natural language text can be effectively described using Context Free Grammars (CFGs) [10]. CFGs can be used in our approach to detect noun phrases in documents. Numerous methods based on CFGs have been proposed to identify noun phrases [2,31,56,66]. For example, Voutilainen [66] developed a tool named NPtool, which uses constraint grammars [31] to detect English noun phrases. Abney [2] proposed finite-state cascades to parse unrestrict text and Silberztein [56] presented a NLP (Natural Language Processing) platform named NooJ based on finite-state analysis. In this study, to utilize the advantage of statistical methods and CFGs, a statistical parser is employed [32,33], which is a Java implementation of probabilistic natural language parsers using PCFGs (Probabilistic Context Free Grammars) [61]. The empirical experimental results show that the parser can achieve an accuracy of more

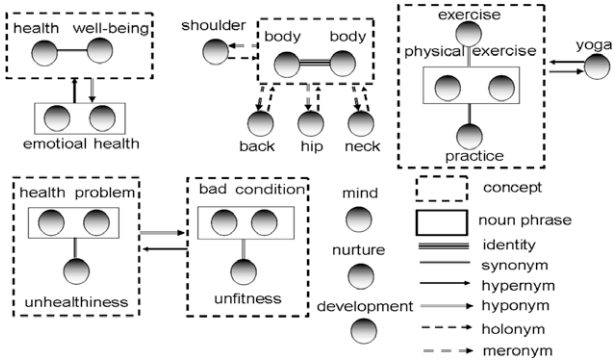

Fig. 2. Identified results from the sample text.

than 80%. The parsing technique here is called partial parsing, as the parser aims to pick up appropriate noun phrases from the document rather than analyzing the whole syntactic structure of the document.

After noun phrases are extracted by the parser [61], we need to obtain the synsets of the noun phrases from WordNet in order to identify the semantic relationships between terms. First, if a noun phrase is directly encoded in WordNet, its corresponding synsets are acquired for the next processing step. Otherwise, there are two cases. First case is to identity the relationship between a single term and a noun phrase. If the noun phrase is composed of several adjectives and a noun that belongs to the synset of the single term, our method determines that the noun phrase is a hyponym of the single term. For example, for term 'restaurant' and noun phrase 'Italian restaurant', our method determines that noun phrase 'Italian restaurant' is the hyponym of term 'restaurant'. Otherwise, only the noun found in the noun phrase is used for the relationship identification and concept frequency calculation. Second case is to identity the relationship between two noun phrases. First, our method additionally constructs two lists to store the single terms found in the noun phrases. Second, we compare the two lists from the beginning and remove the shared terms from the lists recursively. Third, the synsets of the remaining terms in the two lists are acquired from WordNet, respectively. Finally, our method identifies the relationship between the acquired synsets. For example, to identify the relationships between two noun phrases 'steel plant' and 'steel factory', the shared single term 'steel' is removed. Next, the synset of 'plant' and the synset of 'factory' are acquired from WordNet. Finally, our method determines that the two noun phrases are synonyms because the two obtained synsets are the same.

3.3. Document representation using noun phrases and semantic relationships

The background knowledge we will use to explore the concepts in the documents is given through a lexical ontology, WordNet [39]. WordNet consists of over 115,000 concepts (synsets in WordNet) and about 150,000 lexical entries (words in WordNet). In addition, WordNet can provide the morphological capability for stemming the terms. Therefore, when we use WordNet as background knowledge, stemming has only been performed for terms that do not appear as lexical entries in WordNet. To explore what beneficial effects can be achieved for document clustering by considering both noun phrases and semantic relationships, we use the WordNet-based clustering method proposed by Hotho et al. [22] as our baseline and propose our approach for document representation.

3.3.1. Baseline text document representation

In this section we explain the WordNet-based clustering method proposed by Hotho et al. [22]. In the vector space model, a document d is represented as a feature vector $\vec{d} = (tf_{t_1}, \ldots, tf_{t_i})$, where tf_t returns the absolute frequency of term $t \in \mathscr{T}$ in document $d \in \mathscr{D}$, where $\mathscr{D}$ is the set of documents and $\mathscr{T} = \{t_1, t_2, \ldots, t_i\}$ is the set of all different terms occurring in $\mathscr{D}$. In the WordNet-based clustering method [22], first, the concepts in documents are identified as a set of terms that have identity or synonym relationships, i.e., synsets in the WordNet ontology. Then, the concept frequencies are calculated as follows:

$$cf_c = \sum_{t_m \in \mathbb{T}(c)} tf_{t_m} \tag{1}$$

where $\mathbb{T}(c)$ is the set of different terms that belong to concept c. Note that WordNet returns an ordered list of synsets based on a term. The ordering is supposed to reflect how common it is that the term is related to the concept in standard English language. More common term meanings are listed before less common ones. Hotho et al. [22] has showed that using the first synset as the identified concept for a term can improve the clustering performance more than that of using all the synsets to calculate concept frequencies. In this study, we also use only the first synset as the concept for a term for computing concept frequencies.

Hypernyms of concepts can represent such concepts up to a certain level of generality. For example, the concept corresponding to 'meat' can represent the concept corresponding to 'beef'. The concept frequencies are updated as follows:

$$hf_c = \sum_{b \in \mathbb{H}(c,r)} cf_b \tag{2}$$

where $\mathbb{H}(c, r)$ is the set of concepts c_H, which are all the concepts within r levels of hypernym concepts of c. In WordNet, $\mathbb{H}(c, r)$ is obtained by gathering all the synsets that are hypernym concepts of synset c within r levels. In particular, $\mathbb{H}(c, \infty)$ returns all the hypernym concepts of c and $\mathbb{H}(c, 0)$ returns just c.

To weigh the frequency of a term or a concept in a document with a factor that discounts its importance when it appears in almost all documents, the *idf* (inverted document frequency) of term t in document d is proposed as follows [51]:

$$idf_t = log_2 n - log_2 df_t + 1 \tag{3}$$

where n is the total number of documents and df_t is the document frequency of term t that counts how many documents in which term t appears. Consequently, the *tfidf* measure is calculated as the weight of term t:

$$w_t = tf_t \times idf_t \tag{4}$$

Correspondingly, the weight of each concept c in document d is computed as follows:

$$wh_c = hf_c \times idf_c \tag{5}$$

where idf_c is the inverted document frequency of concept c by counting how many documents in which concept c appears. Finally, a document d is represented as a combination of term weights and concept weights, i.e.,

$$\vec{d} = (w_{t_1}, \ldots, w_{t_i}, wh_{c_1}, \ldots, wh_{c_l}) \tag{6}$$

3.3.2. Proposed text document representations

In the present study, based on single-term analysis as well as noun phrase analysis, we exploit four document representations by considering hypernymy, hyponymy, holonymy, and meronymy for computation of concept weights. The purpose is to study what beneficial effects can be achieved for document clustering by using noun phrases and these semantic relationships.

To utilize noun phrases for document representation, first, tf_{np} is calculated as the frequency of noun phrase $np \in \mathscr{NP}$ in document $d \in \mathscr{D}$, where $\mathscr{NP} = \{np_1, np_2, \ldots, np_k\}$ is the set of all different noun phrases occurring in D. Then, the idf factor is used to compute the weight of noun phrase np in document d as follows:

$$wn_{np} = tf_{np} \times idf_{np} \tag{7}$$

where idf_{np} is the inverted document frequency of noun phrase np.

Next, we adapt the baseline method to calculate concept frequencies by considering single terms as well as noun phrases. Let tf_s be the absolute frequency of term or noun phrases $s \in \mathscr{S}$ in document $d \in \mathscr{D}$, where $\mathscr{D}$ is the set of documents and $\mathscr{S} = \{s_1, s_2, \ldots, s_j\}$ is the set of all different terms or noun phrases occurring in $\mathscr{D}$. Let $\mathbb{s}(c)$ be the set of different terms or noun phrases that belong to concept c. To identify a concept containing single terms and noun phrases, our method first acquires the synsets of single terms. Second, the identified noun phrases are examined for presence in the synsets. Finally, the single terms and noun phrases contained in the same synsets are identified as concepts. The concept frequency of c in a document d is defined as

$$sf_c = \sum_{s_m \in \mathbb{s}(c)} tf_{s_m} \tag{8}$$

For example, based on the concepts identified in Example 1 (Section 3.1), the concept frequency of concept 4 (c_4) in the sample text d can be calculated as follows:

$$sf_{c_4} = tf_{s_1} + tf_{s_2} + tf_{s_3} = 1 + 1 + 1 = 3$$

where s_1, s_2, and s_3, are term 'exercise', noun phrase 'physical exercise', and term 'practice', respectively.

As in the baseline method, in order to consider hypernymy for the concept frequency calculation, we use sf_c in place of cf_c to calculate the weight of concept c as follows:

$$wg_c = idf_c \times \sum_{b \in \mathbb{H}(c,r)} sf_b \tag{9}$$

where $\mathbb{H}(c, r)$ and idf_c is defined the same as in the baseline method. To evaluate the effectiveness of hypernymy for clustering, we propose the first method, 'HyperC', to represent a document as follows:

$$\vec{d} = (w_{t_1}, \ldots, w_{t_i}, wn_{np_1}, \ldots, wn_{np_k}, wg_{c_1}, \ldots, wg_{c_m}) \tag{10}$$

Correspondingly, to consider hyponymy for clustering, we calculate the weight of concept c as follows:

$$wp_c = idf_c \times \sum_{b \in \mathbb{P}(c,u)} sf_b \tag{11}$$

where $\mathbb{P}(c, u)$ is the set of concepts c_P, which are all the concepts within u levels of hyponym concepts of c. In particular, $\mathbb{P}(c, \infty)$ returns all the hyponym concepts of c and $\mathbb{P}(c, 0)$ returns just c. To examine whether or not hyponymy can improve the clustering results, we propose the second method, 'HypoC', to represent a document as follows:

$$\vec{d} = (w_{t_1}, \ldots, w_{t_i}, wn_{np_1}, \ldots, wn_{np_k}, wp_{c_1}, \ldots, wp_{c_m}) \tag{12}$$

Similarly, to determine whether or not holonymy and meronymy are useful for clustering, we compute the weight of concept c, respectively, as follows:

$$wl_c = idf_c \times \sum_{b \in \mathbb{L}(c,v)} sf_b \tag{13}$$

$$wm_c = idf_c \times \sum_{b \in \mathbb{M}(c,w)} sf_b \tag{14}$$

where $\mathbb{L}(c, v)$ is the set of concepts c_L, which are all the concepts within v levels of holonym concepts of c; $\mathbb{M}(c, w)$ is the set of concepts c_M, which are all the concepts within w levels of meronym concepts of c. In particular, $\mathbb{L}(c, \infty)$ and $\mathbb{M}(c, \infty)$ return all

levels of holonym concepts and meronym concepts of c, respectively. Both $\mathbb{L}(c,0)$ and $\mathbb{M}(c,0)$ return just c. Finally, we present two methods, 'HoloC' and 'MeroC', to represent a document, respectively, as follows:

$$\vec{d} = (w_{t_1}, \ldots, w_{t_i}, wn_{np_1}, \ldots, wn_{np_k}, wl_{c_1}, \ldots, wl_{c_m}) \tag{15}$$

$$\vec{d} = (w_{t_1}, \ldots, w_{t_i}, wn_{np_1}, \ldots, wn_{np_k}, wm_{c_1}, \ldots, wm_{c_m}) \tag{16}$$

There is no previous research documenting the impact of the depth parameters (r,u,v,w) on clustering performance; therefore, experiments are required to determine their appropriate values.

4. Experimental results

To evaluate the effectiveness of semantic relationships, we conducted a series of experiments using HyperC, HypoC, HoloC, and MeroC, respectively. Then, the most effective method was compared with the WordNet-based method. The experiments were performed on J2SE 5.0, Windows XP, Pentium 4, 3.0 GHz with 2 GB RAM.

4.1. Experimental setup

4.1.1. Test collections

To ensure the experimental results are independent of one special test collection, we used three collections to test our proposed method. The first one is a collection of 200 documents we call 'Reuters-transcribed-set', which was created by selecting 20 files each from the 10 largest classes in the Reuters-21578 collection. The second one is a subset of the full 20-newsgroup collection of USENET newsgroup articles. There are 2000 documents total in this collection with 20 categories. Both of these two test collections are available from the UCI KDD Archive [64].

The third one is the Reuters-21578 test collection [37]. To perform evaluations more objectively, we investigated the technical report [38] and prepared a base corpus taken from the Reuters-21578 collection. All the documents that have precisely one category assigned to them were selected from the Reuters-21578 collection. Additionally, all documents with an empty document body were also discarded. This resulted in a new base corpus containing 8,654 documents. The distribution of category sizes in the new base Reuter corpus is shown in Fig. 3. This illustrates that there are a few categories occurring extremely often: in fact, the two biggest categories contain about two thirds of all documents in the corpus. This unbalanced distribution would blur best results, because even random clustering would potentially obtain a purity value of 30% or more due only to the contribution of the two main categories. To observe the clustering results caused of our method more accurately, we need to place an upper threshold on category-size so that a few categories do not overwhelm other categories. In addition, we should set the thresholds to ensure the number of categories as numerous as possible, so that we can test our method on a test collection containing many categories. In this way, we can more objectively observe whether or not our method improves the clustering performance. Similar to Hotho et al. [22] and Sedding and Kazakov [54], we leverage the number of documents in each category and the number of categories in the whole collection to conduct the sampling process. First, categories with extremely few documents were discarded, with the minimum amount set at 10. Second, we restricted the category-size to a maximum of 100. Categories containing more documents were reduced in size to comply with the defined maximum. We call the resulting test collection 'Reuters-min10-max100'. It is composed of 42 categories and 1927 documents with an average of 45.88 documents per category (standard deviation of 30.614).

4.1.2. Test algorithms

Based on the document clustering techniques comprehensively described by Steinbach et al. [59], we selected three standard document clustering algorithms for testing the effectiveness of each document representation strategy: (1) bisecting K-means clustering [59], (2) K-means clustering [35], and (3) Hierarchical Agglomerative Clustering (HAC) [27,30]. For the

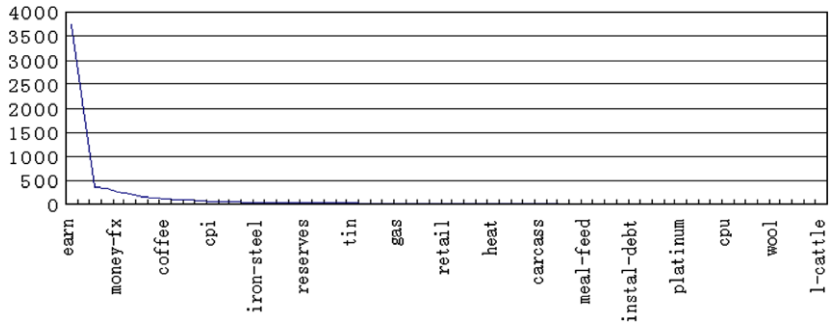

Fig. 3. Category distribution in the new base Reuter corpus.

bisecting K-means clustering, the overall similarity criterion O_p is used to bisect the clusters, i.e., a cluster is bisected so that the resulting 2-way clustering solution optimizes the criterion function:

$$O_p = maximize \sum_{k=1}^{h} \sqrt{\sum_{d_i,d_j \in C_k} sim(\vec{d}_i, \vec{d}_j)} \tag{17}$$

where $\vec{d}_i$ and $\vec{d}_j$ is the feature vector of document d_i and d_j, respectively, C_k is the kth cluster, and h is the total number of clusters.

Both bisecting K-means clustering and K-means clustering utilize the cosine distance as a similarity measure, which measures the similarity of two documents by calculating the cosine angle between them. The cosine distance is defined as follows:

$$sim(\vec{d}_i, \vec{d}_j) = \cos(\vec{d}_i, \vec{d}_j) = \frac{\vec{d}_i \cdot \vec{d}_j}{|\vec{d}_i| \cdot |\vec{d}_j|} \tag{18}$$

where $|\vec{d}_i|$ and $|\vec{d}_j|$ are the lengths of vectors $\vec{d}_i$ and $\vec{d}_j$, respectively, and $\vec{d}_i \cdot \vec{d}_j$ is the dot product of the two vectors.

For the HAC, the UPGMA scheme [27,30] is selected because Steinbach et al. [59] has verified that this method outperforms other HAC methods through empirical experiments. The UPGMA scheme defines the cluster similarity as follows:

$$sim(cluster_i, cluster_j) = \frac{\sum_{d_i \in cluster_i, d_j \in cluster_j} \cos(\vec{d}_i, \vec{d}_j)}{|cluster_i| * |cluster_j|} \tag{19}$$

where d_i and d_j are documents in $cluster_i$ and $cluster_j$, respectively, $|cluster_i|$ is the size of $cluster_i$, and $|cluster_j|$ is the size of $cluster_j$.

4.2. Evaluation measures

To evaluate the quality of the clustering, we adopted two quality measures widely used in the text mining literature for the purpose of document clustering [45,59]. The first one is purity [45], the second one is entropy [59]. Each resulting clustering i from a clustering C of the overall document set D is treated as if it were the result of a query. Each set j of documents from the manual categorization L is treated as if it were the desired set of documents for a query. The precision of clustering $i \in C$ for a given category $j \in L$ is given as follows:

$$Precision_{i,j} = \frac{N_{ij}}{N_i} \tag{20}$$

where N_{ij} is the number of members of category j in cluster i, N_i is the number of members of cluster i, and N_j is the number of members of category j. Precision is the probability that a member of cluster i belongs to category j.

Purity is computed using the maximal precision value for each category j as follows:

$$Purity_{C,L} = \sum_{i \in C} \frac{N_i}{N} \cdot \max_{j \in L} Precision_{i,j} \tag{21}$$

where N is the number of documents in document set D.

The entropy of each cluster C is calculated as:

$$E_i = -\sum_{j \in \mathbb{L}} Precision_{i,j} \cdot \log(Precison_{i,j}) \tag{22}$$

The total entropy for a set of clusters is calculated as the sum of entropies for each cluster weighted by the size of each cluster:

$$Entropy_C = \sum_{i \in C} \left(\frac{N_i}{N} \cdot E_i \right) \tag{23}$$

Both measures have the interval [0,1] as a range. Their difference is that purity measures the purity of the resulting clusters when evaluated against a pre-categorization, while entropy evaluates how homogeneous a cluster is. Basically, we would like to maximize the purity and minimize the entropy of clusters in order to achieve high quality clustering.

4.3. Results and discussion

The clustering results of HyperC, HypoC, MeroC, and HoloC are shown in Figs. 4–6. For Reuters-transcribed-set, the purity of these methods ranges from 0.345 to 0.587; the entropy of these methods ranges from 0.436 to 0.731. For 20-newsgroup, the purity of these methods ranges from 0.613 to 0.985; the entropy of these methods ranges from 0.02 to 0.404. For Reuters-min10-max100, the purity of these methods ranges from 0.326 to 0.618; the entropy of these methods ranges from 0.314 to

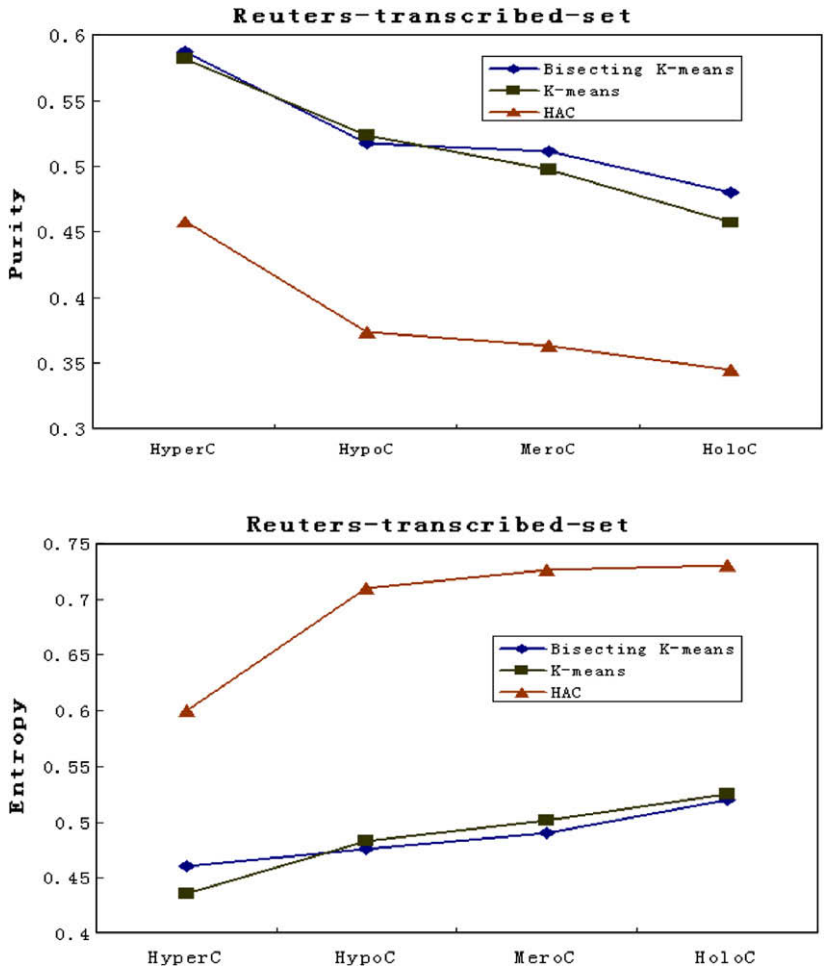

Fig. 4. Purity and entropy of Reuters-transcribed-set.

0.584. From these clustering results, we found that HyperC has the best performance, HypoC the second-best, MeroC the third-best, and HoloC the worst performance, for all of the clustering algorithms. The consistent results mean that hypernymy ranks first in having an effective role for clustering, hyponymy ranks second, meronymy ranks third, and holonymy ranks last.

In general, a set of documents defines a category because they share some general features. These features are expressed by specific terms with different levels of generality. Hypernyms articulate the semantic relationship to a term representing a more general concept. Hyponyms specify relations to subordinate concepts, i.e., with more specificity. Meronyms and holonyms represent a 'part-of' relationship between concepts. The intuition behind the performance of the different semantic relationships can be explained as follows. When documents are categorized by humans, they are usually naturally categorized according to common conceptual features. This intuitively places meronyms and holonyms at a disadvantage for representing documents categorized in such a way, since usually topics that explain a part of something will be included with other topics that it is a part of, and vice versa. The reason that hypernyms perform better than hyponyms could be explained by the fact that document categorization (at least for general collections like those explored in this paper) tends to focus more naturally on the more-general terms rather than the more-specific terms. This corresponds with hypernyms proving more useful at document categorization. According to the clustering algorithms, a set of documents defined as a category shares some general features, which are expressed by concepts in different levels of generality. Hypernyms represent a certain level of generality for a concept, so that they enable the clustering algorithms to more easily find out those general features shared by these documents. We think this is why HyperC outperforms other methods.

Table 1 shows the clustering results as well as the values of r, u, w, and v, which are levels of hypernym, hyponym, meronym, and holonym concepts, respectively. We tested each method's clustering results with values of r, u, w, and v from 1 to 10 and selected the best results. For example, for 20-newsgroup, HperC obtains the best results when $r = 5$ in the K-means algorithm. We found that the values of r, u, w, and v were determined based on the performance of their corresponding

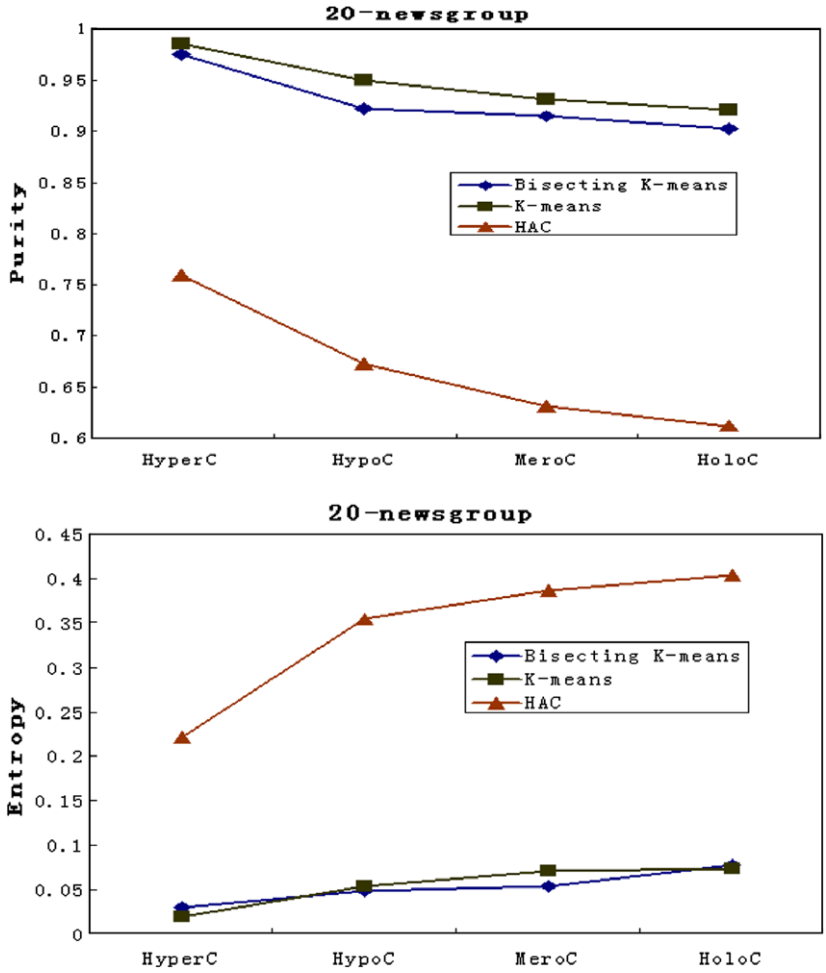

Fig. 5. Purity and entropy of 20-newsgroup.

methods, that is, empirical experiment is required to obtain the appropriate parameter values for different test collections or different clustering algorithms.

Since HyperC obtained the best performance in terms of purity and entropy, we compared HyperC with the baseline method, which utilizes WordNet to exploit synonymy and hypernymy for clustering, in order to gauge the effectiveness of noun phrase analysis. Table 2 shows the comparison of clustering results obtained by the WordNet-based method and the HyperC method, respectively. The two methods' clustering results were tested with different values of *r*, which are levels of hypernym concepts, from 1 to 10 and the best results were selected. The experimental results show that the HyperC method outperforms the WordNet-based method. The percentage of improvement ranges from 0.3% to 10.4% increases in the purity and 0.9% to 9.7% drops in the entropy (lower is better for entropy). The results were consistent among the different test collections.

For the WordNet-based method, Hotho et al. [22] have discussed that the variance of documents within one category is reduced by representation with synonyms and hypernyms, thus improving results of the text clustering. Our HyperC method goes a step further than Hotho's WordNet-based method by identifying noun phrases. Since concepts, which consist of single terms as well as noun phrases, are used to represent documents, different, but semantically similar terms or noun phrases in two text documents can contribute to a good similarity rating if they belong to the same concept. Therefore, the proposed HyperC method has better performance than the WordNet-based method.

In addition, we found that the HyperC method gives more improvements to the bisecting K-means algorithm and the HAC algorithm than it does to the K-means algorithm for the three test collections. We attribute this to the fact that the proposed HyperC method considering noun phrases can support a more exact similarity calculation for the clustering algorithms. In the case of the bisecting K-means algorithm, it splits the clusters iteratively based on overall similarity criteria, but once a cluster is selected to be split, the result can not be changed even if they are not split correctly. The HAC algorithm calculates the similarities between clusters and groups them based on such similarities, and once a document is grouped into a cate-

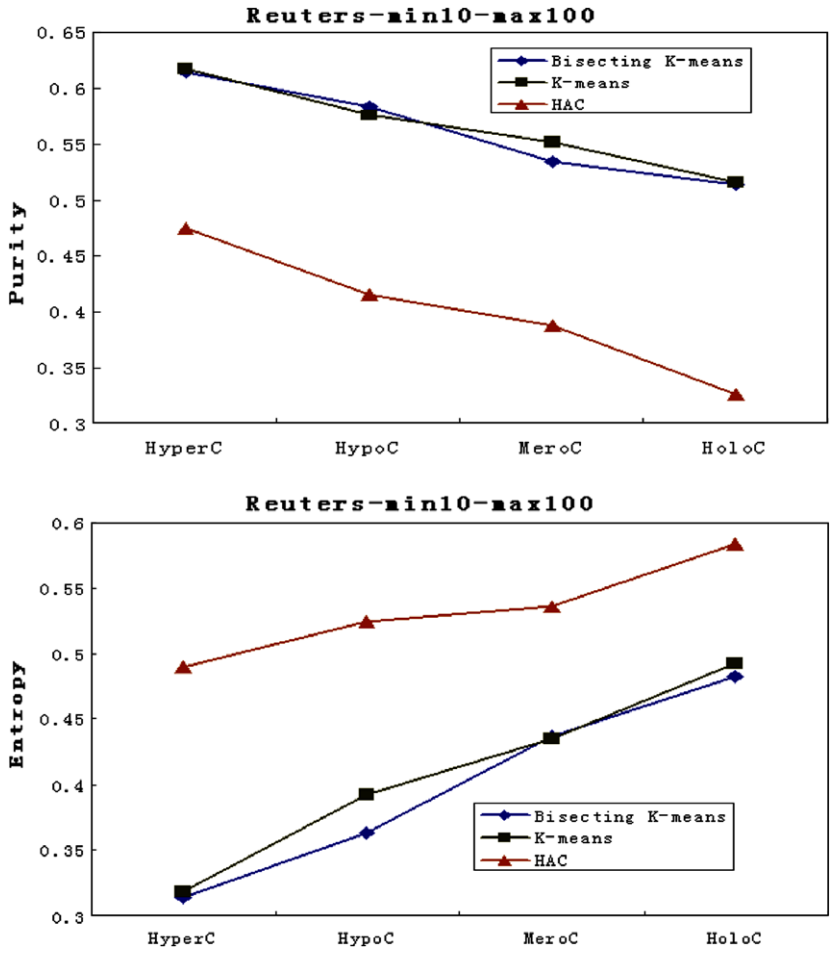

Fig. 6. Purity and entropy of Reuters-min10-max100.

gory inaccurately, the result can not be changed. However, the K-means algorithm can overcome this problem by recomputing the centroid of each cluster and grouping the documents for every iteration. Thus, the accuracy of similarity calculation is more critical for the bisecting K-means algorithm and the HAC algorithm than the K-means algorithm. Thus, we think that our HyperC can show a more striking effect on the bisecting K-means algorithm and the HAC algorithm than it does on the K-means algorithm.

Table 1
Clustering results of HyperC, HypoC, MeroC and HoloC.

	HyperC (r)		HypoC (u)		MeroC (w)		HoloC (v)	
	Purity	Entropy	Purity	Entropy	Purity	Entropy	Purity	Entropy
Reuters-transcribed-set								
Bisecting K-means	0.587	0.46 (5)	0.517	0.475 (3)	0.511	0.49 (1)	0.48	0.52 (5)
K-means	0.582	0.436 (3)	0.523	0.482 (5)	0.497	0.502 (3)	0.457	0.525 (2)
HAC	0.458	0.6 (1)	0.373	0.71 (1)	0.363	0.727 (1)	0.345	0.731 (1)
20-newsgroup								
Bisecting K-means	0.975	0.03 (5)	0.922	0.048 (1)	0.902	0.077 (1)	0.914	0.053 (3)
K-means	0.985	0.02 (5)	0.95	0.053 (7)	0.932	0.071 (1)	0.921	0.073 (7)
HAC	0.76	0.221 (2)	0.673	0.354(3)	0.632	0.387 (5)	0.613	0.404 (1)
Reuters-min10-max100								
Bisecting K-means	0.615	0.314 (5)	0.584	0.363 (3)	0.534	0.437 (1)	0.514	0.483 (5)
K-means	0.618	0.319 (5)	0.576	0.393 (1)	0.552	0.435 (3)	0.516	0.493 (1)
HAC	0.475	0.49 (3)	0.416	0.524 (1)	0.387	0.556 (3)	0.326	0.584 (1)

Table 2
Comparison of the WordNet-based method and the HyperC method.

	WordNet-based method (r)		HyperC method (r)		Comparison	
	Purity	Entropy	Purity	Entropy	Purity (%)	Entropy (%)
Reuters-transcribed-set						
Bisecting K-means	0.483	0.525 (5)	0.587	0.46 (5)	+10.4	−6.5
K-means	0.557	0.448 (5)	0.582	0.436 (3)	+2.5	−1.2
HAC	0.393	0.651 (3)	0.458	0.6 (1)	+6.5	−5.1
20-newsgroup						
Bisecting K-means	0.962	0.047 (7)	0.975	0.03 (5)	+1.3	−1.7
K-means	0.982	0.029 (5)	0.985	0.02 (5)	+0.3	−0.9
HAC	0.668	0.285 (3)	0.76	0.221 (2)	+9.2	−6.4
Reuters-min10-max100						
Bisecting K-means	0.518	0.411 (5)	0.615	0.314 (5)	+9.7	−9.7
K-means	0.588	0.337 (7)	0.618	0.319 (5)	+3	−1.8
HAC	0.431	0.542 (3)	0.475	0.49 (3)	+4.4	−5.2

Although taking the noun phrases into account increases the complexity of the representations, we observed that the three test clustering algorithms still work in a relatively efficient way. Let I be the number of iterations, n be the number of documents to be clustered, k be the number of clusters and d be the number of dimensions. The overall complexities of K-means, bisecting K-means, and HAC are $O(Idkn)$, $O(Idkn)$, and $O((kn)^{3/2}d)$, respectively. Our method increases the value of d by identifying noun phrases and concepts. Suppose $d = d_1 + d_2 + d_3$, where d_1, d_2, and d_3 are the number of single terms, noun phrases, and concepts, respectively. In our experiment, we found that d_2 and d_3 are smaller than d_1. Therefore, $O(Idkn) = O(I(d_1 + d_2 + d_3)kn) = O(Id_1kn)$ and $O((kn)^{3/2}d) = O((kn)^{3/2}(d_1 + d_2 + d_3)) = O((kn)^{3/2}d_1)$. We can see that the computation complexity does not increase greatly.

To conclude, when hypernymy, hyponymy, holonymy, and meronymy are considered for clustering, from highest to lowest, the effectiveness of each semantic relationships is: hypernymy, hyponymy, meronymy, and holonymy. Moreover, since HyperC outperforms the WordNet-based method, we can see that noun phrase analysis improves the WordNet-based method, which means parsing noun phrases enhances the quality of concept-level indexing for clustering.

5. Conclusion and future work

In this paper, we proposed various document representation methods to exploit noun phrases and semantic relationships for clustering. Using WordNet, hypernymy, hyponymy, holonymy, and meronymy were utilized for clustering. Through a series of experiments, we found that hypernymy is most effective for clustering, hyponymy is the second, meronymy is the third, and holonymy is the least. By comparing HyperC and the WordNet-based clustering method, noun phrase analysis was verified to improve upon the WordNet-based clustering methods.

One limitation of the proposed methods is that their performance depends on the comprehensiveness of the noun phrases inspected and the ontology used. In particular, the proposed method will yield good clustering results if the noun phrases can be inspected more precisely and the relations between words or noun phrases are thoroughly represented in the ontology.

We will conduct further research to improve our work in the following ways. First, we will study more NLP (Natural Language Processing) methods to identify noun phrases more accurately. Second, since different levels of hypernyms reflect different levels of generality, we will consider weighing different hypernyms when calculating concept frequencies in order to improve the concept-level representation for clustering. Third, based on the relative usefulness of the semantic relationships, we will combine the four relationships to build the feature vectors for documents. Different weights could be assigned to the related terms, such as meronyms, for computing concept frequencies. In this way, the relationships can be combined and investigated to maximize clustering benefits. Fourth, the clustering results for a variety of domain- and task-specific ontologies can be compared, particularly in the biomedical domain. Good examples are FMA (the Foundational Model of Anatomy) [49], which is known to be ontologically well designed, GO (Gene Ontology) [4], which is most popularly used for biomedical research, and SNOMED CT (Systematized Nomenclature of Medicine – Clinical Terms) [58], which is a practical clinical ontology used by many hospitals.

Acknowledgements

This work was supported by ministry for health, welfare and family affairs, South Korea [A05-0909-A80405-06A2-15010A, Ontology-based EHR (Electronic Health Record) Interoperability Technology Project]. We thank Charles Borchert for his helpful English revision. We also thank the anonymous reviewers whose comments and suggestions led us improve significantly the paper.

References

[1] K. Aas, A. Eikvil, Text categorisation: a survey, Technical Report, Norwegian Computing Center, June 1999.
[2] S. Abney, Partial parsing via finite-state cascades, Nat. Lang. Eng. 2 (4) (1996) 337–344.
[3] B. Adryan, R. Schuh, Gene-ontology-based clustering of gene expression data, Bioinformatics 20 (16) (2004) 2851–2852.
[4] M. Ashburner, C.A. Ball, J.A. Blake, D. Bostein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, et al, Gene ontology: tool for the unification of biology. The gene ontology consortium, Nat. Genet. 25 (2000) 25–29.
[5] M.W. Berry, S.T. Dumais, G.W. O'Brien, Using linear algebra for intelligent information retrieval, SIAM Rev. 37 (4) (1995) 573–595.
[6] C. Borgelt, Accelerating fuzzy clustering, Inform. Sci. (2008), doi:10.1016/ j.ins.2008.09.017.
[7] T.D. Breaux, J.W. Reed, Using ontology in hierarchical information clustering, in: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (Hicss'05), HICSS, Track 4, vol. 04, IEEE Computer Society, Washington, DC, January 2005, pp. 111–112.
[8] H. Bussmann, Routledge Dictionary of Language and Linguistics, 1996.
[9] J. Cheng, M. Cline, J. Martin, D. Finkelstein, T. Awad, D. Kulp, M.A. Siani-Rose, A knowledge-based clustering algorithm driven by gene ontology, J. Biopharm. Stat. 14 (3) (2004) 687–700.
[10] N. Chomsky, Three models for the description of language, IEEE Trans. Inform. Theory 2 (3) (1956).
[11] C. Corsi, P. Ferragina, R. Marangoni, The BioPrompt-box: an ontology-based clustering tool for searching in biological databases, BMC Bioinform. S8 (Suppl 1) (2007) 8. Mar..
[12] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inform. Sci. 41 (6) (1990) 391–407.
[13] M. De Buenaga, J.M. Gmez, B. Daz, Using WordNet to complement training information in text categorization, in: Recent Advances in Natural Language Processing II: Selected Papers from RANLP'97, Current Issues in Linguistic Theory (CILT), vol. 189, John Benjamins, 2000, pp. 353–364.
[14] E.W. De Luca, A. Nrnberger, Ontology-based semantic online classification of documents: supporting users in searching the web, in: Proceeding of European Symposium on Intelligent Technologies EUNITE 2004, Aachen.
[15] P.W. Foltz, Using latent semantic indexing for information filtering, SIGOIS Bull. 11 (1990) 2–3.
[16] S.J. Green, Building hypertext links in newspaper articles using semantic similarity, in: Third Workshop on Applications of Natural Language to Information Systems (NLDB '97), Vancouver, CA, 1997.
[17] S.J. Green, Building hypertext links by computing semantic similarity, IEEE Trans. Knowl. Data Eng. 11 (5) (1999) 713–730.
[18] R. Grishman, Information extraction: techniques and challenges, in: M.T. Pazienza (Ed.), Proceedings of the Summer School on Information Extraction (SCIE-97), LNCS/LNAI, Springer-Verlag, 1997.
[19] K.M. Hammouda, M.S. Kamel, Document similarity using a phrase indexing graph model, Knowl. Inform. Syst. 6 (6) (2004) 710–727.
[20] G. Hirst, D. St-Onge, Lexical chains as representations of context for the detection and correction of malapropisms, in: C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database, The MIT Press, Cambridge, MA, 1998.
[21] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval, 1999.
[22] A. Hotho, S. Staab, G. Stumme, Wordnet improves text document clustering, in: Proceedings of the Semantic Web Workshop of the 26th Annual International ACM SIGIR Conference, Toronto, Canada, 2003.
[23] A. Hotho, S. Staab, G. Stumme, Ontologies improve text document clustering, in: Third IEEE International Conference on Data Mining (ICDM'03), 2003, p. 541.
[24] A. Hotho, A. Maedche, S. Staab, Ontology-based text clustering, in: Proceedings of the IJCAI-2001 Workshop Text Learning: Beyond Supervision, Seattle, USA, August 2001.
[25] C. Hung, S. Wermter, P. Smith, Hybrid neural document clustering using guided self-organization and WordNet, IEEE Intell. Syst. 19 (2) (2004) 68–77.
[26] C. Hung, S. Wermter, Neural network based document clustering using WordNet ontologies, Int. J. Hybrid Intell. Syst. 1 (3, 4) (2004) 127–142.
[27] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., 1988.
[28] J.E. Jackson, A User's Guide to Principal Components, John Wiley and Sons, 1991.
[29] B. Kang, D. Kim, S. Lee, Exploiting concept clusters for content-based information retrieval, Inform. Sci. 170 (2–4) (2005) 443–462.
[30] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley and Sons, Inc., 1990.
[31] F. Karlsson, Constraint grammar as a framework for parsing unrestricted text, in: H. Karlgren (Ed.), Proceedings of the 13th International Conference of Computational Linguistics, Helsinki, vol. 3, 1990, pp. 168–173.
[32] D. Klein, C.D. Manning, Fast exact inference with a factored model for natural language parsing, Adv. Neural Inform. Process. Syst. 15 (2003) 3–10.
[33] D. Klein, C.D. Manning, Accurate unlexicalized parsing, in: Proceedings of the 41st Annual Meeting on Association For Computational Linguistics, Annual Meeting of the ACL, vol. 1, Association for Computational Linguistics, Morristown, NJ, July 2003, pp. 423–430.
[34] T. Kohonen, Self-organization and Associated Memory, Springer-Verlag, 1988.
[35] B. Larsen, C. Aone, Fast and effective text mining using linear-time document clustering, in: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '99, ACM, New York, NY, 1999, pp. 16–22. August.
[36] C. Lee, O.R. Zaïane, H. Park, J. Huang, R. Greiner, Clustering high dimensional data: a graph-based relaxed optimization approach, Inform. Sci. 178 (23) (2008) 4501–4511.
[37] D.D. Lewis, Reuters-21578 Text Categorization Test Collection, Distribution 1.0, 1997.
[38] D.D. Lewis, Readme File of Reuters-21578 Text Categorization Test Collection, Distribution 1.0, 1997.
[39] G.A. Miller, WordNet: a lexical database for English, Commun. ACM 38 (11) (1995) 39–41.
[40] D. Moldovan, A. Novischi, Word sense disambiguation of WordNet glosses, Computer Speech and Language 18 (3) (2004) 301–317.
[41] J. Morris, Lexical cohesion, the thesaurus, and the structure of text, Masters Thesis, Department of Computer Science, University of Toronto, 1988.
[42] J. Morris, G. Hirst, Lexical cohesion computed by thesaural relations as an indicator of the structure of text, Comput. Linguist. 17 (1) (1991) 21C–43C.
[43] C.D. Nguyen, K.J. Cios, GAKREM: a novel hybrid clustering algorithm, Inform. Sci. 178 (22) (2008) 4205–4227.
[44] S. Osinski, D. Weiss, A concept-driven algorithm for clustering search results, IEEE Intell. Syst. 3 (20) (2005) 48–54.
[45] P. Pantel, D. Lin, Document clustering with committees, in: The 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002.
[46] M.F. Porter, An algorithm for suffix stripping, in: K. Sparck Jones, P. Willett (Eds.), Readings in Information Retrieval, Morgan Kaufmann Multimedia Information and Systems Series, Morgan Kaufmann Publishers, San Francisco, CA, 1997, pp. 313–316.
[47] QTagger, <http://web.bham.ac.uk/O.Mason/software/tagger>.
[48] D. Rosaci, An Ontology-based Two-level Clustering for Supporting E-commerce Agents Activities , E-Commerce and Web Technologies, Springer-Verlag, 2005, pp. 31–40.
[49] C. Rosse, J.J. Mejino, A reference ontology for biomedical informatics: the foundational model of anatomy, J. Biomed. Inform. 36 (2003) 478–500.
[50] P. Rosso, E. Ferretti, D. Jimenez, V. Vidal, Text categorization and information retrieval using wordnet senses, in: Proceedings of the Second InternationalWordNet Conference (GWC), 2004.
[51] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620.
[52] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., New York, 1983.
[53] E. SanJuan, F. Ibekwe-SanJuan, Phrase clustering without document context, Adv. Inform. Retrieval (2006) 496–500.
[54] J. Sedding, D. Kazakov, WordNet-based text document clustering, in: COLING 2004 Third Workshop on Robust Methods in Analysis of Natural Language Data, Geneva, Switzerland, August 29th, 2004.

[55] H. Seo, H. Chung, H. Rim, S. Myaeng, S. Kim, Unsupervised word sense disambiguation using WordNet relatives, Computer Speech and Language 18 (3) (2004) 253–273.
[56] M. Silberztein, An alternative approach to tagging, in: Proceedings of the 12th International Conference on Applications of Natural Language to Information Systems, NLDB 2007, pp. 1–11.
[57] A. Smirnov, M. Pashkin, N. Chilov, T. Levashova, A. Krizhanovsky, A. Kashevnik, Ontology-based users and requests clustering in customer service management system, in: Proceedings of International Workshop on Autonomous Intelligent Systems: Agents and Data Mining, St. Petersburg, Russia, 2005.
[58] M.Q. Stearns, C. Price, K.A. Spackman, A.Y. Wang, SNOMED clinical terms: overview of the development process and project status, in: Proceedings of the AMIA Symposium, 2001, pp. 662–666.
[59] M. Steinbach, G. Karypis, V. Kumar, Comparison of document clustering techniques, in: KDD Workshop on Text Mining, 2000.
[60] StopWord List, <http://www.lextek.com/manuals/onix/stopwords2.html>.
[61] The Stanford Parser, <http://nlp.stanford.edu/software/lex-parser.shtml>.
[62] V. Torra, S. Miyamoto, S. Lanau, Exploration of textual document archives using a fuzzy hierarchical clustering algorithm in the GAMBAL system, Inform. Process. Manag. 41 (3) (2005) 587–598.
[63] C.J. Van Rijsbergen, Information Retrieval, second ed., Butterworths, London, 1979.
[64] UCI KDD Archive, <http://kdd.ics.uci.edu/>.
[65] L.A. Urena-Lopez, M. Buenaga, J.M. Gomez, Integrating linguistic resources in TC through WSD, Comput. Humanities 35 (2) (2001) 215–230.
[66] A. Voutilainen, NPtool, a detector of English noun phrases, in: Proceedings of the Workshop on Very Large Corpora, 1993, pp. 48–57.
[67] X. Wan, A novel document similarity measure based on earth mover's distance, Inform. Sci. 177 (18) (2007) 3718–3730.
[68] J. Wen, Z. Li, X. Hu, Ontology based clustering for improving genomic IR, in: Proceedings of the 20th IEEE International Symposium on Computer-Based Medical Systems, CBMS, IEEE Computer Society, Washington, DC, June 2007, pp. 225–230.
[69] P. Willett, Recent trends in hierarchical document clustering: a critical review, Inform. Process. Manag. 24 (5) (1988) 577–597.
[70] O. Zamir, O. Etzioni, Web document clustering: a feasibility demonstration, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98), August 1998, pp. 46–54.
[71] O. Zamir, O. Etzioni, Grouper: a dynamic clustering interface to Web search results, in: Proceedings of the Eighth International World Wide Web Conference, Toronto, Canada, May 1999.