Knowl Inf Syst
DOI 10.1007/s10115-010-0370-4

REGULAR PAPER

On ontology-driven document clustering using core semantic features

Samah Fodeh · Bill Punch · Pang-Ning Tan

Received: 10 December 2009 / Revised: 6 September 2010 / Accepted: 26 November 2010

Abstract Incorporating semantic knowledge from an ontology into document clustering is an important but challenging problem. While numerous methods have been developed, the value of using such an ontology is still not clear. We show in this paper that an ontology can be used to greatly reduce the number of features needed to do document clustering. Our hypothesis is that *polysemous* and *synonymous* nouns are both relatively prevalent and fundamentally important for document cluster formation. We show that nouns can be efficiently identified in documents and that this alone provides improved clustering. We next show the importance of the polysemous and synonymous nouns in clustering and develop a unique approach that allows us to measure the information gain in disambiguating these nouns in an unsupervised learning setting. In so doing, we can identify a *core* subset of semantic features that represent a text corpus. Empirical results show that by using core semantic features for clustering, one can reduce the number of features by 90% or more and still produce clusters that capture the main themes in a text corpus.

Keywords Clustering · Information gain · Semantic features · Ontology · Dimensionality reduction

1 Introduction

Document clustering [20,22,34,42,45,47] is the unsupervised process of partitioning a set of documents into groups containing similar topics. Despite the major advances made over

S. Fodeh (✉)
Yale University, New Haven, CT, USA
e-mail: samah.fodeh@yale.edu

B. Punch · P.-N. Tan
Michigan State University, East Lansing, MI, USA
e-mail: punch@msu.edu

P.-N. Tan
e-mail: ptan@msu.edu

Published online: 29 January 2011

Table 1 Examples of blog posting titles from two topics: security and medicine

#	Topic	Blog posting title
1	Security	Czech experts uncover global virus network
2	Security	The great cure for computer flu
3	Security	How to Keep away from a Trojan Malware
4	Medicine	Swine flu related to 1918 pandemic virus
5	Medicine	Special H1N1 vaccine for pregnant women is available
6	Medicine	A global cure for malaria

the past thirty years, the rapidly growing availability of large tracts of unstructured textual data, such as blog postings, emails, online review forums, and discussion board messages, has created a need for improved text clustering especially in an open-domain environment. Besides the sheer size and high dimensionality of the datasets, the documents are often written by many authors, each with their own writing style, including vocabulary. The lexical variations and diverse nomenclature used, even among documents with similar topics, can adversely affect the performance of clustering algorithms. For example, consider the titles of six blog postings shown in Table 1. The first three titles are from the topic of Internet security while the last three titles are related to medicine. Existing clustering algorithms will have difficulty partitioning them correctly based on only the words that appear in the titles. This is because none of the blog postings share any words in common with other blog postings from the same topic. Furthermore, the titles of the first and fourth blog postings contain the same word "virus" and thus are likely to be assigned to the same cluster even though they belong to different topics.

Unsupervised clustering of text corpus is therefore a challenge due to the diversity of vocabulary used and the general lack of "guidance" as to which documents belong to which cluster. An example of such guidance is the *must-link* and *must-not-link* constraints used in semi-supervised clustering [4]. Although such guidance has been shown to improve clustering performance, creating the constraints is both time-consuming and error-prone as the constraints themselves must be done by hand. Furthermore, they are not a general solution that can be used across many datasets as they are specific to the dataset itself. The lack of such general guidance makes the partitioning task an ill-posed problem. There is no single methodology known to produce an optimal clustering across all datasets.

In order to provide a more general form of cluster guidance, researchers have begun to investigate alternative clustering approaches that automatically incorporate *background knowledge* from external sources to guide the partitioning [14,31]. By background knowledge we mean general domain information that can be exploited to improve clustering.

One general approach to encoding background knowledge is the use of an *ontology*. An ontology is a hierarchy of concepts related via domain-specific relations. In particular, there has been much work done on the use of *semantic ontologies* as an aid to clustering. In a semantic ontology, a concept is a word's *sense*. The relationships between senses are typically one of class–subclass. WordNet (http://wordnet.princeton.edu), and MeSH (http://www.nlm.nih.gov/mesh) are examples of existing semantic ontologies currently used in document clustering. These ontologies can be used to replace the original words/terms in a document by the most appropriate concept as dictated by the surrounding *context* of a document. This process is known as word-sense disambiguation (WSD). WSD, if performed correctly, can overcome two fundamental problems in text clustering: *polysemy* and *synonymy*. A polysemous term is a term that has multiple, disjoint meanings. For example, the term "virus" in Table 1 is a polysemous term that maps to multiple meanings, such as "computer malware"

or as "ultramicroscopic infectious agent that replicates itself within cells of living hosts", depending on the content of the document. On the other hand, a synonymous term maps to the same concept as different words in the document. For example, the terms "swine flu" and "H1N1" map to the same concept. Polysemy and synonymy have an effect on increasing or decreasing the semantic similarity between documents, which in turn affects document clustering. With the help of an ontology, many researchers have used WSD to disambiguate terms, replacing those terms with their contextually appropriate concepts. It is these concepts can then be used to cluster the documents.

This approach, however, has not proven to be as useful as first hoped. For example, approaches that expand the feature space by replacing a term with its potential concepts only increases the feature space without necessarily improving clustering performance [14]. Approaches that select the *best* concept to replace the term suffer from the potential inaccuracy of selection, leading to poorer cluster performance.

Our goal in this work is to show a different utility for word disambiguation: feature reduction that both maintains, or even improves, clustering and identification of a *theme* of a document based on the features identified. We do this in stages. First, we discuss a simple look-up procedure that identifies the *nouns* in a document. This procedure is a simple lookup for the stemmed word in the WordNet noun database. If the word *could* be a noun, we count it as such. Our analysis shows that such a simple procedure often produces results that are comparable to (or occasionally better than) those using WSD. Second, we show that polysemous and synonymous nouns are relatively prevalent (accounting for nearly 30% of all nouns) in a text corpus. Third, we show that the polysemous and synonymous nouns strongly participate in the formation of the clusters by examining their influence on the pair-wise document similarity matrix. Specifically, we observe a high correlation between the similarity matrix computed using all nouns and the similarity matrix computed using only polysemous and synonymous nouns. The observed correlation is significantly higher than that obtained using a random subset of nouns to compute the document similarity matrix. Finally, we show that it is possible to extract a subset of the disambiguated nouns (known as the core semantic features) that are "cluster-aware". To do this, we introduce an information gain measure to determine the contribution of each disambiguated noun to clustering. This is a novel contribution since the information gain measure is computed without knowing the ground truth labels of the documents. We show that the core semantic features derived from this approach capture the main themes of a text corpus. Our empirical results show that the core semantic features help to reduce the number of features by at least 90% in tested datasets, yet produce clusters comparable to (or occasionally better than) those clustering using all the nouns or all the mapped concepts from a given ontology.

A side effect of using this greatly reduced core semantic features is that some documents are no longer represented (they have no features in the new feature set). To cluster these documents, we use a centroid-based method to assign the "uncovered" documents to their respective clusters. The centroids are initially identified by computing the mean values of the feature vectors for "covered" documents. Our approach is unique in that it performs subspace clustering using a set of core semantic features and a subset of the text corpus (i.e., on "covered" documents containing the core semantic features). Although the percentage of uncovered documents can be very high, the centroid-based method does not appear to significantly degrade the clustering performance on the majority of the datasets investigated in this study. Our centroid-based semantic clustering approach may potentially be used to cluster new documents by assigning them to their nearest centroids. Furthermore, the core semantic features act as an effective, human understandable, summary of the clusters.

The remainder of this paper is structured as follows: Sect. 2 describes some of the related work in the area of ontology-driven document clustering. Section 3 summarizes the results reported in the literature on the value of using ontology for clustering. Section 4 presents a simple preprocessing approach for noun identification using WordNet. The effects of polysemy, synonymy, and WSD on document clustering are discussed in Sect. 5. Sections 6 and 7 describe our proposed methodology for extracting core semantic features using the information gain measure, Sect. 8 presents our experimental evaluation, and finally, in Sect. 9, we conclude the paper.

2 Related work

An ontology defines a set of concepts and their relationships within a domain. To date, ontological-driven techniques have been developed for a variety of text mining tasks including text categorization [12,17,30,32,40], language modeling [1,8], information extraction [7], and information retrieval [5,25,38] problems.

This paper focuses on the use of semantic ontology for text clustering. Current approaches can be divided into two major categories, namely, concept mapping and embedded methods. Concept mapping methods simply replace each term in a document by its corresponding concept(s) extracted from an ontology before applying the clustering algorithm. These methods are appealing because they can be applied to any clustering algorithm. Furthermore, the mapping of terms into concepts incurs only a one-time cost, thus allowing the clustering algorithm to be invoked multiple times (for different cluster initialization, parameter settings, etc) without the additional overhead of re-creating the concepts. However, their main limitation is that the quality of the clusters is highly dependent on the correctness of the WSD procedure used. Embedded methods, on the other hand, integrate the ontological background knowledge directly into the clustering algorithm. This would require modifications to the existing clustering algorithm, which often leads to substantial increase in its both runtime and memory requirements. Instead of performing WSD explicitly, these methods assume the availability of a distance/similarity matrix for all pairs of words in a text corpus computed based on the WordNet's concept hierarchy. Since a word can be mapped to several synsets, multiple paths may exist between any two words and the algorithm has to decide which path to use when computing the distance measure. Thus, embedded methods are still susceptible to incorrect mapping issues related to WSD.

Techniques that employ the concept mapping approach for clustering with WordNet as its semantic ontology include [14,29,31], and [36]. Hotho et al. [14] constructed a bag of words feature representation that contains the original terms and their disambiguated senses and hypernyms up to a designated level τ. Sedding and Kazakov [31] extended this work by exploring the benefits of disambiguating the terms using their part of speech tags. The main limitation of both approaches is the increase in dimensionality of the data, which is partly due to the disambiguation of each polysemous term into multiple word senses (also known as synsets) from WordNet. Hotho et al. [14] and Sedding and Kazakov [31] also augmented the WordNet retrieved synsets with the original nouns present in the documents and a set of hypernyms obtained from the higher levels of WordNet's semantic relation hierarchy, thereby increasing the dimensionality considerably. To overcome this limitation, [29] considered two approaches to reduce the number of features. The first approach restricted the feature set to 41, manually selected, high-level lexical categories from WordNet (e.g., Animal, Event, Location, and Person) while the second approach identified groups of similar words based on their hypernym and hyponym relations to the selected lexical categories and used the groups

to form the feature set. Termier et al. [36] applied hierarchical clustering on the concepts to reduce the dimensionality of the data before applying the clustering algorithm. In both [29] and [36], the features are selected without considering their potential impact on clustering, unlike the core semantic feature approach proposed in this paper. Furthermore, the WSD procedure may introduce erroneous features that have adverse effects on clustering. To improve the robustness of ontological-driven clustering algorithms, [11] developed an ensemble method that combines clustering solutions obtained from the nouns and semantic concepts. Their results showed that the ensemble method is effective when applied to data sets where sense information is useful to disambiguate words that are used in different contexts.

Embedded methods were investigated by [19], who employed the WordNet ontology to parameterize the distance measure used in clustering. Specifically, they computed the mutual information between words based on a combination of their term frequencies and the semantic relationships between their senses in WordNet. The mutual information matrix is then used to define the distance measure for k-means clustering. The approach used by [19] is analogous to the idea of using semantic kernels in text clustering. This idea, which has been investigated by several authors including [2] and [10], seeks to map the documents into a latent semantic space to enable the computation of distance between two document vectors based on the similarity of their words. Though it is applicable to WordNet ontology, neither [2] nor [10] have used a semantic ontology in their work. Instead, the semantic similarity between terms is computed based on their co-occurrence in the text corpus. Since embedded methods do not explicitly identify the semantic features of a document, they cannot be used to interpret the resulting clusters. Furthermore, the method does not reduce the dimensionality of the original data.

In addition to WordNet, there have been growing efforts to utilize Wikipedia as the source of ontological knowledge for document clustering. For example, [3] investigated the problem of clustering short text in news feeds and blog posts using Wikipedia. Concepts from Wikipedia were first extracted using a concept mapping approach developed by [13] and then augmented with the original terms before applying standard clustering algorithms such as bisecting k-means and hierarchical clustering. Hu et al. [16] extended the approach by generating a hierarchical representation of the documents using natural language processing techniques along with ontological concepts from both Wikipedia and WordNet. They also applied a feature selection approach to reduce the dimensionality of the data. However, similar to the WordNet concept mapping approaches described earlier, their feature selection step does not consider the effect of the selected features on clustering. Furthermore, unlike WordNet, Wikipedia is not a structured thesaurus and thus cannot be easily used to handle synonymy and polysemy issues. Hu et al. [15] developed a method to fully utilize the structural relationships in Wikipedia (including synonymy, polysemy, and hypernymy relations) to enhance the mapping of terms in text documents to the Wikipedia concepts. The mapped concepts are then clustered using traditional k-means algorithm. However, since it does not perform any dimensionality reduction, the number of Wikipedia concepts used for clustering can be very large.

3 Evaluating the value of using an ontology

Table 2 summarizes the results reported by some of the previous works on the value of using an ontology for clustering. The empirical results have been inconclusive—some researchers suggested that the use of an ontology is helpful for clustering, while others have reported that the ontological concepts adds no value and sometimes impairs performance of document clustering. Wang et al. [39] had performed a comparative study to examine the effect of using ontological concepts in document clustering. They evaluated different clustering

Table 2 Results reported from previous work on incorporating ontological background knowledge in document clustering

Reference	Ontology	Data	Measure	With ontology	Without ontology
[14]	WordNet	Reuters	Purity	**0.61**	0.57
[41]	WordNet	AI journal	Entropy	**0.56**	1.45
		Journ. of Ecology	Entropy	**0.52**	1.41
		Economic Journ.	Entropy	**0.79**	1.55
		Historical Abs.	Entropy	**1.09**	1.54
[29]	WordNet	Reuters_{13}	F-measure	**0.60**	0.49
		Reuters_{25}	F-measure	**0.58**	0.50
[27]	WordNet	LA Times	Precision	0.73	**0.79**
			Recall	**1.0**	0.74
[36]	WordNet	Reuters	Accuracy	0.81	**0.96**
[31]	WordNet	Reuters	Entropy	0.74	**0.59**
			Purity	0.47	**0.58**
[11]	WordNet	Reuters	Entropy	1.19	**0.97**
		20Newsgroups	Entropy	0.88	**0.86**
[3]	Wikipedia	Google News	Accuracy	**0.85**	0.63
[46]	MeSH	Medline	Entropy	0.12	0.13
			Purity	0.93	0.92
			F-measure	**0.84**	0.80
[15]	Wikipedia	Reuters	Purity	**0.65**	0.60
		OHSUMED	Purity	**0.45**	0.41

For each row, the best clustering performance is shown in bold

methods, such as, k-means, bisecting k-means, and hierarchical clustering. Though their experiments showed that hierarchical clustering using concepts appear to give better results, the improvement is not always significant.

One major difficulty in comparing the previously published results is their lack of uniformity, especially in terms of the benchmark data and baseline algorithms used. Furthermore, the ontology-driven methods are used in conjunction with other preprocessing techniques (e.g., part-of-speech tagging and dimensionality reduction), which makes it hard to determine whether the differences in clustering performance are due to the ontology used or the effect of other factors. Aside from these differences, the application of an ontology may alter clustering results in many ways. For example, WordNet ontology can be used to help filter uninformative terms (such as verbs and adjectives), to handle polysemy and synonymy issues, and to capture the main themes of documents using higher level categories in WordNet's concept hierarchy. Each of these factors (removal of uninformative terms, disambiguation of polysemous/synonymous terms, and replacement of terms with their categories) has varying influence on the clustering results. We will analyze the differential impact of each factor on cluster quality and dimensionality of the feature space in the next two sections.

4 Using nouns as cluster features

An ordinary document may contain some non-descriptive and redundant terms that may affect the clustering performance. Therefore, the documents must be preprocessed to remove these

uninformative terms. Our preprocessing steps include stopwords removal, noun extraction, and stemming. Stopwords are non-descriptive words in a text. Examples are pronouns, determinants, and numbers. Finally, we identify the nouns in the documents and use those as the features. Note that we do *not* use any parsing techniques to identify the Part of Speech (PoS) of any word. Rather, we use WordNet to identify whether a word *potentially* could be a noun by checking if the stemmed word exists in the WordNet noun database. In this way, we identify "nouns" without the overhead of parsing sentences in a document. Though not thoroughly accurate, it is relatively fast. We also used the morphy stemmer from WordNet as provided by the Natural Language ToolKit (NLTK) (http://www.nltk.org) package in order to stem the nouns. The set of stemmed nouns that resulted from preprocessing will form the feature vector for each document.

To evaluate the effectiveness of our WordNet-based noun identification method, we used samples generated from two benchmark datasets: Reuters-21578 [23] and 20newsgroups (http://people.csail.mit.edu/jrennie/20Newsgroups). The details of each dataset are presented in Sect. 8. We use spherical k-means as our clustering algorithm. In addition to being scalable and fast, spherical k-means is a popular baseline method used by previous researchers on ontology-based clustering algorithms.

Table 3 shows a comparison of clustering performance between using all the words (excluding stopwords) as features versus using only the stemmed nouns (as identified by WordNet) as features. Overall, we observed more than 50% reduction in the number of features after preprocessing the data. For 13 out of the 19 datasets, the decrease in the feature space size helped improve the document clustering performance. In fact, 5 of the datasets showed at least 15% improvement in cluster purity when stemmed nouns are used. The results in Table 3 suggested that using WordNet to identify nouns not only reduced the feature set dimensionality but also improved the cluster purity for many of the datasets.

With the exception of [11], most of the previous studies shown in Table 2 used the clustering of all terms (after stopword removal) as their baseline and compared the performance of their ontology-driven methods against it. Clearly, the analysis in this section suggests such a comparison may not be sufficient because one could achieve quite significant improvement in cluster purity by simply removing terms not identified as nouns by WordNet. Unless the results for noun identification are also reported, it is hard to tell whether it is necessary to apply complex operations such as WSD to improve clustering performance.

5 Effect of polysemy and synonymy on clustering

This section analyzes the effect of incorporating semantic relations (such as polysemy and synonymy) into document clustering. Specifically, we compare the relative changes in clustering performance after disambiguating the nouns against removal of non-nouns as described in the previous section. We also investigate the importance of polysemous and synonymous nouns in the formation of clusters.

5.1 Context-based word-sense disambiguation

We employ the WSD procedure described in [11], which replaces the nouns by their most appropriate senses as used in the context of the document. For example, consider the term "cat", which has eight meanings as a noun in WordNet. If it is used in a document that contains other terms such as "kitten", "Persian", and "pet", we expect its sense refers to the "feline mammal" sense of cat, and not one of the other seven (such as a farm machine or

Table 3 Comparison between clustering using terms after stopwords removal versus nouns

Dataset	Terms after stopword removal		Nouns after stopword removal & stemming		% Improvement in purity
	# Terms	Purity	# Nouns	Purity	
B401	8,056	0.771	3,509	0.958	25
B402	7,363	0.763	3,018	0.974	29
B403	10,229	0.750	4,305	0.907	21
B404	11,207	0.975	4,557	0.979	1
B405	8,304	0.926	3,629	0.960	4
B406	10,468	0.961	4,295	0.957	0
B407	9,130	0.650	4,159	0.650	0
B408	10,803	0.970	4,717	0.957	−1
B409	9,375	0.647	4,244	0.635	−1
B410	10,875	0.560	4,715	0.887	59
B411	10,283	0.965	4,563	0.952	−2
B412	11,205	0.645	4,997	0.655	2
B413	9,435	0.640	4,340	0.657	3
B414	7,728	0.867	3,545	0.907	4
B415	6,300	0.862	2,926	0.890	3
B416	7,147	0.935	3,135	0.940	0
Multi10	13,615	0.440	5,177	0.538	15
Multi6	45,112	0.461	13,801	0.486	5
Reuters	14,261	0.603	5,922	0.650	8

a particular type of X-ray). However, if the term "cat" appears in a document that contains other nouns such as "construction", "builder", and "home", its sense most likely refers to a "Caterpillar", which is a large tracked vehicle used for moving earth in construction and farm work.

More formally, our WSD approach can be described as follows. Let $S_i = \{s_{i1}, s_{i2}, \ldots, s_{ik}\}$ denote the set of all senses associated with the noun t_i according to the WordNet ontology. Given a document d, we determine the most appropriate sense $\hat{s}_i$ of a noun t_i by computing the sum of its similarity to other noun senses in d, i.e.,

$$\hat{s}_i = \arg\max_{s_{il} \in S_i} \sum_{t_j \in d} \max_{s_{jm} \in S_j} \delta(s_{il}, s_{jm}) \tag{1}$$

where $\delta(s_p, s_q)$ is the WordNet similarity between two senses s_p and s_q. Furthermore, since the senses of a given noun in the WordNet hierarchy are arranged in descending order according to their common usage, we restrict our consideration to the first three senses of each noun. The number of senses to examine, three, was chosen empirically, based on results comparing the use of all senses to the top three. Using only the top three senses yielded similar clustering results at a reduced computation cost [11].

This approach is still computationally expensive. However, it is only a one-time cost and need only be performed once on each document. Once extracted, the senses are stored to be

used at a later time. Furthermore, this disambiguation process can be done in parallel since the task is done independently for each document.

Several similarity measures $\delta(s_p, s_q)$ can be used in Eq. (1) to compute the similarity between two senses [24,44] and [18]. We experimented with some measures such as Path Length similarity measure (http://wordnet.princeton.edu,) and Wu-Palmer similarity measure; however, we noticed that Wu-Palmer gave better results. Therefore, we continued to use it in our experiments. Wu-Palmer is a path-based measure which considers the shortest path between the concepts and their depth in the hierarchy. It computes the similarity between two senses by finding the least common subsumer (LCS) node that connects their senses. The LCS of two senses, s_p and s_q, is the lowest common node between the paths of s_p and s_q from the root of WordNet hierarchy (i.e. the LCS has the maximum depth between all the common subsumers). Once the LCS has been identified, the Wu-Palmer distance is given by the following equation:

$$\delta_{\text{Wu\_Palmer}}(s_p, s_q) = \frac{2d}{L_p + L_q + 2d} \tag{2}$$

where L_p is the path length between s_p and the LCS, L_q is the path length between s_q and the LCS and d is the depth of the LCS from the root. The depth of a node s is computed as:

$$d(s) = \text{path\_length}(s, \text{root}) + 2 \tag{3}$$

Figure 1 shows part of the WordNet hierarchy that associates the first sense of the nouns "dog" and "cat". There are two common subsumers between both senses which are "animal" and "carnivore". Wu-Palmer strategy favors the Lowest Common Subsumer in the hierarchy between the two senses. In this case, "carnivore" is the LCS because the depth of "carnivore" is 13 whereas the depth of "animal" is 8 in the hierarchy. Computing Wu-Palmer similarity score using "carnivore" as the LCS give a higher similarity between "dog" and "cat" .866 compared to .64 when using "animal" as a common subsumer.

$$\delta_{\text{Wu\_Palmer}}(\text{dog}, \text{cat}) = \frac{2 \times 13}{2 + 2 + 2 \times 13} = 0.866 \tag{4}$$

5.2 Evaluating the effect of word-sense disambiguation

Equation (1) was used to identify the semantic features (concepts) that correspond to the original nouns. The semantic representation of the documents is then established by creating a binary vector for each document where the existence of the concept in a document is denoted by 1 and 0 otherwise. Spherical k-means with cosine similarity measure is then applied on all the datasets. The cluster purity results are shown in the second last column of Table 4. For datasets B407, B409, B412, and B413, there is significant improvement in cluster purity after replacing the nouns by their semantic features. However, in 12 out of the 19 datasets, the clustering results after applying WSD is within $\pm 2\%$ of the cluster purity obtained using non-stopword nouns (column 5), though these results are still better than using all non-stopword terms (column 3). These results suggest applying WSD, which is an expensive operation, often yields comparable performance as removing non-noun terms using WordNet. The number of concepts extracted (column 6) is also higher than the number of nouns (column 4). Therefore, to demonstrate their effectiveness, ontology-driven clustering algorithms must not only compare their results against using all the terms, which is the norm in most of the previous studies, they must also compare against using the nouns only.

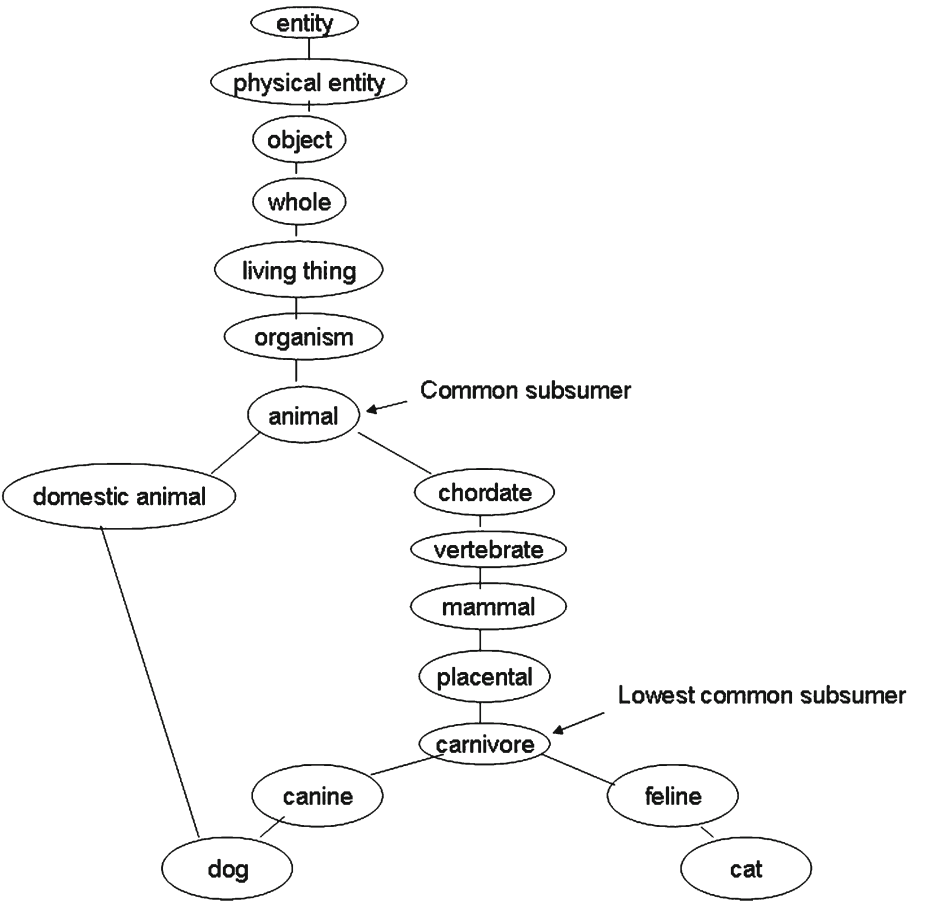

Fig. 1 The semantic network between the first senses of dog and cat

5.3 Polysemy

A noun that has multiple meanings is called a polysemous noun. In WordNet, these multiple meanings are called senses. Having polysemous nouns within a document potentially increases clustering error. This is because the "correct" meaning, based on the context of the document, is not reflected by each noun's frequency in a document. However, disambiguating polysemous nouns by their corresponding "correct" senses and using the sense in clustering has a noticeable effect. This disambiguation directly affects the feature space and changes the similarity between those documents, as shown in the following example.

Example 1 Let $D = \{d_1, d_2, d_3\}$ be three documents and $\{n_1 = artificer, n_2 = beadwork, n_3 = artist\}$ be the corresponding nouns. Suppose n_1 has 3 possible senses ($s_1 = inventor, s_2 = artisan, s_3 = armor$), n_2 has 2 senses ($s_4 = beading, s_5 = astragal$) and n_3 has 1 sense ($s_6 = artist$). The Wu-Palmer similarity between s_1 (inventor) and s_4 (beading) is 0.55, establishing those two concepts as the correct meaning of artificer and beadwork in document d_1. Similarly, the words artificer and artist in d_2 are mapped to s_6 (artist) and s_2 (artisan) because they have highest Wu-Palmer similarity (0.85). Finally, s_5

Table 4 Comparison between clustering using nouns (N) versus concepts (C)

Dataset	Terms (T) after stopword removal		Nouns (N) after stopword removal & stemming		Concepts (C) after word-sense disambiguation		% Improvement from N to C
	# Terms	Purity	# N	Purity	# C	Purity	
B401	8,056	0.771	3,509	0.958	3790	0.959	0
B402	7,363	0.763	3,018	0.974	3327	0.964	−1
B403	10,229	0.750	4,305	0.907	4630	0.917	1
B404	11,207	0.975	4,557	0.979	4775	0.979	0
B405	8,304	0.926	3,629	0.960	3907	0.962	0
B406	10,468	0.961	4,295	0.957	4530	0.977	2
B407	9,130	0.650	4,159	0.650	4451	0.917	41
B408	10,803	0.970	4,717	0.962	5007	0.970	1
B409	9,375	0.647	4,244	0.635	4569	0.917	43
B410	1,087	0.560	4,715	0.887	5068	0.887	0
B411	10,283	0.965	4,563	0.952	4843	0.973	2
B412	11,205	0.645	4,997	0.655	5306	0.902	36
B413	9,435	0.640	4,340	0.657	4559	0.833	26
B414	7,728	0.870	3,545	0.907	3856	0.882	−3
B415	6,300	0.862	2,926	0.890	3197	0.867	−2
B416	7,147	0.935	3,135	0.940	3400	0.940	0
Multi10	13,615	0.440	5,177	0.538	5563	0.458	−9
Multi6	45,112	0.461	13,636	0.486	14589	0.490	.8
Reuters	14,261	0.603	5,922	0.650	6559	0.638	−1

(astragal) and s_6 (artist) are the appropriated concepts for d_3 because the pair has highest similarity (0.66). The situation is shown in the matrices below.

$$\begin{array}{c} \\ d_1: \\ d_2: \\ d_3: \end{array} \begin{array}{c} n_1 \; n_2 \; n_3 \\ 1 \; 1 \; 0 \\ 1 \; 0 \; 1 \\ 0 \; 1 \; 1 \end{array} \Longrightarrow \begin{array}{c} \\ d_1: \\ d_2: \\ d_3: \end{array} \begin{array}{c} s_1 \; s_2 \; s_3 \; s_4 \; s_5 \; s_6 \\ 1 \; 0 \; 0 \; 1 \; 0 \; 0 \\ 0 \; 1 \; 0 \; 0 \; 0 \; 1 \\ 0 \; 0 \; 0 \; 0 \; 1 \; 1 \end{array}$$

Notice that d_1 and d_2 share a common noun n_1 even though they are not from the same topic. Without an ontology, the cosine similarity between the documents is relatively high (.5). After disambiguation, the similarity decreases to 0. The same can be said about the document pair d_1 and d_3.

This example illustrates how the similarity between documents changes after resolving polysemy, substituting the "correct" sense for the noun. From a clustering point of view, documents d_1, d_2, d_3 might have been placed in the same cluster based on their noun features. However, after disambiguation, d_1 cannot be placed in the same cluster as d_2 and d_3 because the cosine similarity between d_1 and both documents is zero as we have shown. Of course, WSD does not always make a difference in similarity. For example, the cosine similarity between d_2 and d_3 did not change after disambiguation.

5.4 Synonymy

Synonymy occurs when more than one noun is used to express the same meaning. Synonymy makes clustering difficult as multiple nouns may represent a single semantic feature. In WordNet, synonymous nouns are combined into a single concept called a *synset*. Having synonymous nouns as part of the documents obscures the shared meaning, and subsequently the similarity of documents. That is because each synonymous noun is an individual component in the document's feature vector. However, if those synonymous nouns are replaced by their corresponding concepts, then the similarity between the documents should increase.

Example 2 Let $D = \{d_1, d_2\}$ be a pair of documents constructed from the following set of nouns $\{n_1 = net, n_2 = network, n_3 = mesh, n_4 = structure\}$. Suppose n_1, n_2, and n_3 are synonyms that correspond to the same sense (s_1) and n_4 has only 1 sense (s_2). The data matrix before and after performing word-sense disambiguation using information from a given ontology is shown below:

$$\begin{array}{c|cccc} & n_1 & n_2 & n_3 & n_4 \\ d_1: & 1 & 1 & 0 & 1 \\ d_2: & 0 & 0 & 1 & 0 \end{array} \Longrightarrow \begin{array}{c|cc} & s_1 & s_2 \\ d_1: & 1 & 1 \\ d_2: & 1 & 0 \end{array} \tag{5}$$

Without an ontology, the similarity between documents d_1 and d_2 is equal to 0 even though both documents are about the same topic. By mapping the nouns n_1 and n_2 in d_1 to their corresponding synset s_1 and the noun n_3 in d_2 to the same synset s_1, the similarity between documents d_1 and d_2 increases to 0.707, thus increasing the likelihood that both documents will be placed in the same cluster.

5.5 Importance of polysemous and synonymous nouns in clustering

Polysemy and synonymy are two fundamental document aspects that affect the similarity value computed between two documents in a text corpus. While resolving polysemy decreases similarity between unrelated documents, it also increases the dimensionality of the problem. On the other hand, resolving synonymy increases similarity but reduces the dimensionality of the feature space. We have shown in Sect. 5.2 that disambiguating the polysemous and synonymous nouns often yields comparable performance as using all the nouns, even though it is a more expensive operation. This begs the question: how prevalent and important are polysemous and synonymous nouns in document clustering?

In this section, we investigate the role polysemous and synonymous nouns play in document clustering. First, we observe that the polysemous and synonymous nouns form, on average, only 30% of all the nouns in the datasets we examined. Nevertheless, we show that the correlation of pair-wise document similarity using polysemous and synonymous nouns with pair-wise document similarity using all nouns is very high. Furthermore, we show that the purity of the clusters obtained using just the polysemous/synonymous nouns is higher than that obtained using randomly selected non-polysemous/non-synonymous nouns. This is an interesting finding that can be utilized to decrease the dimensionality of the feature space (to be discussed in Sect. 7).

We highlight this finding in more detail in the following discussion using two datasets as examples. In this experiment, we represent documents using five types of features:

X_{all}:	the set of all nouns
X_{poly}:	the set of all polysemous nouns
X_{syn}:	the set of all synonymous nouns
X_{both}:	the set of all polysemous and synonymous nouns
X_{rand}:	a randomly selected nouns that are neither polysemous nor synonymous

There are three sets of X_{rand} created in this study; each has the same number of features as X_{poly}, X_{syn} and X_{both} respectively. To evaluate the importance of polysemous and synonymous nouns, we examine the pair-wise document similarity matrices obtained using X_{all}, X_{poly}, X_{syn}, X_{both} and X_{rand}. Let S_{all} be the cosine similarity matrix for every pair of documents using X_{all} as the feature set. We also compute the cosine similarity matrix using each of the reduced feature sets described above and denote them as S_{poly}, S_{syn}, S_{both}, and S_{rand} respectively. If polysemous and synonymous nouns play an important role in document clustering, then the correlations between the similarity matrices (S_{all}, S_{poly}), (S_{all}, S_{syn}), and (S_{all}, S_{both}) should be significantly higher than that between (S_{all}, S_{rand}). A high correlation between S_{all} and S_{poly}, S_{syn}, or S_{both} indicates that polysemous and synonymous nouns are the dominant features in clustering. They are also likely to make a difference in clustering. A low correlation between (S_{all}, S_{rand}), on the other hand, suggests that using any random subset of the non-polysemous and non-synonymous nouns as features will significantly distort the cosine similarity between documents. In addition to the correlation between their cosine similarity matrices, we also compare the purity of clusters obtained after applying spherical k-means clustering to each of the feature set.

Table 5 shows a summary of our results for all the datasets. One important detail to note is that neither synonymous nor polysemous nouns are necessarily contained in *all* the documents and, as a result, some documents could not be clustered. We addressed this issue by adding a subset of random nouns (A) (that contains nonpolysemous or synonymous nouns) to the X_i feature set for complete document coverage. We also use the same subset (A) with the random nouns in order to maintain the same feature vector dimensionality. Table 5 shows that the correlation using polysemous and synonymous nouns is always high, i.e., above 85%, for most of the datasets. This indicates that those nouns retain much of the information needed to compute the cosine similarity between documents and thus strongly participate in forming the final clusters. In contrast, when using a random subset of the same dimensionality of non-polysemous and non-synonymous nouns as the documents features, the correlation was consistently lower, which means the features are not as dominant when computing similarity between documents.

Figure 2 shows the average correlations of dataset B414 for each X_i type using only 27% of the total nouns. We report the average correlation because the set of nouns (A) used to augment the features X_{poly} or X_{syn} are randomly selected, and thus, the experiment can be repeated several times using different sets of A. This percentage includes features from X_i or X_{rand} plus features from (A) to ensure all the approaches have the same number of features. Notice that the correlation between (S_{all}, S_{poly}) and (S_{all}, S_{both}) is significantly higher than the correlation between (S_{all}, S_{syn}) and (S_{all}, S_{rand}). A similar trend is observed in dataset B412 (see Fig. 3).

In terms of cluster quality, combining polysemous and synonymous nouns seems to produce higher cluster purity compared to using a random subset of nouns (see columns 6 and 7 in Table 5). This result further supports the previous analysis that shows the polysemous and synonymous nouns carry more weights in terms of finding the true clusters of a domain compared to non-polysemous and non-synonymous nouns. The cluster purity using polysemous and synonymous nouns is slightly worse than using all nouns for most of the datasets due to the loss of information. In Sect. 7, we present an approach to further reduce dimensionality by

Table 5 Comparison between the different feature sets used to represent documents in terms of the number of features, cluster purity, and correlation between their respective similarity matrices

Data	Number of features			Cluster purity			Correlation of	
	All nouns	Polysemous & synonymous nouns	% Nouns that are polysemous or synonymous	All nouns	Polysemous & synonymous nouns	Random nouns	Polysemous & synonymous nouns	Random nouns
B401	3,509	963	28	0.960	0.904	0.703	0.875	0.521
B402	3,018	873	29	0.977	0.964	0.879	0.896	0.606
B403	4,305	1,228	29	0.907	0.896	0.711	0.900	0.578
B404	4,557	1,330	29	0.980	0.957	0.880	0.914	0.672
B405	3,629	1,008	28	0.960	0.919	0.827	0.889	0.558
B406	4,295	1,314	31	0.958	0.905	0.830	0.897	0.646
B407	4,159	1,172	28	0.650	0.623	0.776	0.890	0.604
B408	4,717	1,378	29	0.963	0.803	0.737	0.878	0.577
B409	4,244	1,214	29	0.635	0.618	0.754	0.912	0.613
B410	4,715	1,393	30	0.888	0.705	0.645	0.891	0.562
B411	4,563	1,353	30	0.953	0.943	0.744	0.914	0.557
B412	4,997	1,402	28	0.655	0.710	0.700	0.911	0.642
B413	4,340	1,196	28	0.658	0.674	0.638	0.905	0.508
B414	3,545	974	27	0.908	0.897	0.606	0.855	0.510
B415	2,926	741	25	0.890	0.860	0.636	0.850	0.462
B416	3,135	879	28	0.940	0.811	0.686	0.877	0.490
Multi10	5,177	1,589	30	0.538	0.380	0.30	0.874	0.466
Multi6	13,636	5,582	40	0.486	0.422	0.357	0.940	0.445
Reuters	5,922	2,122	36	0.603	0.602	0.136	0.951	0.577

selecting a subset of salient features (known as core semantic features) after applying WSD on the polysemous/synonymous nouns and taking into consideration their potential impact on clustering. We show that the cluster purity obtained using the core semantic features are better than using the polysemous/synonymous nouns alone and are comparable to using all nouns (or better when the disambiguated senses are helpful to clustering).

6 Cluster-based information gain

To further reduce dimensionality, we use an information theoretic approach to select salient semantic features. In particular, we seek to identify nouns that have high information gain after being replaced by their corresponding semantic concepts. A high information gain means that the average entropy of the semantic concepts after disambiguation is significantly lower than the entropy of the original noun (before disambiguation). In other words, most of the documents that contain each of the disambiguated semantic concepts are from the same class (though the documents that contain the original noun belong to different classes). Unfortunately, computing information gain requires knowledge of ground truth, which is

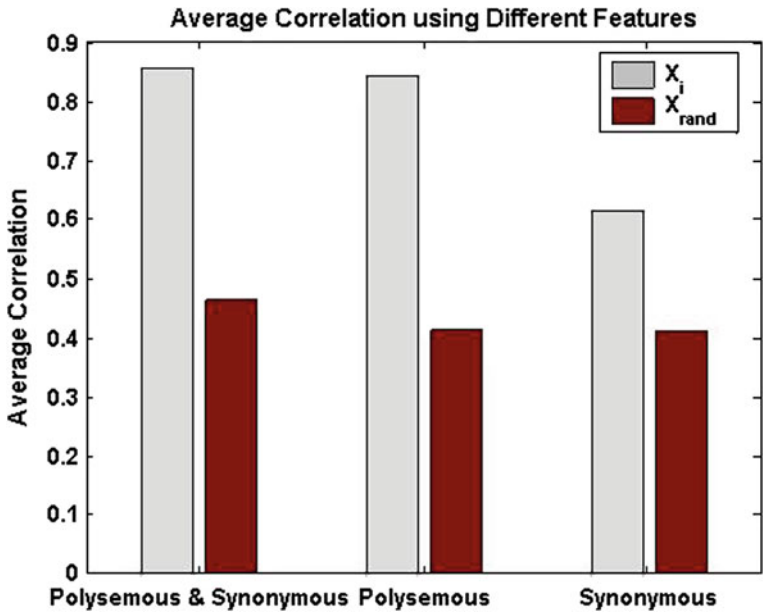

Fig. 2 Dataset B414 correlations using different X_i

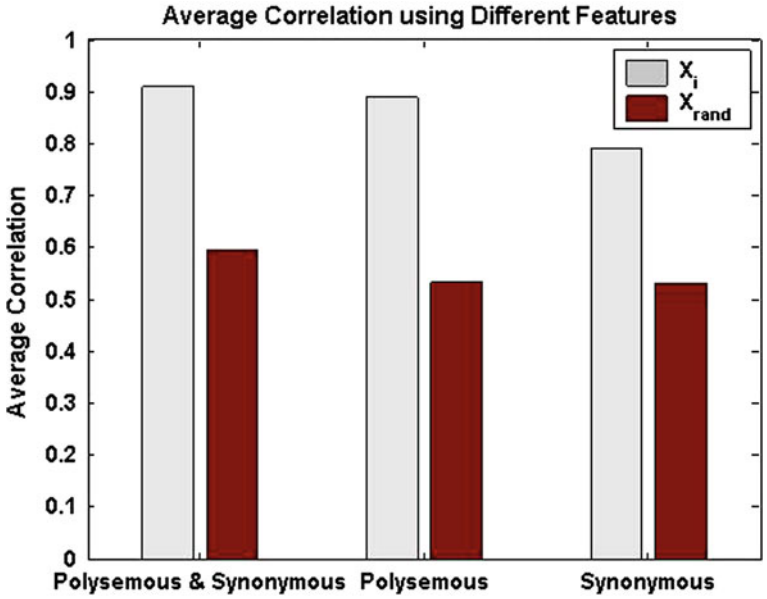

Fig. 3 Dataset B412 correlations using different X_i

unavailable due to the unsupervised nature of the clustering task. To compensate for this, we approximate the ground truth using clusters obtained from all nouns and all semantic concepts. Since spherical k-means is sensitive to the initial centroids, clustering is repeated a

Table 6 Document distribution across clusters before and after disambiguation

	π_{n1}	π_{n2}
Frequency before disambiguation		
n	3	3
	π_{s1}	π_{s1}
Frequency after disambiguation		
c1	2	1
c2	0	1
c3	1	1

number of iterations to obtain more stable clusters to simulate the ground truth. The following steps describe how the information gain of a noun n is computed:

1. Cluster the documents separately using the nouns as features and the semantic concepts as features. Let π_n be the set of noun clusters and π_s be the set of semantic clusters.
2. Compute the entropy of each noun n using the π_n clusters. Let $p(i|n)$ be the fraction of documents containing noun n that belong to cluster i. The entropy of a noun n is

$$e_{\text{noun}}(n) = -\sum_{i \in \pi_n} p(i|n) \log p(i|n)$$

3. Compute the average entropy of the concepts associated with this noun as retrieved by our WSD approach. Let $C(n) = \{c_1, c_2, \ldots, c_l\}$ be the set of concepts that disambiguate the noun n and let $p(j|n, c_i)$ be the fraction of documents containing noun n (before disambiguation) and concept c_i (after disambiguation) that were assigned to semantic cluster j. Furthermore, $p(c_i|n)$ corresponds to the fraction of documents containing noun n that were disambiguated to concept c_i. The average entropy of the set of concepts associated with noun n is:

$$e_{\text{concept}}(n) = -\sum_{c_i \in C(n)} p(c_i|n) \sum_{j \in \pi_s} p(j|n, c_i) \log p(j|n, c_i)$$

4. The information gain from disambiguating the noun n is computed as the difference in entropy before and after disambiguation:

$$\text{Gain}(n) = e_{\text{noun}}(n) - e_{\text{concept}}(n)$$

Example 3 Consider the example shown in Table 6, in which a noun n is replaced (using WSD) with concepts $\{c1, c2, c3\}$. π_{n1} and π_{n2} are the noun clusters, and π_{s1} and π_{s2} are the semantic clusters.

The entropy values before and after disambiguation can be computed as follows:

$$\begin{aligned}
e_{\text{noun}}(n) &= -\left(\frac{3}{6}\log\left(\frac{3}{6}\right)\right) - \left(\frac{3}{6}\log\left(\frac{3}{6}\right)\right) = 1 \\
e_{\text{concept}}(n) &= \frac{3}{6}\left(-\frac{2}{3}\log\left(\frac{2}{3}\right) - \frac{1}{3}\log\left(\frac{1}{3}\right)\right) + \frac{1}{6}\left(-1 \times \log\left(1\right)\right) \\
&\quad + \frac{2}{6}\left(-\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right)\right) = 0.7925 \\
\text{Gain}(n) &= e_{\text{noun}}(n) - e_{\text{concept}}(n) = 1 - 0.7925 = 0.2075
\end{aligned}$$

7 Core semantic features extraction

So far, the results in Sect. 5.5 have indicated that polysemous and synonymous nouns play an important role in clustering. Although using such nouns alone reduces the dimensionality considerably, they also degrade clustering performance due to loss of information. As we have noted, disambiguating all nouns may increase the dimensionality of the feature space without significantly improve cluster purity (except in 4 of the 19 datasets shown in Table 4). In this study, we seek an approach that reduces the number of features dramatically while achieving cluster purity that is comparable to using all nouns (or better when the disambiguated senses are beneficial to clustering).

Specifically, the proposed clustering framework aims to utilize ontological knowledge from WordNet to obtain a good clustering solution using only a small subset of the semantic features, i.e., the *core semantic features*. These features are not only useful for clustering, but once identified, they may represent the main theme of the topics in the documents. In our method, we globally evaluate the significance of the nouns to decide whether their disambiguated concepts should be added to the final subset of semantic features. We establish this significance using information gain as described above in Sect. 6.

Overall, we use two criteria to establish whether a noun is a core feature:

1. The noun should be a polysemous or synonymous noun as described in Sect. 5.5. We impose further restriction that it be in the top 30% of the most frequent nouns. This final restriction further reduced the dimensionality but did not empirically change the clustering results. This is likely due to the fact that, according to our experiments, polysemous and synonymous are approximately 60% of the most frequent nouns.
2. The noun should achieve either an information gain that is greater than a predefined threshold t or an entropy equals to 0 after disambiguation, i.e., $e_{\text{concept}}(n) = 0$. The first part of this requirement ensures that disambiguating an ambiguous noun makes a considerable change in the document distribution across the clusters. The second part of this requirement is related to the nouns for which the distribution of the documents across the clusters after disambiguation produce entropy 0; in this case each sense of the nouns brings similar documents together in one class.

When a noun passes both requirements, it is added to the core semantic feature sets. Once the core semantic features are identified, spherical k-means is used to cluster the documents to produce the base clusters. However, since the set of core features is only a small subset of the entire features set, only a subset of the documents will be clustered which we call "covered documents". That is, there will be a number of documents which become "uncovered", not contain any of the core semantic features. We address such uncovered documents by mapping them to the clusters that have the closest centroids.

Algorithm 7.1 Algorithm for Extracting Core Semantic Features

Input: *A set of documents, D, a set of nouns N, a set of concepts C, a set of noun clusters π_n, a set of semantic clusters π_c, the list of polysemous and synonymous nouns $M_{polysynm}$, the list of frequent nouns M_{freq}, and a threshold t.*

Output: *Document clusters D_c using as a feature set the core semantic features Fc.*

1. *Initialize the set of core semantic features $F_c = \emptyset$*
2. *for each $n \in N$*

(a) *Identify the concept set c associated with n using the above described method.*
(b) *Identify the information gain IG of n by clustering with and without the concept(s) c*
(c) *if all of the following conditions hold:*
   i. $n \in M_{poly\_synm}$
   ii. $n \in M_{freq}$
   iii. $IG \geq t$
   then add c to the list of core semantic features i.e., $F_c \leftarrow F_c \cup c$

3. *Identify the document subset* $\hat{D}$*, which are the documents covered by the newly identified core features* F_c
4. *Use spherical k-means to cluster* $\hat{D}$ *with feature set* F_c.
5. *Identify the document set* $\acute{D} = D - \hat{D}$*, the uncovered documents*
6. *Map the uncovered documents* $\acute{D}$ *to the best of the existing centroids from step 4*

8 Experimental evaluation

This section presents the empirical validation of our proposed algorithm for extracting core semantic features. The performance of our algorithm will be evaluated in terms of cluster purity, amount of feature reduction, and interpretability of the cluster centroids.

8.1 Datasets

We consider two benchmark document datasets in our experiment: Reuters-21578 and 20Newsgroups. The details of each dataset is given below.

8.1.1 Reuters-21578

Although Reuters-21578 [23] has been widely used for evaluating document clustering algorithms, this dataset has several known limitations. First, more than half of the documents are not annotated with class labels while some of them are assigned to multiple classes. Furthermore, the class distribution is not uniform. The size of some classes such as earn and acquisition is relatively large, while others such as reserve and veg-oil have few documents. To address these limitations, we sampled a dataset that contains the top 20 largest classes and which for the remainder of this paper we will term "Reuters". In this dataset, we discard the unlabeled documents and assign documents with multiple labels to the first class listed that is in the 20 classes. Furthermore, to maintain size among the classes, we sample a maximum of 200 documents from each class.

8.1.2 20Newsgroups

The 20Newsgroups dataset (http://people.csail.mit.edu/jrennie/20Newsgroups,) contains articles from 20 different newsgroups, each corresponding to a different class. The original data contains 18,846 documents. We created two main datasets based on the number of classes included: multiple-class datasets and pair/binary datasets.

Multiple-class datasets

Unlike Reuters-21578, 20Newsgroups documents have one class label. However, since some of the classes are highly correlated. As a result, the 20 classes are also aggregated into

6 classes, grouping those classes which are related. For example, the subclasses "rec.autos", "rec.motorcycles", "rec.sport.baseball", and "rec.sport.hockey" are aggregated into the class "recreation". Following the approach used by [33] and [9] and originally published by [21], we generated 2 multiple class datasets; one is the *Multi6* dataset in which we sampled 6,000 articles from the aggregated classes and each class is represented by 1,000 articles. The other dataset randomly draws from 10 classes from the original classes, where each class is represented by 500 documents. This dataset is called *Multi10*.

Binary-datasets This is a collection of 17 binary datasets (B401, B402, . . ., B416) that were randomly generated from a pool of 10 classes. Each binary dataset contains 400 documents, out of which 200 are randomly sampled from each class. The average number of nouns is approximately 4,000 for each dataset.

8.2 Clustering using core semantic features

Having described our approach, we here report on the performance of spherical k-means clustering using core semantic features. We chose spherical k-means as our underlying clustering algorithm for several reasons. First, it is a popular algorithm used by many ontology-driven clustering methods. Furthermore, since our goal is to evaluate the utility of the core semantic features, we avoid using algorithms that perform any additional feature transformations during clustering. k-means fits our criteria because it considers every feature in the input data, unlike other document clustering algorithms such as spectral clustering, which performs clustering on eigenvectors of a Laplacian matrix constructed from the feature vectors of its input data.

We compared the performance of the following four methods on the datasets described in the previous subsection:

1. Spherical k-means clustering using all nouns as features.
2. Spherical k-means clustering using a combination of all nouns and their corresponding concepts (including hypernyms of each concept up to five levels distant in the WordNet concept hierarchy). This approach is similar to the one given by [14], except we use binary features rather than the term frequency.
3. Spherical k-means clustering using the core semantic features (labeled CSF).

The clustering parameters used were the same for all four methods. The parameter k is set to the known number of classes for these data sets. Because spherical k-means is dependent on initial centroid location, clustering is repeated 50 times for each method. Empirically, we observed that the results do not change considerably even if the number of iterations increases to 100. The quality of the clustering results is evaluated using purity, which is a supervised measure that computes the fraction of documents correctly assigned to their ground truth clusters. In addition, the amount of reduction in the number of input features is used as another criterion for evaluating the usefulness of the core semantic features. Given a baseline method B, we compute the percentage of reduction in the number of input features as follows:

$$\%\text{Reduction} = \frac{\#\text{Features(B)} - \#\text{Features(CSF)}}{\#\text{Features(B)}}$$

where CSF denote the proposed core semantic features approach. The results of our experiments are summarized in Table 7.

The following observations are made when comparing the CSF approach against other baseline methods:

Table 7 Comparison between the core semantic feature (CSF) approach against three baseline methods (using all nouns, or Hotho's method) in terms of cluster purity and amount of feature reduction

Data	ALL nouns		Hothos' method		Core semantic features (CSF)				Feature reduction using CSF compared to using		
	# Features	Purity	# Features	Purity	# Features	# Docs covered	Covered doc purity	All doc purity	All nouns (%)	All concepts (%)	Hotho feat. (%)
B401	3,509	0.958	8,922	0.942	258	339	0.973	0.949	93	93	97
B402	3,018	0.974	7,827	0.939	307	357	0.977	0.957	90	91	96
B403	4,305	0.907	10,589	0.844	308	262	0.927	0.899	93	93	97
B404	4,557	0.979	10,976	0.959	406	333	0.997	0.985	91	91	96
B405	3,629	0.960	9,164	0.952	262	349	0.979	0.957	93	93	97
B406	4,295	0.957	10,484	0.950	334	352	0.980	0.967	92	93	97
B407	4,159	0.650	6,268	0.945	243	321	0.968	0.932	94	95	96
B408	4,717	0.962	11,458	0.910	292	315	0.939	0.935	94	94	97
B409	4,244	0.635	10,457	0.625	328	337	0.955	0.927	92	93	97
B410	4,715	0.887	11,417	0.787	225	260	0.826	0.817	95	96	98
B411	4,563	0.952	11,019	0.942	344	353	0.992	0.975	92	93	97
B412	4,997	0.655	12,079	0.625	390	335	0.907	0.887	92	93	97
B413	4,340	0.657	11,422	0.895	334	336	0.812	0.795	92	93	97
B414	3,545	0.907	8,959	0.855	165	227	0.748	0.787	95	96	98
B415	2,926	0.890	7,615	0.857	131	207	0.729	0.772	96	96	98
B416	3,135	0.940	8,055	0.900	218	310	0.948	0.930	93	94	97
Multi10	5,177	0.538	7,607	0.458	70	144	0.805	0.506*	98	98	99
Multi6	13,636	0.486	15,569	0.352	152	839	0.559	0.439*	98	98	99
Reuters	5,922	0.650	8,304	0.622	33	161	0.914	0.305*	99	99	99

see discussion on how we may improve the purity of the multi-class datasets with the *

1. We achieve a feature reduction of at least 90% using the CSF approach on all the datasets compared to using all nouns or all concepts (see columns 10–11 in Table 7). On average, the number of core semantic features used for clustering is close to 6% of the number of nouns used for spherical k-means clustering. Although our intention is to evaluate our CSF method in terms of reducing the dimensionality rather than improving clustering, we noticed that in 12 of 19 datasets, the cluster purity for our CSF approach (column 9 in Table 7) is at least comparable or better than the cluster purity of using all nouns (column 3 in Table 4) even though it has less than 91% of the nouns (as well as semantic concepts). For B407, B409, B412, and B413 (i.e., datasets where sense information helps), the cluster purity using CSF is higher. These results suggest that the core semantic features provide a good representative subset of the ontological concepts used for clustering. There are several possible explanations for the poor performance of the CSF approach in 7 of the remaining 19 datasets. First, 3 of the datasets correspond to the case where ontology does not help (B414, B415, and Reuters). Second, 2 of these datasets have a wide range of topics (Reuters, Multi6 and Multi10), this issue is discussed in more details later in this section. Furthermore, the core semantic features are extracted using the information gain when that gain exceeds the threshold t as described in our algorithm 7.1. Threshold t is an adjustable value that determines the level of information gain sufficient to change the clustering results when replacing a noun with its associated concept. The core semantic features can be used to both remove semantic noise introduced by the WSD process as well as exclude semantic features that have no effect on document distribution across the clusters (estimated classes). The results reported in Table 7 is based on a fixed threshold $t = 0.5$. For datasets such as B413, we achieve a higher purity comparable to using all concepts when the threshold is reduced to 0.3.
2. The ontology-driven clustering approach developed by Hotho has the highest number of features because it augments the original nouns with concepts from the WordNet hierarchy. On average, the number of features selected using our CSF approach is about 3% of the total number of features required by Hotho's method as shown in columns 12 in Table 7. In addition, the clusters purity obtained using core semantic features is better than that produced by Hotho's method. Note that we dropped the frequency information of the features in Hotho's method to maintain an equivalent comparison with the core features that are binary weighted.
3. Comparing column 9 in Table 7 to column 6 in Table 5, the cluster purity using CSF is higher than using polysemous/synonymous nouns for the majority of the datasets (with more than 10% improvements observed in 8 of the 19 datasets). The number of features obtained using CSF is also considerably smaller than the number of polysemous/synonymous nouns. This result suggests our CSF approach was able to selectively identify the polysemous/synonymous nouns to disambiguate without losing much information.
4. In spite of the benefits gained from focusing on the core features, this small portion of the total feature set might not cover all the topics in a dataset. That is, the core semantic features might not include *any* features from some documents, leaving them uncovered by the features being used. This requires mapping the uncovered documents to one of the existing core feature centroids based on "closeness" of those centroids. This is especially prevalent in the case of multi-class datasets where the topics are more varied. For example, in the Multi6 dataset the purity using all nouns is 0.486 but the purity decreases to 0.439 using only the core semantic features to cluster all the documents (uncovered as well). In the Reuters dataset, the purity of the all noun case, 0.65, decreased to 0.305 when using only the core semantic features to cluster all the documents. To better solve the "uncovered" documents problem, we modified the centroid mapping approach as

follows. Instead of mapping the *entire* set of the uncovered documents to their core feature centroids, a random subset of the uncovered documents were mapped to their closest centroids. After mapping, the centroids of the clusters were updated to reflect the new added documents. This was done iteratively until all the uncovered documents were mapped. This modification improved the final cluster purity for Multi6 and Reuters. For Multi6, the purity was raised from 0.439 to 0.495. For Reuters, the purity of their clusters increased to 0.433 from 0.305. In the Multi6 case, the cluster purity was comparable to those obtained using all nouns, as was the case for most of the other datasets. However, for the Reuters dataset, which has 20 topics, the purity remains below the purity of that for all nouns.

In short, our empirical results showed that the core semantic features not only produced clusters with comparable (or sometimes higher) purity as using all nouns, it also reduces the number of features significantly. Despite its smaller number, these core semantic features are informative and sufficiently capture the main themes of a text corpus. Thus, it can be used to efficiently cluster new documents using a reduced number of semantic concepts.

8.3 Effect of using core semantic features

Our strategy for selecting the core features using information gain can be used to find the main topics in a dataset. Generally, after clustering is performed, topics can be inferred from the centroids of the formed clusters. If cluster purity indicates that the clusters are sufficiently representative, then we assume that the centroids of the clusters cover the main topics of the documents and each cluster contains documents that share a similar topic. In this experiment, we show the advantages of using the core semantic features for clustering. Due to space limitations, the results are shown for dataset B410 only, although similar plots can be drawn for other datasets. First, we compare the distances of the documents to the respective cluster centroids when clustering using the core semantic features. Since there are only two clusters, the distances can be visualized in a 2-d plot as shown in Fig. 4. Each data point is marked as either a circle or an asterisk, depending on its ground truth class. Observe that all the points marked as circles (denoting class 1) have a significantly larger distance to the centroid of cluster 2 than the centroid of cluster 1. On the other hand, if we use all the concepts as features, the distance from a document to the centroid of its opposite class is no longer pushed to its maximum value (see Fig. 5).

As previously noted, the WSD process could potentially introduce erroneous semantic features into the text corpus. Since the core semantic features are frequent and have high information gain, we expect such features to be more accurate when used to guide the clustering process. To examine the quality of the core semantic features, we compare the cluster centroids obtained using k-means clustering on all nouns against the centroids obtained from core semantic features. Table 8 shows the list of top 15 features that form the cluster centroids for the B410 dataset.

The dataset contains documents belonging to the "science.space" and "science.electronics" categories. Using all nouns as features (columns 1 and 2 in Table 8), observe that the top features of the cluster centroids contain many noisy terms (e.g., make and years for centroid 1 and Article, Time, and Email for centroid 2). On the other hand, using the core semantic concepts as features (columns 3 and 4 in Table 8), most of the top features that form the centroids clearly identify the topic of the class. For example, the second cluster centroid contains concepts, such as, "orbit", "mission", "satellite", "atmosphere", "exploration", "gravity", and "astronomy", which all related to the "science.space" class, whereas the centroid for the first cluster includes "circuit", "ampere", "transformer", "chip", and "resistor" as its top-ranked

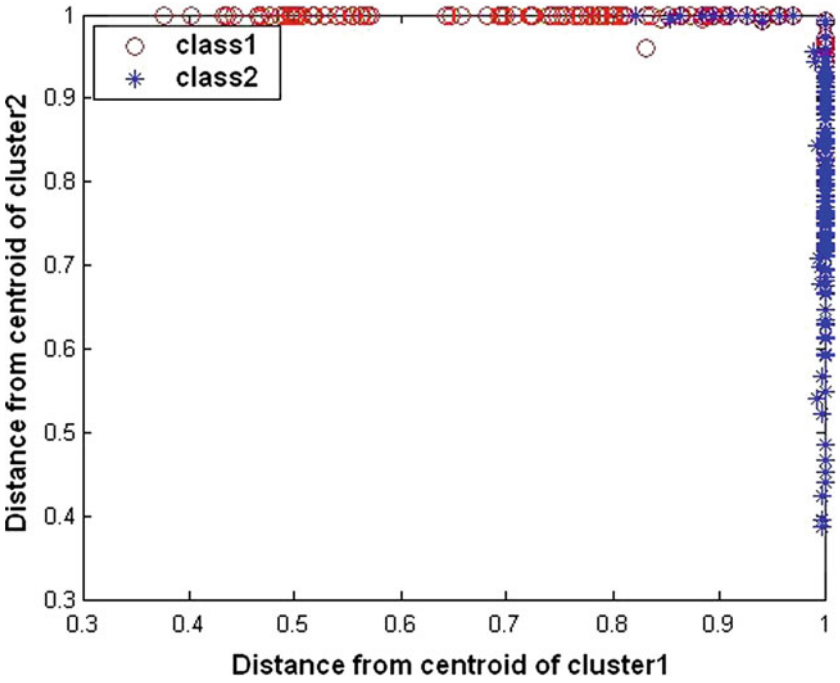

Fig. 4 Distances of documents from centroids of clusters using the core concepts

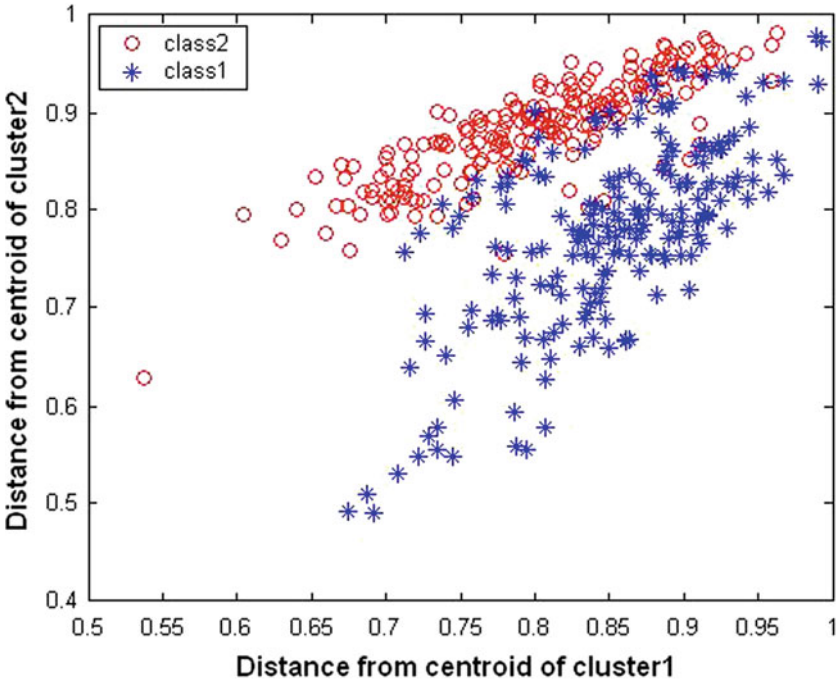

Fig. 5 Distances of documents from centroids of clusters using all the concepts

Table 8 Top 15 features of the centroids in the clusters using nouns and the clusters using the core concepts

Top 15 nouns		Top 15 core concepts	
Noun_centroid1	Noun_centroid2	Concept_centroid1	Concept_centroid2
Space	Article	Circuit	Orbit
Article	Work	Ampere	Mission
NASA	Time	Transformer	Satellite
System	Power	Chip	Military
Orbit	Make	Police	Budget
Time	Email	Resistor	Landing
Make	Question	Advice	Mary
Years	Good	Chips	Satellite
Earth	Circuit	Wiring	Gravity
Program	Line	Connection	Astronomy
Work	Problem	Amplifier	Exploration
Science	Id	Player	Astronaut
Shuttle	Back	Memory	Comet
Research	Current	Port	Mars
Launch	University	Exit	Dynamics

features, which is consistent with the "science.electronics" class. A question might be asked about the concept "Mary", which appears in the top 10 concepts in the "science.space". The reason is because "Mary" exists as an individual concept in WordNet and "Mary Shafer", a NASA employee, was a frequent contributor to the newsgroup "science.space".

In short, the results of this section suggest that the core semantic features produce cluster centroids that are informative and relate to the main topics of the documents.

9 Conclusion

This paper presents a methodology for clustering using core semantic features. Our analysis shows that clustering using terms identified by WordNet as nouns often produce results that are comparable to those using WSD. Furthermore, the polysemous and synonymous nouns play an important role in clustering, even though their disambiguation does not necessarily lead to significant improvement in cluster purity. We also showed that it is possible to select a subset of the semantic features that are useful for clustering. To do this, we introduced an "unsupervised" information gain measure to determine whether a "disambiguated" noun should be used as a feature in clustering. Our experimental results showed that the core semantic features were sufficient to not only substantially reduce the dimensionality of the feature set, but also maintain or possibly improve clustering using all nouns (especially when the sense information is helpful).

References

1. Andrzejewski D, Zhu X, Craven M (2009) Incorporating domain knowledge into topic modeling via dirichlet forest priors. ICML 25–32

2. Al Sumait L, Domeniconi C (2007) Local semantic kernels for text document clustering. In: SIAM international conference on data mining workshop on text mining
3. Banerjee S, Ramanathan K, Gupta A (2007) Clustering short texts using Wikipedia. SIGIR 787–788
4. Basu S, Banerjee A, Mooney RJ (2002) Semi-supervised clustering by seeding. ICML 19–26
5. Bodner RC, Song F(1996) Knowledge-based approaches to query expansion in information retrieval. Adv Artif Intell 146–158
6. CLUTO Family of Clustering Software Tools: http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview
7. Craven M, DiPasquo D, Freitag D, McCallum A, Mitchell TM et al (2000) Learning to construct knowledge bases from the World Wide Web. Artif Intell 118:69–113
8. Chemudugunta C, Smyth P, Steyvers M (2008) Combining concept hierarchies and statistical topic models. CIKM 1469–1470
9. Dhillon I, Mallela S, Modha D (2003) Information-theoretic co-clustering. KDD 89–98
10. Farahat AK, Kamel MS (2010) Enhancing document clustering using hybrid models for semantic similarity. In: SIAM international conference on data mining workshop on text mining
11. Fodeh SJ, Punch W, Tan PN (2009) Combining statistics and semantics via ensemble model for document clustering. SAC 1446–1450
12. Gabrilovich E, Markovitch S (2006)Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge. NCAI 21:1301–1306
13. Gabrilovich E, Markovitch S (2007) Computing semantic relatedness using Wikipedia-based explicit semantic analysis. IJCAI 1606–1611
14. Hotho A, Staab S, Stumme G (2003) WordNet improves text document clustering. In: SIGIR 2003 semantic web workshop. 541–544
15. Hu J, Fang L, Cao Y (2008) Enhancing text clustering by leveraging Wikipedia semantics. SIGIR 179–186
16. Hu X, Sun N, Zhang C, Chua TS (2009) Exploiting internal and external semantics for the clustering of short texts using world knowledge. CIKM 919–928
17. Ifrim G, Theobald M, Weikum G (2005) Learning word-to-concept mappings for automated text classification. In: Workshop on learning in web search (LWS 2005). 18–25
18. Jiang JJ, Conrath DW (1997) Semantic similarity based on corpus statistics and lexical taxonomy. International Conference Research on Computational Linguistics (ROCLING X)
19. Jing L, Zhou L, Ng MK, Huang JZ (2006) Ontology-based distance measure for text clustering. In: SIAM SDM workshop on text mining
20. Kandylas V, Upham SP, Ungar LH (2009) Finding cohesive clusters for analyzing knowledge communities. Knowl Inf Syst 17:335–354
21. Lang K (1995) NewsWeeder: learning to filter netnews. ICML 331–339
22. Larsen B, Aone C (1999) Fast and effective text mining using linear-time document clustering. KDD 16–22
23. Lewis D (1997) Reuters-21578 text categorization test collection. AT&T Labs Research
24. Lin D (1998) An information-theoretic definition of similarity. ICML 1:296–304
25. Mandala R, Tokunaga T, Tanaka H (1999) Complementing WordNet with Roget's and Corpus-based Thesauri for information retrieval. In: The 9th conference of the European chapter of the association for computational linguistics. 94–101
26. MeSH, National Library of Medicine Controlled Vocabulary: http://www.nlm.nih.gov/mesh
27. Moravec P, Kolovrat M, Snasel V (2004) LSI vs. WordNet ontology in dimension reduction and information retrieval. DATESO 288–294
28. Natural Language Toolkit: http://www.nltk.org
29. Recupero D (2007) A new unsupervised method for Document Clustering by using WordNet Lexical and Conceptual Relations. SIGIR 10:563–579
30. Rosso P, Ferretti E, Jimenez D et al (2004) Text categorization and information retrieval using WordNet senses. In: 2nd Global WordNet international conference. 299–304
31. Sedding J, Kazakov D (2004) WordNet-based text document clustering. In: 3rd workshop on Robust methods in analysis of natural language processing data. 104–113
32. Siolas G, d'Alche Buc F (2004) Support vector machines based on a semantic kernel for text categorization. IJCNN'00 5:205–209
33. Slonim N, Tishby N (2000) Document clustering using word clusters via the information bottleneck method. SIGIR 208–215
34. Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. In: Proceedings of KDD workshop on text mining 34:35–36
35. Tan PN, Steinbach M, Kumar V (2005) Introduction to data mining. Addison-Wesley Longman Publishing Co, Boston

36. Termier A, Rousset MC, Sebag M (2001) Combining statistics and semantics for word and document clustering. IJCAI 1:49–54
37. The 20 Newsgroups data set: http://people.csail.mit.edu/jrennie/20Newsgroups/
38. Vorhees E (1993) Using WordNet to disambiguate word senses for text retrieval. SIGIR 171–180
39. Wang P, Hu J, Zeng HJ et al (2007) Improving text classification by using encyclopedia knowledge. ICDM 332–341
40. Wang P, Domeniconi C (2008) Building semantic kernels for text classification using Wikipedia. KDD 713–721
41. Wang Y, Hodges J (2006) Document clustering with semantic analysis. HICSS 3:54c–54c
42. Wikipedia: http://wikipedia.edu
43. WordNet: http://wordnet.princeton.edu
44. Wu Z, Palmer M Verb (1994) Semantics and lexical selection. MACL 133–138
45. Xiong H, Steinbach M, Ruslim A et al (2009) Characterizing pattern preserving clustering. Knowl Inf Syst 19:133–138
46. Yoo I, Hu X, Song I (2006) Integration of semantic-based bipartite graph representation and mutual refinement strategy for biomedical literature clustering. KDD 791–796
47. Zhong S, Ghosh J (2005) Generative model-based document clustering: a comparative study. Knowl Inf Syst 8:374–384

Author Biographies

Samah Fodeh received her MS and Ph.D. degrees in Computer Science from the Michigan State University. She is an associate research scientist in Yale Center for Medical Informatics at Yale University. Her research interests are data mining, information extraction, and bioinformatics.

Bill Punch received his B.S. in Biochemistry in 1979 and his Ph.D. in Computer Science in 1989 from the Ohio State University. He is an associate professor in the Department of Computer Science and Engineering at Michigan State University, as well as the director of MSU's High Performance Computer Center. His interests are primarily the theory and application of evolutionary computation, but also does work in data mining and computer science education. He is the first author of the textbook "The Practice of Computing Using Python", published by Pearson.

Pang-Ning Tan received the Ph.D. degree in computer science from the University of Minnesota. He is an assistant professor in the Department of Computer Science and Engineering at Michigan State University. His research interests include data mining, Web intelligence, and machine learning. He is the first author of the textbook Introduction to Data Mining, published by Addison Wesley. He is a member of the ACM and the IEEE.