

A new unsupervised method for document clustering by using WordNet lexical and conceptual relations

Diego Reforgiato Recupero

Received: 16 December 2006 / Accepted: 19 September 2007 / Published online: 16 October 2007
© Springer Science+Business Media, LLC 2007

Abstract Text document clustering provides an effective and intuitive navigation mechanism to organize a large amount of retrieval results by grouping documents in a small number of meaningful classes. Many well-known methods of text clustering make use of a long list of words as vector space which is often unsatisfactory for a couple of reasons: first, it keeps the dimensionality of the data very high, and second, it ignores important relationships between terms like synonyms or antonyms. Our unsupervised method solves both problems by using ANNIE and WordNet lexical categories and WordNet ontology in order to create a well structured document vector space whose low dimensionality allows common clustering algorithms to perform well. For the clustering step we have chosen the bisecting k -means and the Multipole tree, a modified version of the Antipole tree data structure for, respectively, their accuracy and speed.

Keywords Clustering · Text documents · Bisecting k -means · Multipole · Antipole · WordNet

1 Introduction

With the growth of world wide web and online services, more and more information is available and accessible online. Due to the large diversity of the web and organizational intranets, the need of finding the truly relevant data has become a challenging task for applications such as information retrieval, business intelligence or enterprise portals. When querying a search engine, only a small number of relevant web pages is returned together with a large number of irrelevant pages. This web search low precision happens because there are many different contexts in which the keywords typed by the user occur. Thus, since only the first few results are consulted by the user because the limited amount of time, a lot of relevant pages will never be discovered. Text clustering methods can be used to structure all the results of a search engine query. So far, existing text clustering methods

D. Reforgiato Recupero (✉)
Dipartimento di Matematica e Informatica, Università degli Studi di Catania, Catania, Italy
e-mail: diegoref@dmf.unict.it

only relate documents that use identical terminology ignoring conceptual similarity of terms such as defined in resources like ANNIE or WordNet. ANNIE is an information extraction component of GATE (Cunningham et al. 2002) which we use for its accurate entity, pronoun and nominal co-references extraction. Three intuitive goals of WordNet (Miller 1995) project are the creation of dictionary and thesaurus, the support for automatic text analysis and artificial intelligence and the determining of semantic connections between sets of synonyms, for tracing morphological connections between words. Taking into account ANNIE and WordNet features will improve the quality of the resulting clustering.

One of the most critical problems for document clustering is the high dimensionality of the natural language text; this problem is known as *curse of dimensionality* (Friedman 1994). Different techniques to solve the curse of dimensionality have appeared in literature (Zervas and Ruger 1999, Fodor 2002, Parson et al. 2004).

In this paper, we present a new unsupervised method for text document clustering which reduces the size of vector space by using WordNet words lexical information and increases the clustering accuracy by using ANNIE's features. Section 2 introduces the concept of vector space model. In Sect. 3 we show which operations are performed to the documents before the document vectors are created. In Sect. 4 we discuss two different strategies that make use of WordNet to create the document vectors. The resulting document vectors will be the input data for the clustering algorithms described in Sect. 5. We offered two clustering schemas:

- the bisecting k -means (Steinbach et al. 2000), in which we have to choose the number of clusters k to create;
- the Multipole tree, a slightly different version of Antipole (Cantone et al. 2005), in which we choose a “tightness” measure (an integer value) where the higher the measure the smaller the cluster radius and hence the larger the number of generated clusters.

Multipole clustering is much faster than bisecting k -means. An experimental evaluation on real text data has been conducted and Sect. 6 reports and explains its major results. In Sect. 7 we point to some related work whereas Sect. 8 ends the paper with some concluding remarks.

2 Basic definitions

In this section we define the basic notations used in the rest of the paper. Given a set \mathcal{D} of documents we want to assign each document $d \in \mathcal{D}$ to clusters of similar documents discovering their natural categories. By relying on the vector space model, we represent each document $d \in \mathcal{D}$ by a vector of frequencies of the features it contains:

$$\vec{d} = (f_1, \dots, f_n).$$

Usually features are document terms; in our method features are classes of words. We express the set of documents \mathcal{D} as a $m \times n$ matrix, where m is the number of documents in \mathcal{D} and n is the number of features. The entry (i, j) contains the number of times the feature j occurs within the document i .

To measure the similarity between two documents d_1 and d_2 we use one of the most common distances, the cosine of the angle between the two vectors, which tries to

Table 1 The stop-words list used in the second preprocessing step

Subject pronouns	<i>I, you, he, ...</i>
Object pronouns	<i>Me, you, him, ...</i>
Possessive pronouns	<i>Mine, yours, his, ...</i>
Demonstrative pronouns	<i>This, that, ...</i>
Possessive adjectives	<i>My, your, his, ...</i>
Prepositions	<i>In, at, on, with, ...</i>
Auxiliaries	<i>Am, are, can, have, ...</i>
Articles	<i>The, a, ...</i>
Conjunctions	<i>And, but, while, because, ...</i>

approach the semantic closeness of documents through the size of the angle between term weight vectors associated to them:

$$\cosine(\vec{d}_1, \vec{d}_2) = \frac{(\vec{d}_1 \bullet \vec{d}_2)}{\|\vec{d}_1\| \cdot \|\vec{d}_2\|}$$

where \bullet denotes the vector dot product and $\|\cdot\|$ is the length of a vector. A cosine measure of 0 means the two documents are unrelated whereas a cosine measure close to 1 means that the documents are closely related.

Given a cluster C of \mathcal{D} , the centroid c of C is defined as the element with the smallest sum of distances to all the other documents of C :

$$c = \arg \min_{d_j \in C} \sum_{i=0, \dots, |C|} \cosine(\vec{d}_i, \vec{d}_j).$$

3 Text document preprocessing

In order to increase clustering performance, our method, as well as all the other methods of document clustering, requires different operations of data preprocessing.

1. First of all, we break each document down in sentences and use the link parser (<http://www.link.cs.cmu.edu/link/>) to extract only entities within noun and verb phrases; we remove articles, HTML tags (if in presence of HTML documents) and special characters and all entities recognized by ANNIE as dates.
2. Second, we have created a stop-words list containing non descriptive words that, if considered, would make the clustering imprecise (like pronouns, auxiliaries, prepositions, etc.).¹ Our stop-words list contains about 250 words. Table 1 shows all the categories of elements contained in such a list.
3. A third issue has concerned the words consisting of several terms. Those have been reduced and replaced to their basic stem by using the morphological capabilities of WordNet; we applied the Porter stemmer algorithm (see Porter 1980) to the words that do not appear as lexical entries in WordNet. We also used the nominal coreference of ANNIE in order to understand when two different words or compound words are

¹ The pronouns recognized by ANNIE as valid are replaced with the entity they refer to; such words have been removed from \mathcal{D} .

referred to the same entity; we consider coreferenced words as one entity of the map we will generate.

4. A further matter has been raised by rare words and how they affect the results. Our assumptions is that rare words do not help for identifying the appropriate cluster but, rather, they might add noise to the distance measure degrading the entire performance. Thus, our strategy has been to discard a word if it occurs less than a fixed threshold of the total number of words in \mathcal{D} . The threshold is a percentual value varying in the set $\{0.5, 1, 1.5, 2\}$.

4 Using WordNet for the features matrix

The vector space of many text clustering algorithms is typically very high since they consider the different lexical words as different entries in the document vector. In this way, each vector document has 1,000s of elements, many of which are 0. Methods for normalizing these vectors (usually to be unit length) are therefore needed; the big drawback of these methods is the loss of information and, consequently, vectors which are very dissimilar may appear to be close resulting then in a bad clustering.

WordNet (Miller 1995) is an online lexical reference system which organizes nouns, verbs, adjectives and adverbs into synonym sets (*synsets*), each representing one underlying lexical concept, and specifies a number of relationships such as hypernym, synonym, meronym which can exist between the synsets in the lexicon. WordNet has already been applied for reducing the dimensionality of such vectors (see the related work section for references). It comprises a core ontology and a lexicon. We have used WordNet 2.1 which consists of 109377 synsets and 144684 lexical entries.

Our method exploits different WordNet relations in order to create the features which form the dimensionality of our vector space. Two different strategies for finding features have been explored: the first one uses the WordNet lexical categories whereas the second one uses the WordNet ontology. They produce the input for the clustering algorithms. They both also succeed in keeping low the vector space dimensionality as well as forming well structured features vectors.

4.1 The WordNet lexical categories (WLC) technique

By using WordNet lexical categories we have mapped each document word remained after the preprocessing to lexical categories. WordNet 2.1 has 41 lexical categories for nouns and verbs. Tables 2 and 3 list such categories. Thus, for each document $d \in \mathcal{D}$ we associate a vector of fixed length 41 having in the entry i the number of words which belong to category i . For example, the word “dog” and “cat” both belong to the same category (“noun.animal”) and therefore they increment the counter of the same vector entry.

Since many words may have different senses, they have usually also multiple categories they refer to; a word sense disambiguation technique is then required in order to not add noise to the representation decreasing consequently the clustering quality. For example, the word *Washington* has 3 categories (*noun.location*, *noun.group*, *noun.person*) since it can be the name of the American president, a place (the state or the city) or a group if intended as the concept of capital.

Table 2 WordNet nouns lexical categories

Tops	Act	Animal	Artifact
Attribute	Body	Cognition	Communication
Event	Feeling	Food	Group
Location	Motive	Object	Person
Phenomenon	Plant	Possession	Process
Quantity	Relation	Shape	State
Substance	Time		

Table 3 WordNet verbs lexical categories

Body	Change	Cognition	Communication
Competition	Consumption	Contact	Creation
Emotion	Motion	Perception	Possession
Social	Stative	Weather	

...saying Tony Blair was right about that. That was the speech given by Mr Blair.
 After that he said : "I agree with Prime Minister Tony Blair's statement".
 Although the Prime Minister has announced ...

Fig. 1 Part of news document about Prime Minister Tony Blair

Different words expressing the same entity may also be present; we try to guess them by looking at coreferenced words discovered by ANNIE. For example, for the document shown in Fig. 1, ANNIE would create the following coreferenced list: *Tony Blair, Mr Blair, Prime Mister Tony Blair, Prime Minister*. In such a case we would keep only one of these entities which would count each reference to all the others reducing the size of the map we will be generating.

4.1.1 Disambiguation strategies

Wordnet assigns terms to concepts in an ambiguous way. Adding or replacing terms by concepts may add noise to the representation and produce a loss of information. Word sense disambiguation has been extensively studied in literature and it is not the purpose of this paper to determine which method is the best. Our study has been limited to understand if the use of a word sense disambiguation technique helps to create a better representation of terms in order to get a better clustering. We have thus used two different techniques for word disambiguation. The first is called *disambiguation by context*: this technique returns the concept which maximizes a function depending on the conceptual vicinity. Given a concept c , its semantic vicinity is defined as the set of all its direct sub and super concepts. This technique has been successfully applied in Hotho et al. (2003) where it is also more extensively discussed.

The second technique tries to resolve the correct sense of a polysemic word using a *concept map* as its context. A *concept map* is a list of concepts connected each other with linking phrases forming meaningful proposition between concepts. The algorithm is fully described in Canas et al. (2003). Let w be the word for which we want to find a sense and c a given concept map. The algorithm first selects the key concepts from c . Then it checks

whether any of these words are not present in WordNet, by making the morphological transformations. Synsets of the words within the concepts are clustered using the hypernym distance based on WordNets hypernym relation: only one synset per word is allowed in each cluster. At the end, several clusters will be generated, each with a different weight depending on the number of words in the cluster and the hypernym distance. The cluster with the highest weight that contains a synset s of w will be the selected cluster and s will be chosen as the sense of w .

The *disambiguation by context* is much faster than using *concept maps*. One study we conducted was to see if the clustering precision was better by using any disambiguation method not depending on some in particular. Thus, all the experiments, described in Sect. 6, have been in three different ways: without any disambiguation strategy, with the first strategy and with the second strategy. It turned out that, for each experiment (10 runs for experiment), the use of a disambiguation strategy gave better vectors representations: in particular, the disambiguation by using *concept maps* gave better representations than the *disambiguation by context*.

4.2 The WordNet ontology (WO) technique

Our second strategy for creating features vectors exploits WordNet ontology. An ontology defines a set of representational terms, that include concepts and relations. WordNet ontology is organized via the hypernym/hyponym relation, (superior/inferior concepts, basically given two words, there is an hyponymy relation between the concepts they refer to in a particular context of usage). For each word WordNet returns a set of lists, one for each synonymous, ranked according to the frequency of the usage in English of the synonymous it refers to. Each list is hierarchical: the root is the synonymous and the descendants are grouped according to the superior/inferior relation. For example, the third hypernyms set returned by WordNet for the word “president” is

*<head of state → representative → negotiator → communicator → person → being
→ living thing → object → entity >*

whereas, among the hyponyms, one of the output sets contains only the term,

<ex – president > .

The symbol \rightarrow in the above hypernyms list specifies the superior relation from left to right (*head of state* is superior than *politician* which is superior than *leader*, and so on).

Our bottom-up algorithm for creating WordNet ontology features is shown in Fig. 2. It takes as input the output (lexical categories vectors) of WLC technique and computes and returns the new set of categories W .

First, three sets are initialized: the set L is equal to the most frequent r lexical categories from WLC (line 1), the set T contains all the document words which belong to categories in L (line 2) and W , the output set, is initialized to the empty set (line 3). A strategy for choosing reasonable values of r is to select, in decreasing order for number of words, the categories until $y\%$ of the total number of words is retrieved. We have seen experimentally that $y = 25$ is a good trade-off between keeping low the space dimensionality and having a well structured vector space. Then, lines 4–15 deal with the creation of set W . A word $w \in T$ is assigned to an element-set $S \in W$ if each word $w_S \in S$ is in relationship with w according to the hypernyms or hyponyms relation. If that is not satisfied, a new element-set

Algorithm WO**Input:** $Output_{WLC}, \mathcal{D}$ **Output:** W

```

1.  $L \leftarrow \{\text{the most frequent } r \text{ lexical categories from } Output_{WLC}\}$ 
2.  $T \leftarrow \{w \in d : d \in \mathcal{D} \text{ and } w \text{ is assigned to } l, l \in L\}$ 
3.  $W \leftarrow \{\emptyset\}$ 
4. for each word  $w \in T$  do
5.    $found \leftarrow \text{false}$ 
6.   for each  $S \in W$  do
7.     if  $w \in \text{hypernyms}(s)$  or  $w \in \text{hyponyms}(s) \forall s \in S$  then
8.        $S \leftarrow S \cup w$ 
9.        $found \leftarrow \text{true}$ 
10.    end if
11.  end for each
12.  if  $found = \text{false}$  then
13.     $W \leftarrow W \cup \{w\}$ 
14.  end if
15. end for each
16. Merge together all the sets  $S_i \in W$  with more than  $x\%$  duplicates
17.  $R \leftarrow \{S_i \in W : S_i \text{ contains duplicates}\}$ 
18. for each duplicate word  $w$ 
19.   let  $S_j \in R : dist_h(w, S_j)$  is minimum
20.   keep  $w$  in  $S_j$  and remove  $w$  from all the others  $S_k \in R$ 
21. end for each
22. return  $W$ 

```

Fig. 2 The WO Algorithm

containing w is added to W . Every element-set of W forms a different class of words. We refer to the classes so created as the features of our vector space model.

Sometimes a word can be assigned to multiple sets. We perform two operations for removing such duplicates. First, all the sets having more than a fixed number x of duplicates in common (line 16) are merged. Experimentally, a value for x of 50% is a reasonable choice. Let S_m, \dots, S_p be the element-sets of S that still have duplicates and $W' = \{w'_1, \dots, w'_j\}$ be these duplicates. Given $S_i \in S$ and a word $w \in S_i$, let w_1, w_2 be two words of the same output list of hypernyms or hyponyms of w and let $f_h(w_1, w_2)$ be a function which measures how far the nodes w_1 and w_2 are in terms of number of hops. We define the distance $dist_h$ between a duplicate $w'_o \in W'$ and an element-set $S_i \in S$ as the sum of every distance f_h of w'_o and each non duplicate words in S_i divided by the size of S_i :

$$dist_h(w'_o, S_i) = \frac{\sum_{w_j \in S_i; w_j \text{ non-dup.}} f_h(w'_o, w_j)}{|S_i|}$$

The word w'_o is kept in the most correlated set (the one with the minimum $dist_h$) and it is discarded from all the others (lines 17–21).

As example, let us suppose that at the end of the process the set W contains two sets of words, $S_1 = \{\text{head of state, representative, negotiator}\}$ and $S_2 = \{\text{representative, communicator}\}$ with the consequent set of duplicates $W' = \{\text{representative}\}$. We will have $dist_h(\text{representative}, S_1) = 2/3$ and $dist_h(\text{representative}, S_2) = 1$ since the word *representative* comes right after *head of state* (1 hop) and right before *negotiator* (1 hop) in one of the hypernyms list returned for *president* whereas there is one node (two hops) between *representative* and *communicator*. By leaving the word *representative* in S_1 and removing it from S_2 we get a better structured vector space.

At the end of the procedure the set W contains all the new features for all the documents $d \in \mathcal{D}$. For each document d , the entry i of the new features vector contains the number of words in d that belong to the class i .

With this procedure, the number of obtained features will usually be much higher than the initial r . By choosing reasonable values of r this strategy is still able to keep the vector space low enough and experiments show that it improves a lot the resulting clustering.

Note that ANNIE is also able to recognize proper nouns, organizations, dates and locations and pronominal coreferencing improving the quality of the *WO* Algorithm.

5 Text documents clustering

Once the features vectors have been created, the clustering is performed by the bisecting k -means (Steinbach et al. 2000) or the Multipole, a slightly different version of Antipole clustering method (Cantone et al. 2005). The cosine of the angle between two vectors is the distance we considered to compute their similarities. We refer to our algorithms as $WLC_{Bisecting}$, $WLC_{Multipole}$, $WO_{Bisecting}$ and $WO_{Multipole}$ according to which strategy for finding features we are adopting (WLC for WordNet lexical categories or WO for WordNet ontology) and which clustering algorithm is being used (bisecting k -means or Multipole).

5.1 Bisecting k -means

We have used the bisecting k -means algorithm introduced in Steinbach et al. (2000). However, since the second step of this method uses the basic k -means algorithm, we will give some details of the basic k -means used. In our implementation of k -means, at the first iteration (that is for $t = 1$), the initial k centroids $q_1^1, q_2^1, \dots, q_k^1$ are computed by using the Gonzalez (1985) algorithm to find mutually distant centroids; then, the remaining objects are assigned to a class according to the relation $x_i \in C_j^1$ iff $d(x_i, q_j^1) \leq d(x_i, q_i^1)$, $1 \leq j, i \leq k$, $i \neq j$. After each iteration t , new centroids are computed in such a way that the performance index, $\gamma_i = \sum_{x \in C_i^t} |x - q_i^t|^2$, $i = 1, 2, 3, \dots, k$, is minimized. This achieves the condition $q_i^{t+1} = \frac{1}{n_i^t} \sum_{x \in C_i^t} x$. If $q_i^{t+1} = q_i^t$, the process finishes, otherwise, the objects are grouped again. This implementation of k -means clustering takes time $\mathcal{O}(t \cdot k \cdot |\mathcal{D}|)$, with k the number of clusters and t is the number of iterations. Normally, $k, t < < |\mathcal{D}|$.

5.2 Multipole

The Antipole clustering (Cantone et al. 2005) algorithm of bounded radius is performed by a top-down procedure starting from a given finite set of points S which checks if a given splitting condition is satisfied. This condition asks for two points whose distance is greater than the radius. If there are no two such points, then splitting is not performed and the given subset is a cluster on which an approximate centroid is then found. Otherwise, a suitable pair of points (A, B) of S called Antipole is generated and the set is partitioned by assigning each point of the splitting subset to the closest endpoint of the Antipole (A, B). As seen in Cantone et al. (2005) the randomized algorithms used by Antipole clustering makes its construction much faster than k -means's. The Antipole clustering has a worst-case complexity of $\frac{\tau(\tau-1)}{2} \cdot |\mathcal{D}| + o(|\mathcal{D}|)$ in the input size $|\mathcal{D}|$, where τ is the bounded radius (see Cantone et al. 2005 for further details).

In this paper we implemented the Multipole Tree, a version of the Antipole Tree with a different splitting condition. Instead of two we check for k nodes whose distance is greater than the radius. If they are found we generate a vector of points (A_1, \dots, A_k) called Multipole and the set is partitioned by assigning each point of the splitting subset to the closest endpoint of the Multipole (A_1, \dots, A_k) . Comparisons between Multipole and Antipole have been presented in Reforgiato (2007) where it is also shown that the former outperforms the latter in both clustering and querying time.

6 Experimental evaluation

The experiments show the evaluation of our method on some real text documents in comparison with four state of the art text clustering algorithms, FTC and HFTC (Beil et al. 2002), the bisecting k -means (Steinbach et al. 2000) and the k -secting k -means (Larsen and Aone 1999). We have implemented all algorithms in C# under Visual Studio 2005 using the .NET package (Crowe) for accessing WordNet 2.1. The ANNIE and link parser (<http://www.link.cs.cmu.edu/link>) preprocessing interface have been written in JAVA and ANSI C. For FTC and HFTC we have used a public domain implementation of the Apriori algorithm (Borgelt). The experiments have been run on a Pentium III PC with 730 MHz clock speed with 2 GB of RAM running Microsoft Windows XP Version 2002. Section 1 reports the used data sets whereas Sects. 2 and 3 report, respectively, the measures employed to evaluate the clustering quality and the main experimental results with running times. Since we are interested in running times we decided to use the disambiguation by context technique.

6.1 Data sets

To test any document clustering approach different measures exist in literature; a manually predefined categorization of the data sets helps to understand the resulting clustering. We have used three data sources from which we have generated four different data sets. Each data set reflects the conditions in a wide range of real life applications. They are:

- Reuters²: this corpus contains 21578 documents and 135 topics created manually. We divided topics in two classes according to their meaning. For example, topics like *cocoa*, *coconut* and *coffe* are all contained in the same class since they are related fruits. Each document in the corpus has been assigned to one or more topics based on its content. We selected only documents associated with only one topic and topics with less than 10 associated documents have been discarded. From the resulting documents we have created two data sets, one with 1504 articles grouped by 13 topics and the other with 1657 articles grouped by 25 topics.
- *Classic*: this corpus consists of 1,400 CRANFIELD documents from aeronautical system papers, 1,460 CISI documents from information retrieval papers and 1,033 MEDLINE documents from medical journals. All these documents have been divided in 3 classes. The Classic corpus is freely downloadable.³

² <http://www.daviddlewis.com/resources/testcollections/reuters21578/>

³ <ftp://ftp.cs.cornell.edu/pub/smart>

Table 4 Data Sets

Data sets _{#classes}	# Articles	# Words	# Entities
Reuters ₁₃	1,504	938,591	10,487
Reuters ₂₅	1,657	1,075,393	12,919
Classic ₃	3,891	2,763,466	44,391
ManGen ₃₀	2,421	1,304,919	16,875

- *ManGen*: to create this corpus we have been retrieving about 20 documents per day for 4 months (starting from January 2006) from a set of news web site.⁴ We have created a list of 30 classes where each class is associated to a set of topic words and synonyms. We have assigned each document to the most related class according to its topic. There are numerous algorithms for topic detection (Allan 2002), hence, we do not address this problem in this paper.

Table 4 gives a summary description of the used data sets. The column # *words* indicates the total number of words after the preprocessing; column # *entities* gives the total number of entities (proper nouns, organizations, dates, locations) found by using ANNIE in the correspondent dataset.

6.2 Clustering evaluation

To evaluate the quality of non-hierarchical clustering we have used the entropy (Boley 1998, Barbara et al. 2002, Dhillon 2001) whereas for hierarchical clustering we have used the *F*-measure (Beil et al. 2002, Larsen and Aone 1999, Nickerson et al. 2001, Van Rijsbergen 1979). We have also used the Silhouette method (Bolshakova and Azuaje 2003) as an intrinsic measure of non-hierarchical clustering quality.

Let D be a documents data set, $C = \{C_1, \dots, C_m\}$ a clustering and $K = \{K_1, \dots, K_n\}$ a correct classification.

- *Entropy*: The entropy of C is a measure of the disorder within all the clusters and it is defined as:

$$E(C) = \sum_{C_j} \frac{n_j}{|D|} \sum_i -p_{ij} \ln(p_{ij}) \quad (1)$$

where n_j is the number of elements of the cluster C_j and p_{ij} is the probability that a member of C_j belongs to the class K_i . It returns values in the interval $[0, \ln(|K|)]$; the lower the entropy is and the purer is the produced clustering.

- *F-measure*: To evaluate the clustering quality for hierarchical clustering, the *F*-measure is typically used. It is defined as

$$F(C) = \sum_{K_i \in K} \frac{|K_i|}{|D|} \max_{C_j \in C} \{F(K_i, C_j)\}$$

where the *F*-measure of cluster C_j and class K_i is defined as:

⁴ <http://www.cnn.com>, <http://www.nytimes.com>, <http://www.usatoday.com>

$$F(K_i, C_j) = \frac{2 \cdot R(K_i, C_j) \cdot P(K_i, C_j)}{R(K_i, C_j) + P(K_i, C_j)}$$

having precision $P(K_i, C_j) = \frac{n_{ij}}{|C_j|}$, recall $R(K_i, C_j) = \frac{n_{ij}}{|K_i|}$ and n_{ij} is the size of class K_i . The F -measure score is in the range $[0,1]$; the higher the value is and the better is the resulting clustering. Since this measure is typically used for hierarchical clustering, we have used it to analyze all the hierarchical clustering algorithms we have compared.

- *Silhouette*: Let C_j with $j = 1, \dots, c$ be a cluster. The silhouette technique assigns to the i_{th} sample of C_j a quality measure, $s(i)$ with $i = 1, \dots, m$, known as “silhouette width”. This value is a confidence indicator on the membership of the i_{th} sample in cluster C_j and it is defined as :

$$s(i) = (b(i) - a(i)) / \max(a(i), b(i)),$$

where $a(i)$ is the average distance between the i_{th} sample and all the samples included in C_j ; $b(i)$ is the minimum average distance between the i_{th} sample and all of the samples clustered in C_k with $k = 1, \dots, c$ and $k \neq j$. $s(i)$ May assume a value in $[-1,1]$: 1 when the i_{th} sample has been assigned to an appropriate cluster and -1 when it has been assigned to a misclassified cluster. Let S_j be the sum of all samples’ silhouette widths in C_j : it characterizes the heterogeneity of the cluster C_j . For each clustering, the global silhouette value is defined as:

$$GS_u = \frac{1}{c} \sum_{j=1}^c S_j.$$

7 Results

The data sets above have been first cleaned by using the preprocessing seen in Sect. 3 (the threshold discussed about the rare words has been set to 1.5).

Since the entropy measure favors larger number of clusters, we have always measured the produced clusterings of each algorithm with the same number of clusters. Figures 3 and 4 show the entropy and the global silhouette values for $WLC_{Bisecting}$, $WO_{Bisecting}$, FTC, bisecting k -means and 9-secting k -means on all the data sets. For the basic bisecting k -means and 9-secting k -means, the adopted vector space has unit length (details about such a vector space can be found in Steinbach et al. 2000).

In all experiments $WLC_{Bisecting}$ and $WO_{Bisecting}$ outperform all the other competitors. WO technique is usually the best for preparing the vector space for clustering but it is slower than WLC. Figure 5 depicts the runtime of $WLC_{bisecting}$, $WO_{bisecting}$, $WLC_{Multipole}$ and $WO_{Multipole}$ with respect to the number of documents on the regular Reuters corpus.

For a fair comparison of $WLC_{Multipole}$ and $WO_{Multipole}$, we have compared them with HFTC, and the hierarchical versions of bisecting k -means and 9-secting k -means. Moreover, we have compared them with $WLC_{Bisecting}$ and $WO_{Bisecting}$ by using the hierarchical version of bisecting k -means. As far as the Multipole clustering is concerned, details on how to choose an optimal radius can be found in Cantone et al. (2005) and in Reforgiato (2007). Since it is not possible to know a-priori the number of clusters that all the hierarchical clustering methods generate, for each algorithm we obtain a single clustering on each data set. Table 5 reports the F -measure for such algorithms. As shown in the results,

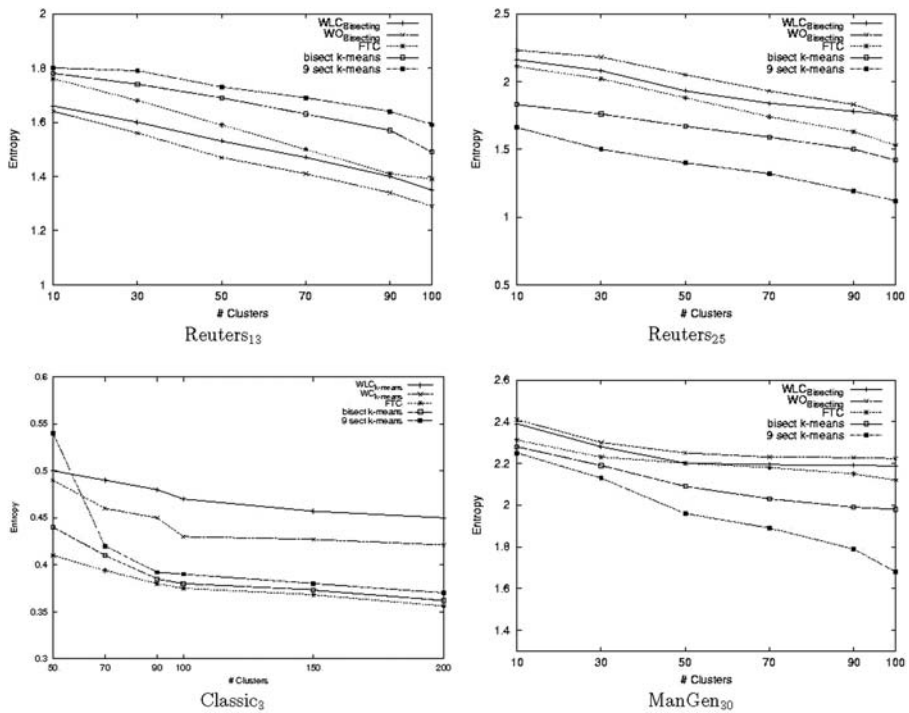


Fig. 3 Entropy comparison for all the non-hierarchical clustering algorithms on the four data sets averaged over 10 runs

Multipole is much faster than the bisecting k -means and its resulting clustering is not too much worse.

To improve clustering quality, a refinement may be used. Clustering obtained by both bisecting k -means and all hierarchical clustering algorithms can be further improved by using the basic k -means algorithm. If the centroids of the clusters produced by these two techniques are used as initial centroids for the basic k -means, then this will change the initial centroids and readjust the clusters. For all experiments we have performed, this refinement usually improves the clustering quality. Table 6 reports the F -measure on the used data-sets for $WLC_{Bisecting}$, $WO_{Bisecting}$, $WLC_{Multipole}$, $WO_{Multipole}$ adopting such a refinement for the hierarchical version of bisecting k -means and Multipole.

Monte Carlo cross-validation technique (Smyth 1996) has been applied when using the bisecting k -means clustering. First, each dataset has been randomly divided M times into disjoint train and test partitions where the test partition is a fraction β of the overall data. For each partition, k is varied from 1 to k_{max} and the clustering is performed on the training data. Each model with k components is applied to the unseen data in the test partition and the Silhouette value is computed for each. This process is repeated M times and the M cross-validated estimates are averaged for each k . Figure 6 shows the global silhouette values for $WLC_{Bisecting}$ and $WO_{Bisecting}$ on all the data sets.

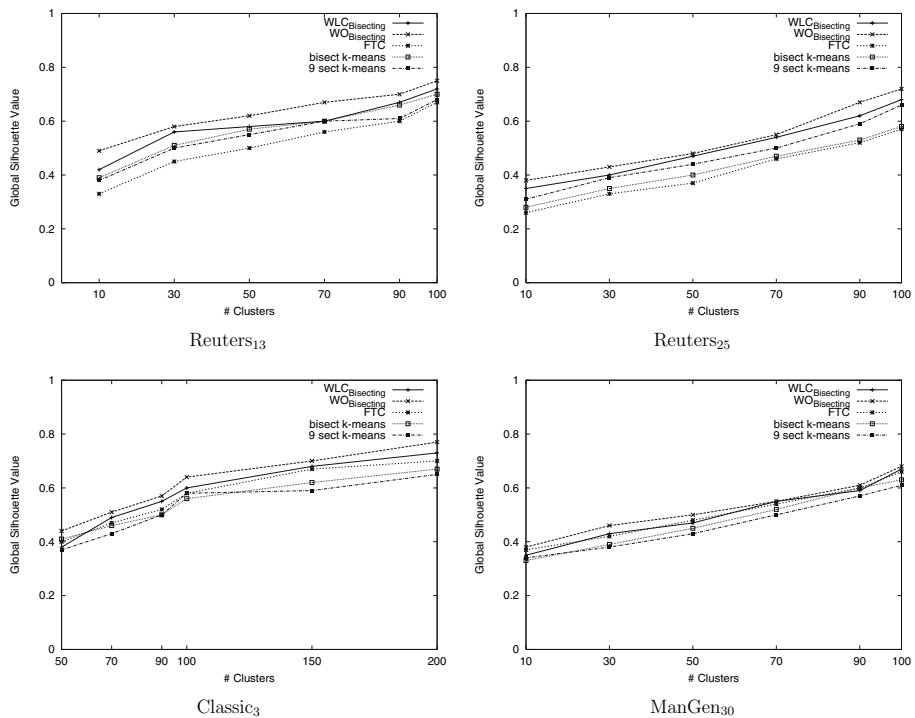


Fig. 4 Global Silhouette values for all the non-hierarchical clustering algorithms on the four data sets averaged over 10 runs

Table 5 *F*-measure comparison for all the hierarchical clustering algorithms on the four data sets over 10 runs

Data set	WLC _{Bis}	WO _{Bis}	WLC _{Mul}	WO _{Mul}	HFTC	Bis.	9-s
Reuters ₁₃	0.59	0.60	0.54	0.55	0.56	0.49	0.48
Reuters ₂₅	0.58	0.61	0.52	0.51	0.57	0.50	0.46
Classic ₃	0.57	0.58	0.54	0.52	0.56	0.54	0.51
ManGen ₃₀	0.59	0.63	0.57	0.59	0.55	0.49	0.46

Table 6 *F*-measure comparison for WLC and WO with hierarchical version of bisecting *k*-means and Multipole adopting the refinement

Data set	WLC _{Bisecting}	WO _{Bisecting}	WLC _{Multipole}	WO _{Multipole}
Reuters ₁₃	0.61	0.62	0.57	0.58
Reuters ₂₅	0.60	0.59	0.55	0.55
Classic ₃	0.59	0.60	0.56	0.60
ManGen ₃₀	0.62	0.64	0.58	0.57

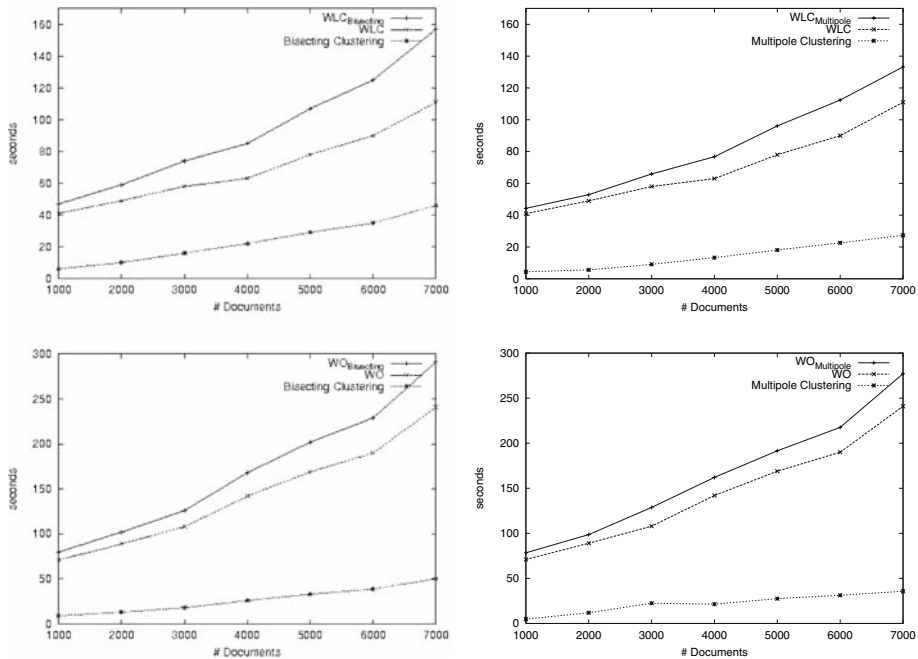


Fig. 5 Scalability of WLC_{Bisecting}, WO_{Bisecting}, WLC_{Multipole} and WO_{Multipole} on the Reuters corpus

8 Related work

Recently, clustering has been proposed as a tool for browsing large document collections and organizing the results returned by a search engine in response to a user's query (Zamir et al. 1997). Many different methods have been proposed in literature. They include the SuffixTree Clustering (Zamir and Etzioni 1998), the bisecting k -means (Steinbach et al. 2000), k -secting k -means (Larsen and Aone 1999), Scatter/Gather (Cutting et al. 1992), FTC and HFTC (Beil et al. 2002). Even if the bisecting k -means outperforms other well known hierarchical clustering algorithms (Steinbach et al. 2000), an approach which uses frequent term sets like FTC and HFTC, allows to reduce drastically the dimensionality of the data and to deal with very large data sets. In this context, WordNet is useful to reduce the dimensionality of the data. It has been already been used by Green (1997, 1999) to construct lexical chains from the occurrence of terms in a document: WordNet senses that are in some way related receive higher weights than senses that appear in isolation from others in the same document. The senses with the best weights are selected and the corresponding weighted term frequencies constitute a base vector representation of a document. Another application of WordNet, (Dave and Lawrence 2003), uses synsets as features for document representation and subsequent clustering. In this work the word sense disambiguation has not been performed showing that WordNet synsets decreases clustering performance in all the experiments. Other works (Moldovan and Mihalcea 2000, Voorhees 1994) have explored the possibility to use WordNet for retrieving documents by carefully choosing a search keyword. A successful integration of WordNet resource for a document categorization task is shown in de Buenaga Rodriguez et al. (2000), Urena Lopez et al. (2001) where WordNet is applied to a supervised scenario building the vectors

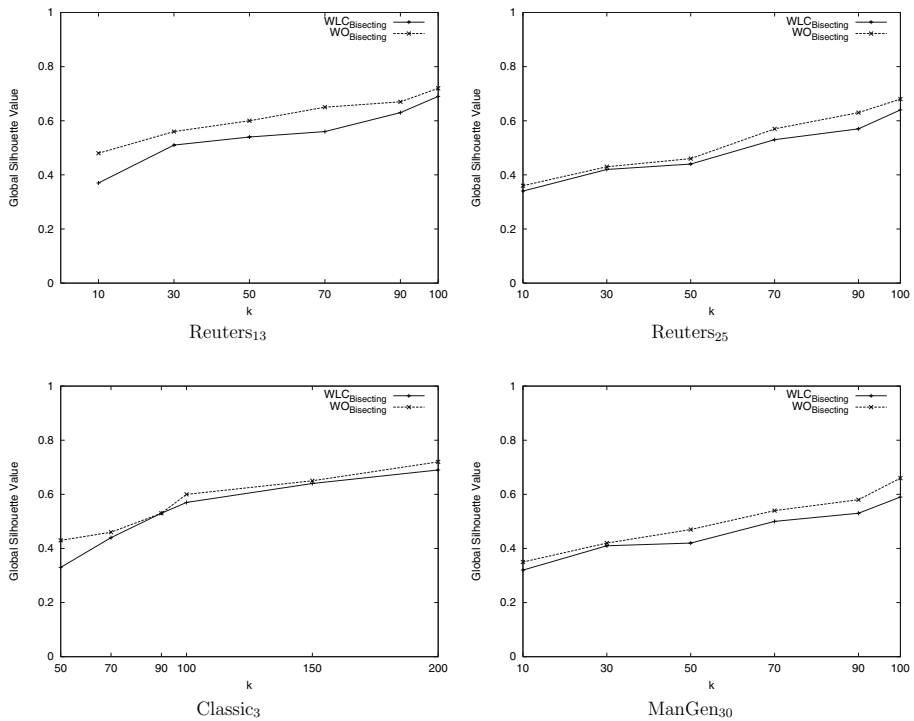


Fig. 6 Monte Carlo cross validation with $M = 100$, $\beta = 0.5$

manually without considering WordNet relations. Conversely, (Hotho et al. 2003) uses WordNet in an unsupervised scenario taking into account the WordNet ontology and lexicon and some strategy for word sense disambiguation achieving improvements of the clustering results.

Semantic features selection has been shown promising in Chua and Kulathuramaiyer (2004), Jing et al. (2006) where WordNet based semantic approach and ontologies are able to provide a better set of features for category representation.

A technique for feature selection by using WordNet to discover synonymous terms based on cross-referencing is introduced in Chua and Kulathuramaiyer (2004). First, terms with overlapping word senses co-occurring in a category are selected. A signature for a sense is a synset containing synonyms. Then, the list of noun synsets is checked for all senses for signatures similarity. The semantic context of a category is aggregated by overlapping synsets of different terms senses. The original terms from the category that belongs to the similar synsets will be finally added as features for category representation.

In Jing et al. (2006) the authors describe how to find mutual information between terms (TMI) by using the background knowledge through the ontology of WordNet and introduce a distance measure (TMID) used in the clustering phase.

In Sedding and Kazakov (2004) the authors explore the benefits of partial disambiguation of words by their PoS and the inclusion of WordNet concepts; they show how taking into account synonyms and hypernyms, disambiguated only by PoS tags, is not successful in improving clustering effectiveness because the noise produced by all the incorrect senses extracted from WordNet. Adding all synonyms and all hypernyms into the

document vectors seems to increase the noise. A possible solution to that would be to use a word-by-word disambiguation in order to choose the correct sense of a word: consequently, only the hypernyms for the correct sense would be considered.

9 Conclusions and future work

In this paper we have presented WLC and WO, two techniques that employ WordNet for the creation of a low and well-structured vector space for document clustering. Two recent effective clustering schemes, the bisecting k -means and the Multipole tree, have been then applied to the generated documents vectors. ANNIE's entity, pronoun and nominal coreference extraction has been used in the preprocessing. To the best of our knowledge there is no work about document clustering which takes into account ANNIE capabilities. Comparisons with many other document clustering algorithms have been carried out indicating the good performances of WLC and WO. The main considerations we have made are:

- Clustering obtained with WO is usually the best but it is computationally more expensive than WLC.
- The clustering quality of all algorithms, in general, decreases if we do not apply the preprocessing to the documents in \mathcal{D} especially the word sense disambiguation technique.
- The refinement adopted for readjusting clusters usually improves the clustering quality.
- Our future work will investigate the two following aspects:
- A first one is to further improve the preprocessing since it is the most critical step for the generation of an appropriate document representation.
- The other one is how the use of WordNet benefits the cluster labeling task: having documents represented by concepts rather than isolated words it is possible to find common concepts and give a label to each resulting cluster.

Acknowledgments The author would like to strongly thank anonymous reviewers for the time and effort they spent in evaluating this manuscript.

References

- Allan, J. (2002). Introduction to topic detection and tracking. In *Topic detection and tracking: Event-based information organization* (pp. 1–16). Kluwer Academic Publishers.
- ANNIE. Annie—a robust cross-domain ie system. <http://www.gate.ac.uk/ie/annie.html>
- Barbara, D., Li, Y., & Couto, J. (2002). Coolcat: An entropy-based algorithm for categorical clustering. In *Proceedings of the 11th international conference on Information and knowledge management* (pp. 582–589).
- Beil, F., Ester, M., & Xu, X. (2002). Frequent term-based text clustering. *KDD 02*. pp. 436–442.
- Boley, D. (1998) Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4), 325–344.
- Bolshakova, N., & Azuaje, F. (2003). Improving expression data mining through cluster validation. *Information Technology Applications in Biomedicine*, 19–22.
- Borgelt, C. (2000) Apriori—association rule induction/frequent item set mining. <http://www.fuzzy.cs.uni-magdeburg.de/borgelt/apriori.html>
- Canas, A. J., Valerio, A., Lalinde-Pulido, J., Carvalho, M., & Arguedas, M. (2003). Using wordnet for word sense disambiguation to support concept map construction. *SPIRE*, 2857, 350–359.
- Cantone, D., Ferro, A., Pulvirenti, A., Reforgiato, D., & Shasha, D. (2005). Antipole tree indexing to support range search and k-nearest-neighbor search in metric spaces. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 17(4), 535–550.

- Chua, S., & Kulathuramaiyer, N. (2004). Semantic feature selection using wordnet. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 166–172).
- Crowe, M. (2000). Wordnet.net library. <http://www.opensvn.csie.org/WordNetDotNet/>
- Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). Gate: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, July 2002.
- Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey, J. W. (1992). Scatter/gather: A cluster-based approach to browsing large document collection. In *Proc. ACM SIGIR 92* (pp. 318–329).
- Dave, D. M. P. K., & Lawrence, S. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *WWW 03 ACM* (pp. 519–528).
- de Buenaga Rodriguez, M., Gomez Hidalgo, J. M., & Diaz Agudo, B. (2000). Using wordnet to complement training information in text categorization. In N. Nicolov & R. Mitkov (Eds.), *Recent advances in natural language processing II: Selected papers from RANLP'97, current issues in linguistic theory (CILT)* (pp. 353–364). Amsterdam/Philadelphia: John Benjamins.
- Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proc. of the 11th international conference on knowledge discovery and data mining* (pp. 269–274).
- Fodor, I. K. (2002). A survey of dimension reduction techniques. *LLNL technical report, UCRL ID-148494* URL: <http://www.llnl.gov/CASC/sapphire/pubs.html>
- Friedman, J. H. (1994). An overview of predictive learning and function approximation. In V. Cherkassky, J. H. Friedman, & H. Wechsler (Eds.), *From statistic to neural networks, Proc. NATO/ASI Workshop* (pp. 1–61).
- Gonzalez, T. F. (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38, 293–306.
- Green, S. J. (1997). Building hypertext links in newspaper articles using semantic similarity. *NLDB 97* (pp. 178–190).
- Green, S. J. (1999). Building hypertext links by computing semantic similarity. *TKDE*, 11(5), 50–57.
- Hotho, A., Staab, S., & Stumme, G. (2003). Wordnet improves text document clustering. *ACM SIGIR Workshop on Semantic Web*.
- Jing, L., Zhou, L., Ng, M. K., & Huang, J. Z. (2006). Ontology-based distance measure for text clustering. *SIAM conference on data mining*.
- Larsen, B., & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *Proc. of the 5th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 16–22).
- Urena Lopez, L. A., Gomez de Buenaga Rodriguez, M., & Gomez Hidalgo, J. M. (2001). Integrating linguistic resources in tc through wsd. *Computers and the Humanities*, 35(2), 215–230.
- Miller, G. (1995). Wordnet: A lexical database for English. *CACM*, 38(11), 39–41.
- Moldovan, D. I., & Mihalcea, R. (2000). Using wordnet and lexical operators to improve internet searches. *IEEE Internet Computing*, 4(1), 34–43.
- Nickerson, A., Japkowicz, N., & Milios, E. (2001). Using unsupervised learning to guide re-sampling in imbalanced data sets. In *Proc. of the 8th international workshop on AI and statistics* (pp. 261–265).
- Parson, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explorations Newsletter*, 6(1), 90–105.
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130–137.
- Reforgiato, D. (2007). Hierarchical clustering data structure comparisons. *Technical Report*.
- Van Rijsbergen, C. J. (1979). Information retrieval, 2nd ed. Dept. of Computer Science, University of Glasgow.
- Sedding, J., & Kazakov, D. (2004). Wordnet-based text document clustering. *3rd Workshop on Robust Methods in Analysis of Natural Language Data*, 104–113.
- Smyth, P. (1996). Clustering using monte carlo cross-validation. *Knowledge Discovery and Data Mining*, 126–133.
- Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. In *Proc. TextMining Workshop, KDD 2000*.
- Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *Proc. of ACM-SIGIR* (pp. 61–69).
- Zamir, O., & Etzioni, O. (1998). Web document clustering: A feasibility demonstration. In *Proc. ACM SIGIR 98* (pp. 46–54).
- Zamir, O., Etzioni, O., Madani, O., & Karp R. M. (1997). Fast and intuitive clustering of web documents. *KDD 97*, 287–290.
- Zervas, G., & Ruger, S. M. (1999). The curse of dimensionality and document clustering. In *Proc. of the IEE Searching for Information: AI and IR Approaches* (pp. 19/1–19/3).