J Intell Inf Syst (2012) 39:577–610
DOI 10.1007/s10844-012-0204-9

# DoSO: a document self-organizer

**Gerasimos Spanakis · Georgios Siolas · Andreas Stafylopatis**

Received: 2 February 2012 / Revised: 25 April 2012 / Accepted: 26 April 2012 /
Published online: 12 May 2012


**Abstract** In this paper, we propose a Document Self Organizer (DoSO), an extension of the classic Self Organizing Map (SOM) model, in order to deal more efficiently with a document clustering task. Starting from a document representation model, based on important "concepts" exploiting Wikipedia knowledge, that we have previously developed in order to overcome some of the shortcomings of the Bag-of-Words (BOW) model, we demonstrate how SOM's performance can be boosted by using the most important concepts of the document collection to explicitly initialize the neurons. We also show how a hierarchical approach can be utilized in the SOM model and how this can lead to a more comprehensive final clustering result with hierarchical descriptive labels attached to neurons and clusters. Experiments show that the proposed model (DoSO) yields promising results both in terms of extrinsic and SOM evaluation measures.

**Keywords** Document representation · Document clustering · SOM · Wikipedia

G. Spanakis (✉) · G. Siolas · A. Stafylopatis
National Technical University of Athens, Athens, Greece
e-mail: gspana@ece.ntua.gr

G. Siolas
e-mail: gsiolas@islab.ntua.gr

A. Stafylopatis
e-mail: andreas@cs.ntua.gr

*Present Address:*
G. Spanakis
Intelligent Systems Lab, Athens, Greece

## 1 Introduction

The rapid proliferation of text documents raises the need to create systems able to explore and utilize the available information. Automatic organization of documents into topic categories is an important step in indexing, retrieval, management and mining of text data effectively and efficiently in terms of space, time and quality. Document clustering is a (generally) unsupervised learning task that generates groups of similar documents with each group ideally representing a specific topic (Willett 1988). This process combines aspects of Machine Learning (ML), Information Retrieval (IR) and Natural Language Processing (NLP).

The Self-organizing map (SOM) (Kohonen et al. 2001) is one of the most widely used neural models and has some interesting features over other neural networks. It is an unsupervised learning method which produces a topology preserving mapping between a high dimensional input space and a low dimensional map space in low computational time. Groups of nodes with short distances to each other form clusters, visually identified on the map. Due to the preservation of topologic properties the intercluster relative positioning and distances are also meaningful. The main advantage of such a mapping is the ease by which a user gains an idea regarding the structure of the data by analyzing the cluster distribution on the map.

Traditional clustering algorithms in text mining are usually based on the Bag-of-Words (BOW) approach (Salton et al. 1975) for representing documents. The two main disadvantages of the BOW model are (a) that it treats all words the same way, ignoring any syntactic or semantic relationship among them and (b) that it uses a large vector space and, hence, introduces the curse of dimensionality.

In order to overcome the limitations of the BOW model, one possible idea would be to incorporate some sort of external knowledge into documents representation. For example, Wikipedia has become one of the largest knowledge repositories and new content is being added to it daily by users around the globe. As a corpus for knowledge extraction, Wikipedia's advantages are not limited to its size and the easiness of article updating, but also comprise hierarchical category organization, dense link structure between articles, sense disambiguation capability based on the URLs, brief anchor texts and well structured sentences.

This paper utilizes a way to simultaneously enrich and compress document representation with background knowledge provided by Wikipedia, in order to enhance performance in a clustering task. The integration of Wikipedia articles into the document representation is performed by mapping one or more words of the document -forming a topic- to the corresponding Wikipedia article, creating what we call document concepts, instead of a bag of words.

A SOM model is adjusted to the described document representation model in order to produce clusters with labels (conceptual clustering), informative of the content of the documents in a collection assigned to each specific cluster.

The remainder of this paper is structured as follows: Section 2 provides a brief presentation of the related work in the field. Section 3 describes the document representation model based on concept extraction using Wikipedia. The Self-organizing Map architecture is presented in detail in Section 4. The characteristics and capabilities of the proposed method are studied in Section 5. Finally, some remarks conclude the paper in Section 6.

## 2 Related work

Most text mining techniques use the BOW model for document representation, i.e. each document is represented by a feature vector, where each dimension corresponds to a term (Salton and McGill 1983). The values of the vector reflect the weight of each term in the document (the well known *tf-idf* model, Jones 1972) and similarity between documents is measured by any vector distance metric such as cosine, *Dice*, *Jaccard* e.t.c. Vector space model introduces high dimensionality in the input space and ignores semantic information or conceptual patterns of documents (e.g. it breaks multi-word expressions or considers polysemous words as being the same entity). In this high dimensional space, feature selection and feature weighting are to affect positively the clustering results (Mitra et al. 2002), but cannot overcome the above BOW problems. There have been efforts to semantically enrich the BOW model (which is based only on terms) by analyzing terms on phrase level and introducing a new phrase-based indexing model for documents (Hammouda and Kamel 2004) or by analyzing terms on sentence, document and corpus level and introducing document matching based on this analysis (Shehata et al. 2010). These directions, although promising, are based solely on exploiting the textual information contained inside the documents and do not make any use of the huge amount of background knowledge available nowadays, mostly thanks to the Semantic Web and in particular to the large number of ontologies that are accessible to us.

Motivated by the wish to harness this wealth of information, there has been a growing amount of research aiming to enhance text clustering by introducing external knowledge. Contextual information (Pullwitt 2002) was an early attempt to this goal. A two-stage model was introduced using features based on sentence categories (as an alternative approach to the original vector space model) which included contextual information. At the first stage, a set of sentence categories was extracted and at the second stage it was used to describe a new vector space for document representation. However, the most promising direction seems to be the exploitation of external knowledge provided by ontologies. Emerging research in ontologies provides many novel methodologies for enriching the analysis and visualization of documents by using semantic features (Breaux and Reed 2005). Early work utilized the semantic lexicon WordNet (Sedding and Kazakov 2004) in order to enhance document representation. WordNet's dense hierarchical structure was used to perform syntax-based disambiguation by assigning to each word a Part-of-Speech tag and by enriching the BOW data representation with synonyms and hypernyms. Apparently, the noise introduced by incorrect senses retrieved from WordNet developed to be a bottleneck for its use in document enrichment. Hotho et al. (2003) improved baseline clustering results by utilizing WordNet's hypernym relations and performing word sense disambiguation combined with feature weighting.

More recently, Wikipedia, the online encyclopedia has been used in many applications involving information retrieval (Li et al. 2007), entity extraction (Toral and Munoz 2006), text categorization (Gabrilovich and Markovitch 2006; Wang et al. 2009) and text clustering (Hu et al. 2009; Bloehdorn et al. 2006). Wikipedia does not provide (in its original shape) a fully structured knowledge base (ontology), but many researchers work towards this direction, i.e. turning Wikipedia into a full-fledged ontology (Bizer et al. 2009; Suchanek et al. 2008; Navigli and Ponzetto 2010).

Using these organized forms of Wikipedia or its structured counterpart DBpedia (Bizer et al. 2009), there is ongoing research in the field of named entity recognition (Cucerzan 2007), giving the possibility to map entities to people, organizations, locations and so on.

Gabrilovich and Markovitch (2006, 2007) propose a method to improve text classification performance by enriching document representation with Wikipedia concepts. The mapping between each document and Wikipedia concepts is achieved through a feature generator which acts like a retrieval engine. It receives a text fragment, which can be words, sentence, paragraph, or the whole document, and outputs the most relevant Wikipedia articles to the text fragment. The titles of the retrieved Wikipedia articles are further filtered and those with high discriminative capacity are used as additional features to enrich the representation of the corresponding documents. Empirical evaluation shows that their method can greatly improve classification performance.

Wikipedia has also been applied for text clustering. Banerjee et al. (2007) use a method similar to the one applied in Gabrilovich and Markovitch (2006) for clustering short texts. Their method is different in that they use query strings created from document texts to retrieve relevant Wikipedia articles. The titles of top-ranked Wikipedia articles serve as additional features for clustering Google news. Both methods of Banerjee et al. (2007) and Gabrilovich and Markovitch (2006) only augment document representation with Wikipedia article content without considering the hierarchical structure of Wikipedia or other features of Wikipedia articles. Wikipedia category information has also been used in Wang and Domeniconi (2008) for text categorization and in Hu et al. (2008) for text clustering. These methods extend the Wikipedia concept vector for each document with synonyms and associative concepts based on the redirect links and hyperlinks in Wikipedia.

Huang et al. (2009) developed a similarity measure that evaluates the semantic relatedness between concept sets for two documents, thus documents are connected not only through the coexistence of concepts but also due to concept relatedness. Kiran and Shankar (2010) enrich document representation using information from several knowledge bases (Wikipedia, dmoz, social bookmarks) and extract topics (clusters) from documents using various topic detection techniques.

All of the papers mentioned above, rely on existing clustering techniques (mostly Hierarchical Agglomerative Clustering and $k$-Nearest Neighbors) or other topic based approaches. Agglomerative clustering technique is stable and produces consistent results (providing a dendrogram shape) but is computationally intensive for large document sets. Divisive hierarchical methods (such as the *k-NN* algorithm) are less computationally costly and generally, perform better than the agglomerative methods (Steinbach et al. 2000). There have been attempts in divisive document clustering, to split $n$-way, with $n$ fixed at each level (Larsen and Aone 1999). Fixing the number of clusters at each level imposes a subjective composition of clusters that hinders the ability to discover the inherent structures of the dataset.

Statistical analysis has been also used in text clustering and document organizing. Information theoretic methods (Slonim et al. 2002) and probabilistic approaches (Liu et al. 2002; Hofmann 1999; Vinokourov and Girolami 2002) have been used to define the suitable number of clusters and good results have been reported. The main drawback of these approaches is that they often assume a word distribution or a specific model for each cluster and may need several runs until a stable representation is

achieved. Moreover, conceptual clustering approaches using probabilistic concepts instead of logical representations have been used (Talavera and Bejar 2001).

Other text clustering methods include: summarizing data into subclusters and then generating Gaussian mixtures for their summaries (Jin et al. 2005), building a basis for the application of inductive logic programming into text mining (Junker et al. 2000) and automatically extracting rules for text documents (Soderland 1999).

Neural networks that have been applied to document clustering include the fuzzy Adaptive Resonance Theory (ART) (Carpenter et al. 1991) and the self-organizing maps (Kohonen et al. 2001). ART networks are suitable for document clustering as they can adapt to deal with non-stationary data and appearance of new classes. However, their performance depends upon the order in which input data is processed and they are sensitive to over-fitting and noise (He et al. 2002). The SOM-based methods have two distinct advantages over other document clustering methods. First, they perform non-linear dimensionality reduction, so the input space is mapped onto a lower dimensional space with minimum information distortion. Second, spatial organization of the feature map is achieved after the learning process. This means that the topology preserving clustering causes that similar documents or topics are located closely on the map. The output is usually displayed in a 2-dimensional map on which, in general, a number of seperate clusters can be distinguished, this way helping navigation, browsing and discovery of similar documents. SOM has been applied successfully to text mining tasks (Merkl 1998; Lin et al. 1991; Kohonen et al. 2000). In order to provide more efficient abstractions and different levels of detail, a hierarchical variant of the SOM was introduced (Miikkulainen 1990), having also the advantage of reduced computational cost. However, the sizes of the maps at all levels have to be fixed, so imposing a predefined value on the number of clusters for potentially unknown datasets. This could lead to either indiscriminate (small) maps or over representing (large) maps with underused or overused resources. The growing variants were introduced to address these issues, integrating the hierarchical structure e.g. the growing hierarchical SOM (GH-SOM) (Rauber et al. 2002) which gives good results by abstracting levels of details with dynamic map sizes. However, the map growing depends on a sensitive parameter, set a priori and only one map can be viewed at a time.

An important part of visualization and exploration of the SOM-based methods is to automatically identify and label regions of interest. Multilayered SOMs have been used to organize documents (e.g. the ET-MAP, Chen et al. 1996), where each node is labeled using the best matching terms and nodes with similar terms are merged together to form regions on the maps. This is similar to the approach adopted in the LABELSOM (Rauber 1999), which uses the quantization error to determine the best terms to label the nodes without merging nodes into regions. The WEBSOM uses the term frequency-based measure to determine the labels (Lagus et al. 2004). The method looks at the frequencies of the most discriminating terms occurring in a cluster. The WEBSOM uses more nodes than the other methods, so not every node is labeled, but there is a rather fixed radius of minimum distance between labels. On the map, cluster densities are coloured and smoothed, making them more visually appealing than the ET-MAP and GH-SOM. The WEBSOM also reduces the term vector space size using a random projection to create the document vectors.

However, some difficulties in SOM utilization remained largely untouched, even though a large number of research papers on applications of the SOM were presented

over the years. First, the SOM uses a fixed network architecture in terms of number and arrangement of neural processing elements, which has to be defined prior to training. Obviously, in case of largely unknown input data characteristics, it remains far from trivial to determine the network architecture that provides satisfying results.

Second, hierarchical relations between the input data are not mirrored in a straightforward fashion. Such relations are rather shown within the same representation space and are hard to identify. Hierarchical relations, however, may be observed in a wide spectrum of application domains. Thus, their proper identification remains a highly important data mining task that cannot be addressed conveniently within the framework of the SOM.

In this paper we use the idea of important concepts (Fung et al. 2003; Spanakis et al. 2011), in order to properly initialize the number of neurons and their parameters in a self-organizing map model. The proposed approach takes into account both the relative significance of concepts -extracted by exploiting various features of the ontology- and their frequency in the corpus. The method is able to produce both a hierarchical structure tree with automatically extracted labels for each cluster, (according to the content of the documents assigned to it) and a visualization of the clusters produced.

## 3 Document representation model using Wikipedia

The proposed approach for enriching document representation through Wikipedia knowledge exploitation is presented in Fig. 1 and was first described in our previous work (Spanakis et al. 2011).

### 3.1 Concept extraction from Wikipedia

Our goal is to extract Wikipedia concepts which are described by one or more consecutive words of the document. For example, if a document contains the phrase "data mining", it is desirable to extract both words as an entity and map it to the corresponding Wikipedia article, thus forming a document concept. Please note that, in the same situation, the BOW method would have broken the previous semantic entity into the far more ambiguous singletons "data" and "mining". However, the complexity of extracting all possible document N-grams in order to check whether or not they are mapped onto an existing Wikipedia article is too high, as mentioned in Wang and Domeniconi (2008), and methods to reduce it mostly rely only on restricting document topics to a specific domain (Wang et al. 2003). In recent years, however, several methods have been introduced in order to correctly spot named entities such as the Stanford Named Entity Recognizer (Stanford 2009) or the LingPipe Exact Dictionary-Based Chunker (Alias-i 2008) with promising results.

In our approach, we overcome the bottleneck of N-grams by choosing to annotate each document's text with Part-of-Speech (POS) information using the TreeTagger tool provided by Schmid (1994). Wikipedia articles have descriptive titles, so it is not necessary to perform stemming or remove stop words during document preprocessing. After this procedure, we keep those consecutive words which are nouns and proper nouns (singular or mass or plural) along with prepositions, subordinating or coordinating conjunctions and the word *to* (POS tags in the Penn Treebank

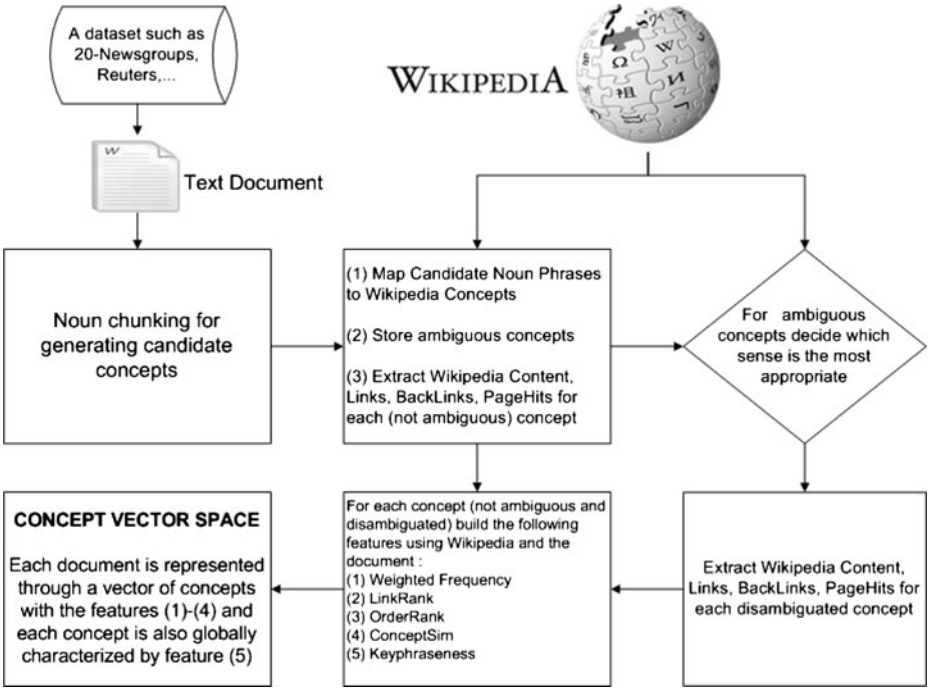


**Fig. 1** Representing a document by its Wikipedia matching concepts

Tagset, Marcus et al. 1993). By grouping all consecutive words in the text having one of the previous POS tags we perform full *Noun Phrase* extraction (e.g. we can extract both "Barack Obama" and "The President of the USA"), while reducing the computational cost of considering every N-gram of the text, including verbs e.t.c. The extracted Noun Phrases form our candidate concepts.

For each candidate concept, we automatically check "on-the-fly" whether it exists or not as a Wikipedia article using the Wikipedia API (2011). If the concept has multiple senses (so there are multiple Wikipedia articles referring to the same Noun Phrase), we perform the disambiguation process described in the next section, in order to choose the most appropriate sense. Once we obtain a unique mapping between the candidate concept and Wikipedia, the concept is selected as a component of the document vector which is about to be formed. For example, we will consider a text fragment from a document from 20 Newsgroups dataset. 20 Newsgroups (20-NG) (Lang 1995) contains approximately 20,000 documents of the full 20-newsgroup collection of USENET news group articles. Each newsgroup belongs to a different category, with varying overlap between them: some newsgroups are very related (e.g. comp.os.ms-windows.misc and comp.windows.x) and others are not related at all (e.g. sci.electronics and alt.atheism). In the following text fragment (from document #59284 of the 20-NG dataset), the extracted concepts are shown in **bold**.

***Depo Provera*** *was developed in the* ***1960s*** *and has been approved for* ***contraception*** *in many other* ***countries****. The* ***UpJohn Company*** *of Kalamazoo, Mich., which will* ***market*** *the* ***drug*** *under the* ***name****,* ***Depo Provera*** *Contraceptive Injection, first*

*submitted it for **approval** in the **United States** in the **1970s**. At that **time**, **animal studies** raised **questions** about its **potential** to cause **breast cancer**.*

It is obvious, how important to a text retrieval task is the ability to find concepts such as "Depo Provera", "UpJohn Company", "United States" which would be broken into two words with no specific content (e.g. "Depo" or "Provera") or with other meaning (e.g. "United" or "States").

At the same time, using the Wikipedia API, for every selected concept $i$, we extract the features presented in Fig. 2.

After the extraction of the features described in Fig. 2 for every concept $i$ in a document $j$, we combine them, to form new document features, as described in the equations below, in order to form a richer document representation.

- Weighted Frequency ($Wfreq$) is defined by:

$$WFreq_{j,i} = size_i * frequency_{j,i} \tag{1}$$

where:

| | |
|---|---|
| $size_i$ | is the number of words that form concept $i$, |
| $frequency_{j,i}$ | stands for how many times concept $i$ occurs in document $j$. |

- $LinkRank$ is a measure of how many links a concept has in common with the total of those contained in a document, thus it is a measure of the importance of the concept to the document and is formally defined as:

$$LinkRank_{j,i} = \frac{|Links_i \bigcap Links_{Doc_j}|}{|Links_{Doc_j}|} \tag{2}$$

where:

| | |
|---|---|
| $Links_i$ | is the set of Links of concept $i$, as defined in Fig. 2, |
| $Links_{Doc_j}$ | is the set of Links of document $j$, defined as all the links of all concepts that represent document $j$. |

- $Content_i$: the corresponding Wikipedia article text
- $Links_i$: links from the corresponding article to other articles
- $BackLinks_i$: articles which have a link to the examined article
- $PageHits_i$: the articles in which the examined article (Noun Phrase) is simply present, either as link or not (plain text)
- $Categories_i$: the list of categories that the corresponding article belongs to

**Fig. 2** Concept features as extracted from the Wikipedia API

- *ConceptSim* is the similarity between the document and the article text of a concept contained in the document, computed in the classic term frequency—inverse document frequency ($tf - idf$) vector space, as defined in Salton et al. (1975) and is given by the following equation:

$$ConceptSim_{j,i} = \cos(\mathbf{v}_j, \mathbf{v}_i) \tag{3}$$

where:

| | |
|---|---|
| $\mathbf{v}_j$ | is the $tf - idf$ vector of document $j$, |
| $\mathbf{v}_i$ | is the $tf - idf$ vector of the Wikipedia article text corresponding to concept $i$, |
| cos | is the cosine function which computes the similarity between the two vectors. |

- *OrderRank* is a measure which takes larger values for concepts that appear at the beginning of the document, based on the observation that important words often occur at the beginning of a document (Xue and Zhou 2009). Formally it is defined as:

$$OrderRank_{j,i} = 1 - \frac{arraypos_i}{|j|} \tag{4}$$

where:

| | |
|---|---|
| *arraypos* | is an array containing all words of the document in the order that they occur in the document, and $arraypos_i$ represents the position of the first occurrence of concept $i$ in the array. If a concept consists of more than one word, then we take into consideration the position of occurrence of the first word of the concept. |
| $\|j\|$ | is the size of document $j$, i.e. how many words form the document. |

Additionally, we define a global (document independent) measure for each concept defined as follows:

- *Keyphraseness* is a measure adapted from (Mihalcea and Csomai 2007), which has a specific value for each different concept, regardless of the document we refer to, and is an indication of how much descriptive and specific to a topic a concept is. It is defined as:

$$Keyphraseness(i) = \frac{BackLinks_i}{PageHits_i} \tag{5}$$

A concept with high *Keyphraseness* value (i.e. most of its occurrences in Wikipedia are links to the corresponding article and not plain text) has more descriptive power than a concept with low *Keyphraseness* value, even if the latter may occur more times in Wikipedia, but less times as a link. *Keyphraseness* is normalized in the interval [0, 1], after the extraction of all concepts from all documents in the corpus, so that the highest *Keyphraseness* value is set to 1 and the lowest to 0.

### 3.2 Concept disambiguation process

If a candidate concept is polysemous, i.e. it has multiple meanings, it is necessary to perform word sense disambiguation to find its most proper meaning in the context where it appears. Many word-sense disambiguation techniques based on Wikipedia resources have been proposed with very satisfying results (Milne and Witten 2008; Ratinov et al. 2011; Bunescu and Pasca 2007; Mendes et al. 2011). However, in order to speed up the procedure of mapping concepts, we adopt a simple and fast, yet effective disambiguation strategy.

*ConceptSim* (as introduced by (3)) is utilized to do explicit word sense disambiguation. It is therefore reminded that *ConceptSim* is based on the cosine similarity between the $tf - idf$ vectors of the document and the examined concept, so the larger the value of *ConceptSim* is, the higher the similarity between the two corresponding text documents is. Thus, the sense with the higher *ConceptSim* is the most appropriate for the examined document. However, in order to provide more accurate disambiguation results we use in addition the categories that each concept belongs to, and we integrate it to the *ConceptSim*, creating a more robust measure of how similar is a concept ($c$) to the examined document ($j$) which will be called $SenseSim_{j,c}$ and is defined by the following equation:

$$SenseSim_{j,c} = \lambda * ConceptSim_{j,c} + (1-\lambda) * Dice(Categories_c, Categories_{Doc_j}) \quad (6)$$

where:

| | |
|---|---|
| $ConceptSim_{j,c}$ | is given by (3), |
| $Categories_c$ | shows the Categories of concept $c$ as defined in Fig. 2, |
| $Categories_{Doc_j}$ | shows the Categories of document $j$, defined as all the categories of all (non-ambiguous) concepts that represent document $j$, |
| $\lambda$ | is a parameter in [0, 1] to weight the text and category overlapping metrics, |
| *Dice* | is the well-known co-efficient defined as: |

$$Dice(A, B) = \frac{2 * |A \bigcap B|}{|A + B|} \quad (7)$$

For instance, the 20-NG dataset, document #67480 belongs to the category comp.windows.x, and the concept "client" in wikipedia refers to several different senses of the word, as listed in Table 1. A small fragment of the context of #67480 document in which the word "client" occurs is the following:

**Table 1** Disambiguation results for concept "client" in document #67480 of 20NG collection

| "Client" senses | *SenseSim* |
|---|---|
| Client (computing) | 0.0578 |
| Client (ancient Rome) | 0.0240 |
| Client (band) | 0.0170 |
| Clients (album) | 0.0168 |
| Client (album) | 0.0097 |

**Table 2** Example representation vector [0, 1] normalized

| *Concept* | *W freq* | *Link Rank* | *Concept Sim* | *Order Rank* | *Keyphraseness* |
|---|---|---|---|---|---|
| Hypertext transfer protocol | 0.9333 | 0.8830 | 0.6628 | 0.0235 | 0.8932 |
| Software versioning | 0.9000 | 0.7395 | 0.6968 | 0.4718 | 0.8324 |
| Software portability | 0.9000 | 0.8601 | 0.7257 | 0.1549 | 0.8153 |
| Web ontology language | 0.9333 | 0.9572 | 0.7289 | 0.3967 | 0.7679 |
| Application software | 0.9000 | 0.9129 | 0.6494 | 0.1620 | 0.7290 |
| Function (computer science) | 0.9000 | 0.5599 | 0.8021 | 0.3521 | 0.0116 |
| Pascal (programming language) | 0.4000 | 1.0000 | 0.9206 | 0.4484 | 0.0009 |
| ... | ... | ... | ... | ... | ... |

*Since the server end is (or was) always at this end (California) it is faster to remotely run the client via DESQview X and have a short hop to the server than running the client locally and having a long hop to the server.*

*SenseSim* measure is computed for each one of these senses, and the meaning with the larger value is selected to be part of the document representation (in this case it is obvious, that the "client" is used with its computing sense). In order to make the disambiguation process more precise and to avoid adding to the documents incorrect senses, it is possible to introduce a *SenseSim* minimum threshold for replacing an ambiguous concept with its most proper sense (otherwise the concept will be dropped). In our experiments this threshold was set to 0.05.

### 3.3 Document representation

After completing the disambiguation process, we end up with a set of concepts which represent the document. Our goal is to construct a vector representation where each component corresponds to the importance of each concept in the document. As previously stated, each concept has four features related to the document which are described by (1) through (4) and one "global" feature which is described by (5).

For instance, a small part of the document concept representation for the #67480 article from the 20-NG dataset is shown in Table 2. The measures *W Freq*, *Order Rank*, *Link Rank*, and *Concept Sim* of the whole document are normalized in the interval [0, 1], as explained above for *Keyphraseness*. Notice the ability to represent as a concept the words "Hypertext Transfer Protocol", which the BOW model would have broken into three words and would have led to loss of descriptive value.

## 4 Self-organizing document map

### 4.1 Self-organizing map (SOM) and visualization

SOM, as proposed in Kohonen et al. (2001) and described thoroughly in Kohonen (1989) and Kangas et al. (1990) is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional) representation of the input space of the training samples, called a map. Self-organizing maps are different from other artificial neural networks in the sense that

they use a neighborhood function to preserve the topological properties of the input space, thus similar items are grouped spatially close to one another.

SOM consists of a number of neural processing elements, i.e., units that are arranged according to some topology, the most common choice of which is marked by a 2D rectangular or hexagonal grid. Each of the units $i$ is further assigned a model vector $m_i$. It is important to note that these model vectors have the same dimensionality as the input patterns.

The training process of SOM may be described in terms of input pattern presentation and model vector adaptation. Each training iteration starts with the random selection of one input pattern $x$. This pattern is presented to the SOM and each unit determines its activation (competition phase). Usually, the Euclidean distance between input pattern and model vector is used to calculate a unit's activation. In this case, the unit having the model vector with the smallest Euclidean distance to the input pattern is designated as the winner. We will use the index $c$ for denoting the winner and $t$ for denoting the current training iteration (note that we make use of discrete-time notation):

$$c(t) = arg\min_i \|x(t) - m_i(t)\| \tag{8}$$

Finally, the model vector of the winner as well as model vectors of units in the vicinity of the winner are adapted. This adaptation is implemented as a gradual reduction of the difference between corresponding components of the input pattern and the model vector, as shown in (9):

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)] \tag{9}$$

Geometrically speaking, the model vectors of the adapted units ($m_i$) are moved a bit toward the input pattern. The amount of model vector movement is guided by a learning rate ($\alpha(t)$), decreasing in time. The number of units that are affected by adaptation, as well as the strength of adaptation depending on a units distance from the winner is determined by a neighborhood function.

Typically, the neighborhood function is a function which is symmetric around the location of the winner and monotonically decreasing with increasing distance from the winner. One commonly used neighborhood function is the Gaussian function:

$$h_{ci}(t) = \exp\left(-\frac{\|r_c - r_i\|^2}{2 \cdot \sigma(t)^2}\right) \tag{10}$$

In this expression, $\|r_c - r_i\|$ denotes the distance between units $c$ and $i$ within the output space, with $r_i$ representing the 2-D location vector of unit $i$ within the grid. The time-varying function $\sigma$ guides the reduction of effect of the neighborhood during training. It is common practice that this neighborhood kernel is selected large enough to cover a wide area of the output space in the beginning of learning. The spatial width of the kernel is reduced gradually during training such that toward the end of the process just the winner unit is adapted.

Many visualization techniques based on the SOM have been developed. Once the learning phase is complete, visual display of the map must be carried out in a way making possible to observe the underlying structure of the data. One of the most widely used methods is the unified distance matrix (U-matrix) (Ultsch and Siemon 1990). The U-matrix method uses a coloring scheme to measure and mark the relative

distances between prototype vectors associated with neighboring neurons. Because clusters are groups of vectors that are close to one another compared to their distance to the other vectors, high values in the U-matrix encode dissimilarity among neurons and correspond to cluster boundaries. On a second stage, the interneuron distances are represented by gray shade, and thus made visible. Different variants also exist. For example in Kraaijveld (1992) a grid (matrix) is used to represent the SOM, where its cells (elements) represent neurons. Each cell is shaded according to the average distance from the neuron to its neighbors, with white representing zero distance, and black the largest possible distance according to some implementation-specific measure.

An alternative way is to visualize the shape of the SOM by projecting the prototype vectors to a suitable low-dimensional output space. Because the cluster structure of the SOM is usually arbitrarily shaped or consists of curved manifolds, a nonlinear projection method, such as Multi-Dimensional Scaling (MDS) (Davison 1983), ISOMAP (Tenenbaum et al. 2000), LLE (Roweis and Saul 2000), Curvilinear Component Analysis (CCA) (Demartines and Herault 1997) and Sammon's mapping (Sammon 1969), is commonly used. Both MDS and Sammon's mapping aim at minimizing an error measure that is a function of the differences of the interpoint distances in the original space and the interpoint distances in the projected space. ISOMAP (Tenenbaum et al. 2000) is originally designed as a generalization of MDS, which computes the geodesic distances on the manifold and uses them for projection.

Another method of visualization is to show the responses of the prototype vectors to data samples. The typical way to show the response is to simply mark the corresponding Best Matching Unit (BMU) on the map. Multiple input vectors will result in a hit histogram of the winning nodes, which shows the distribution of the input space. Map units on cluster borders often have very few data samples, which mean very low hits in the histogram. Therefore, low-hit units can be used to indicate cluster borders. Some promising results have been reported using this method. However, hit histograms consider only the BMU for each data sample while real-world data is usually well represented by more than one unit. This inevitably causes distortions in the final map. A variation of the standard hit histogram, namely, the smoothed data histogram (SDH) (Pampalk et al. 2002), has been developed which counts the data sampleŠs relativeness to more than one map unit. The SDH allows a data sample to "vote," not only for the BMU but also for the next few good matches based on the ranks of distances between the data sample and the corresponding prototype vectors.

Aside from the previous categories, other visualization techniques are available for SOM. A rather different way to project the prototype vectors, the so-called adaptive coordinates (Merkl and Rauber 1997), was proposed to mirror the movement of the prototype vectors during the SOM training within a 2-D space. The visualization-induced SOM (ViSOM) (Yin 2002) has been used to directly preserve the distance information along with the topology on the map.

### 4.2 Our approach: document self-organizer (DoSO)

Generic clustering techniques have the disadvantage that, when applied to documents, they do not provide intrinsic textual descriptions of the clusters obtained, due to the fact that their algorithms were not designed specifically for text. On the other hand, existing conceptual clustering techniques for text are either known to

be rather slow (Stumme et al. 2002), or require extra steps to reduce the number of clusters (Hotho and Stumme 2002).

Our clustering method utilizes Self-Organizing Maps in combination with the idea of important concepts as implemented in our previous work in Spanakis et al. (2011). The goal is to provide a cluster description based on the Wikipedia concepts extracted from the corpus examined. The approach can be hierarchical, i.e. documents can be unified to several different groups depending on a threshold chosen by the user.

First of all, after representing each document with Wikipedia concepts (according to Section 3), we linearly combine the features described in (1) through (4), in order to find the weight of each concept in each document. The final weight of concept $i$ in document $j$ is given by the following equation:

$$\begin{aligned} Weight(j,i) = \alpha * WFreq_{j,i} + \beta * LinkRank_{j,i} + \\ +\gamma * OrderRank_{j,i} + \\ +(1-\alpha-\beta-\gamma) * ConceptSim_{j,i} \end{aligned} \tag{11}$$

The coefficients $\alpha$, $\beta$ and $\gamma$ are determined empirically by experiments and their value range is the interval $[0, 1]$ (see Table 3 in Section 5). This way, we replace the usually sparse BOW model by a more compact concept model, which both reduces the vector space size (an important factor for processing large amounts of oversized documents) and enriches document features with external knowledge from Wikipedia. Each document in the corpus is described by such a vector with its components reflecting the weight of each concept in the document.

Before proceeding with the main method description, let us introduce some definitions:

(a) A *global important concept* is a concept that:

- has a *Keyphraseness* value greater than a specific threshold, defined as *minimum keyphraseness threshold*, and
- appears in more than a minimum fraction of the whole document set, defined as *minimum global frequency threshold*.

Moreover, a *global important k-concept-set* is a set of $k$ global important concepts that appear together in a fraction of the whole document set greater than the minimum global frequency threshold.

(b) Our method uses neurons (just like the classic SOM) which are described by:

- a weight vector (*neuron vector*), which has the same dimension as the vector describing the documents of the corpus. Each vector component reflects the weight of a different concept in the neuron.
- a label (*neuron label*) which is defined by the global important concept-set(s) that are contained in all documents assigned to the neuron by a process that is clarified at the SOM initialization step.
- a position in the 2D space (*neuron position*), which is equivalent to the classic SOM's output space but adjusted to our model and that will be described later.

(c) A global important concept is *neuron important* in a neuron $N_m$, if the weight of the concept in the neuron vector is larger than a specific threshold, defined as *minimum neuron support*.

Our method is carried out in three steps. At the first step, the SOM neurons are determined and documents are assigned to them (based on the *Keyphraseness* of concepts and on the frequency of concepts and concept-sets). A projection based on ISOMAP method (Tenenbaum et al. 2000) is carried out in order to visualize the initial clustering result in the 2-dimensional (2D) space. At the second step, the competition phase of SOM is carried out. At the third step, the weight updating and cooperation of neurons is performed. Finally, a process to discover the clusters of the examined corpus is carried out, generating a hierarchical structure.

*Phase 1: initial neurons selection and initialization* Given the definitions (a) and (b) above, we can determine the neurons (number, labels, initial weight vectors and positions) of our model. For every concept-set that corresponds to the restrictions of definition (a), we construct a neuron comprising all documents that contain this concept-set. We omit $(k-1)$-concept-sets if their concepts appear as $k$-concept-sets (e.g. if *"astronomy"* and *"comet"* are global important concepts but also *"astronomy-comet"* is global important concept-set, we create a neuron only for the *"astronomy-comet"*-concept-set). It is obvious that in this phase, a document may be assigned to more than one neurons. The disjoining of documents/neurons, by determining the winner neuron for each document, is carried out later on.

At this point, the neuron vectors (as defined by definition (b)) are initialized with respect to the documents assigned to them according to the following equation:

$$NW(k,i) = \sum_{j \in M_{ki}} \frac{Weight(j,i)}{|M|} \tag{12}$$

where:

| | |
|---|---|
| $NW(k,i)$ | is the weight of concept $i$ in neuron $k$, |
| $M_{ki}$ | is the set of documents initially assigned to neuron $k$ that contain concept $i$, |
| $Weight(j,i)$ | is the weight of concept $i$ in document $j$, as defined by (11), |

Concepts that have $NW$ value higher than a specific threshold specified by the user (see definition (c)) form the *neuron important* concepts of the neuron. It must be highlighted that the initialization process is largely considered and empirically verified to be important for a faster training of the SOM.

After the determination of neurons number and the initialization of their weight vectors, the most suitable neuron for every document must be found and the document should be assigned only to this neuron. For this purpose, we introduce a measure of how "good" a neuron $N_m$ for a document $Doc_j$ is by defining a function $Score(N_m \leftarrow Doc_j)$. The neuron with the highest value for this measure is selected as the unique neuron for the examined document. If there are more than one neurons

that maximize the *Score* function, the one with the largest concept-set at its label is chosen. The *Score* function is defined as follows:

$$Score(N_m \leftarrow Doc_j) = \sum_x \left[ Weight(j, x) \cdot NW(m, x) \right] - \sum_{x'} \left[ Weight(j, x') \cdot Keyphraseness(x') \right] \quad (13)$$

where:

| | |
|---|---|
| $x$ | represents a global important concept in $Doc_j$, which is neuron important in $N_m$, |
| $x'$ | represents a global important concept in $Doc_j$, which is not neuron important in $N_m$, |
| $Weight(j, x)$ | is the weight of concept $x$ in $Doc_j$ as defined in (11), |
| $Weight(j, x')$ | similarly as above but for $x'$, |
| $NW(m, x)$ | is given by (12), |
| $Keyphraseness(x')$ | is given by (5). |

The first term of the score function rewards neuron $N_m$, if a global important concept $x$ in $Doc_j$ is neuron important in $N_m$. The importance of concept $x$ in different neurons is captured by multiplying its weight in the document $Doc_j$ by its neuron weight in $N_m$. The second term of the function penalizes neuron $N_m$ if a global important concept $x'$ in $Doc_j$ is not neuron important in $N_m$. The weight of $x'$ in $Doc_j$ is multiplied by its *Keyphraseness* value which expresses how important is the concept in general terms.

The neuron that has the greatest *Score* value is the "winner" neuron, at the initialization phase:

$$winner(Doc_j) = \arg\max_m \left\{ Score(N_m \leftarrow Doc_j) \right\} \quad (14)$$

Once the unique assignment of documents to neurons is over, we compute the distances between all neurons in the input space using the following equation:

$$D(N_m, N_n) = \sqrt{\sum_i \left[ (NW(m, i) - NW(n, i))^2 \right]} \quad (15)$$

where $N_m$ and $N_n$ are the two neurons whose distance is calculated, $NW$ stands for the corresponding neuron weights and the summation takes place for all concepts $i$ of the corpus.

We conclude Phase 1 by obtaining a visualization of the DoSO in 2 dimensions using the ISOMAP method (Tenenbaum et al. 2000). Initially, ISOMAP defines the neighborhood graph between all neurons and connects each one with its $k$ nearest neighbors ($k$ is a user specified parameter and for the distances we use (15)). Then, the *graph distances* (defined as the sum of edge weights along the shortest path between two nodes) are computed. Finally, the top $n$ eigenvectors of the *graph distance matrix*, represent the coordinates in the new $n$-dimensional Euclidean space. By setting $n$ to 2, the neurons are now projected into a 2D space, which can be used for applying the SOM algorithm used for training. An example of this visualization can be seen in Fig. 3. Some neurons have already been separated from the others

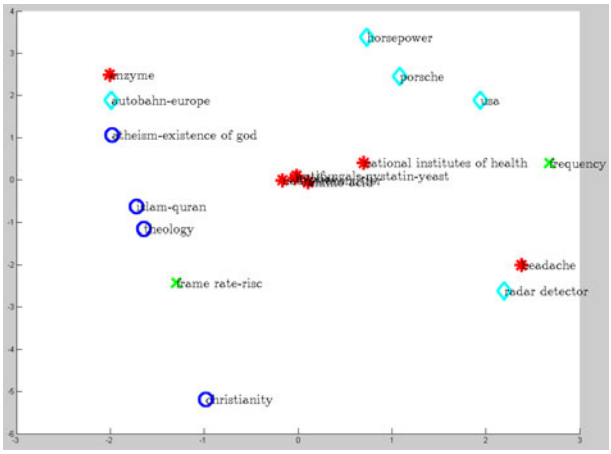


**Fig. 3** DoSO (4 classes) after initialization. ∘ = atheism, × = graphics, ∗ = medicine, ◊ = autos

forming independent clusters, but others need further processing (training) in order to clearly separate. Note that each neuron is assigned an exact position in the 2D space ($\mathbf{r}_m = (x_m, y_m)$) (which is the *neuron position* as described in definition (b)) and we can define the distance between two neurons in the 2D space by the following equation:

$$\Delta(N_m, N_n) = \|\mathbf{r}_m - \mathbf{r}_n\| = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2} \tag{16}$$

*Phase 2: training and competition* After neuron initialization, the training process begins, based on (8) through (10), which are adapted to our system. Moreover, in our approach, the neuron positions (as computed at the initialization step) are changing dynamically based on an updating rule similar to that of classic SOM training. Thus, the *competition phase* involves the decision of which neuron is most suitable for each document $j$ according to the following equation:

$$Sim(N_m, j) = \sum_i \left\{ Weight(j, i) \times NW(m, i) \right\} \tag{17}$$

where the summation takes place for all concepts $i$ of document $j$. Obviously, the winner neuron $m^*$ is the one that maximizes the *Sim* function:

$$m^* = \arg\max_m \{Sim(N_m, j)\} \tag{18}$$

The distance between two neurons $N_m$ and $N_n$ is measured using (15). The neighborhood function is defined as follows:

$$h_{m^*,m}(t) = \exp\left( -\frac{\Delta\,(N_{m^*}, N_m)^2}{2 \cdot \sigma(t)^2} \right) \tag{19}$$

where:

$\Delta$ is the neuron (Euclidean) distance in the output 2-D space, as defined in (16),
$\sigma$ is the width parameter reducing through time.

After the winner neuron is selected for document $j$, we update the weights of neuron vectors (for every concept $i$) according to the following equation:

$$NW(m,i)^{t+1} = NW(m,i)^t + \\ +\eta(t) \cdot h_{m^*,m}(t) \cdot \left[Weight(j,i) - NW(m,i)^t\right] \tag{20}$$

where:

| | |
|---|---|
| $NW(m,i)^{t+1}$ | stands for the updated weight of concept $i$ in neuron $m$, |
| $NW(m,i)^t$ | stands for the old weight of concept $i$ in neuron $m$, |
| $Weight(j,i)$ | is given by (11), |
| $\eta$ | is the learning rate, which decreases through epochs, |
| $h_{m^*,m}$ | is the neighborhood function defined by (19), |

Moreover, we update the neuron position vectors (in the 2D space) according to the equation:

$$\mathbf{r}_m^{t+1} = \mathbf{r}_m^t + \zeta(t) \cdot H_{m^*,m}(t) \cdot \left[\mathbf{r}_{m^*}^t - \mathbf{r}_m^t\right] \tag{21}$$

where:

| | |
|---|---|
| $\mathbf{r}_m^{t+1}$ | stands for the updated neuron $m$ position in 2-D space, |
| $\mathbf{r}_m^t$ | stands for the old neuron $m$ position in 2-D space, |
| $\zeta$ | is the learning rate, which decreases through epochs, |
| $H_{m^*,m}$ | is the neighborhood function defined by the following equation: |

$$H_{m^*,m}(t) = \exp\left(-\frac{D(m^*,m)^2}{2 \cdot \sigma'(t)^2}\right) \tag{22}$$

where:

| | |
|---|---|
| $D$ | is the neuron weight vector distance, as defined in (15), |
| $\sigma'$ | is the width parameter reducing through time (can be the same as $\sigma$ in (19)) |

By this double training scheme, we manage to update both neuron weight vectors and positions at the 2-D space, so by the end of training process we expect (i) the weight vectors to reflect the importance of each concept in each neuron and (ii) the creation of neighborhoods of similar neurons (in terms of their weight vectors) in the 2D space. The final positions of the Fig. 3 neurons are shown in Fig. 4 where it is obvious that further training led to clearly separated neurons (and subsequently document classes).

*Final phase: cluster discovery and hierarchical structure* When self-organizing maps are used for clustering, defining the clusters on the SOM becomes a crucial task. Several fairly complex approaches have been developed. In Himberg (2000) and Vesanto and Alhoniemi (2000) a node is iteratively updated during training based on the learning vectors such that a well-trained SOM represents a distribution of the input data over a two-dimensional surface preserving topology. In this context we can define a cluster as a group of nodes with short distances between them and long distances to the other nodes. In Moutarde and Ultsch (2005), a method based on representation of SOM as a grid (matrix) is utilized. However, as it is necessary with

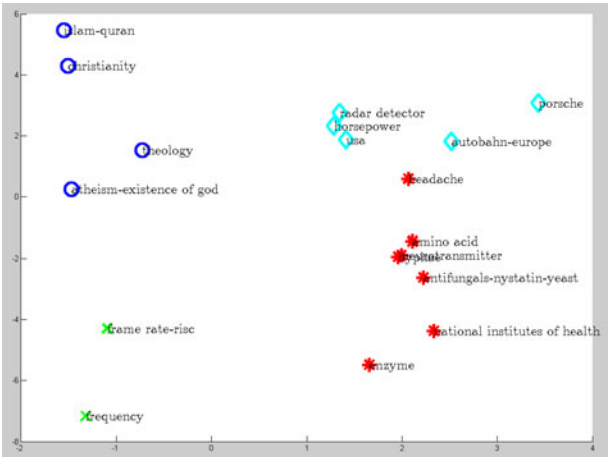


**Fig. 4** DoSO (4 classes) after training. $\circ$ = atheism, $\times$ = graphics, $*$ = medicine, $\Diamond$ = autos

this method to designate some neurons as cluster borders, a larger SOM is required to achieve the same level of detail.

Our algorithm for Cluster Discovery is described in Fig. 5. We start by selecting a neuron $N$ from the set of unselected neurons ($UN$: initially contains all neurons). We initiate a new cluster $C$ containing initially only neuron $N$ (the neuron label is also the initial cluster label). We remove this neuron from $UN$ and then we find those neurons $A$ which fulfil two conditions: (a) they belong to the $k$ nearest neighbors of $N$, where $k$ is selected in the step of visualization (Phase 1) and (b) their distance to $N$ is smaller than a specified threshold $T$, which actually defines the number of clusters that finally will be created. Scaling all distances so that the largest distance between two adjacent neurons equals 1, gives the ability to express distance thresholds as values between 0 and 1, regardless of the actual distances. If for neuron $A$ conditions

**Fig. 5** Description of cluster discovery method

*Cluster Discovery using DoSO*

$Cluster\_Discovery(T)$
Mark all neurons as unvisited $UN = \{Allneurons\}$
**while** $UN \neq \varnothing$ **do**
- Find a neuron $N_i$ from $UN$
- Start a new cluster $C$
- Call $Cluster\_New(C, N_i, T)$

**end while**

$Cluster\_New(C, N, T)$
$C = C \cup \{N\}$
$UN = UN - \{N\}$
**for all** $A \in UN$ which are neighbors of $N$ and $D(N, A) \leq T$ **do**
Call $Cluster\_New(C, A, T)$
**end for**

(a) and (b) hold, then it is added to the existing cluster $C$ (and its label is added to the cluster label) and we recursively apply the same procedure to the neurons adjacent to $A$. If not, we select another neuron from $UN$ and initiate a new cluster. Obviously, the larger the value of $T$, the smaller the number of clusters will be. By choosing different thresholds $T$, we can find a different clustering using the algorithm of Fig. 5.

Note that distances between nodes within a cluster may not necessarily be all smaller than the distance threshold; however, every node must be connected to every other node within the cluster along a path consisting exclusively of edges shorter than the distance threshold. In practice, this means that each node within a cluster must be connected to at least one other node within the same cluster with an edge that is shorter than the distance threshold. The described algorithm may be applied either to a fully trained SOM to discover the final clustering, or to any intermediate SOM snapshot as a monitor of the training progress or even as a part of the termination test. Also, since the algorithm is independent of the SOM training algorithm as well as of the definition of node neighborhood and adjacency, it is in principle applicable to any SOM variant.

By selecting different thresholds, it is straightforward to obtain a hierarchical structure for the clusters of each document corpus we deal with; selecting small values for threshold $T$ leads to many clusters (containing few documents) but larger values of $T$ will force more neurons to merge, forming larger clusters (which contain the smaller ones). Thus, a full hierarchy in a tree form can be generated, by selecting only a few different values for threshold $T$. An example of the hierarchy created for the comp.windows.x cluster of 20-Newsgroup category, is shown in Fig. 6. At the lowest level ($T = 0$) each neuron corresponds to a different cluster (boxes contain

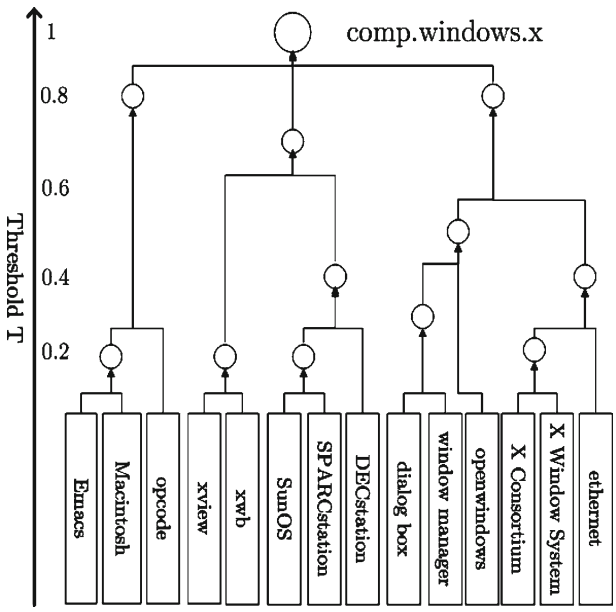


**Fig. 6** 20-Newsgroup comp.windows.x category hierarchy example

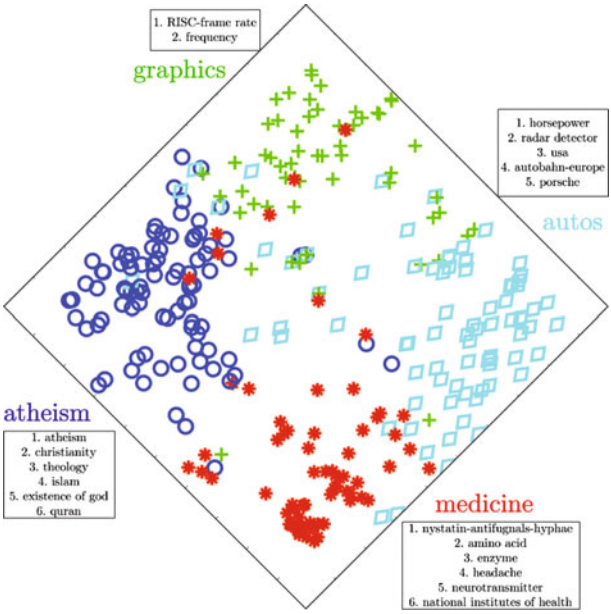


**Fig. 7** Example of how some documents are clustered in the 4-dimensional simplex

the neuron label) and as we move upwards to the hierarchy tree, the value of $T$ increases, leading to neuron mergers (denoted by the circles).

Also, in Fig. 7 we represent an example of a 4-dimensional semantic simplex, where each vertex corresponds to a class (atheism, medicine, autos and graphics). The boxes nearby the vertices represent the neurons labels (after the Cluster Discovery process) and inside the simplex the documents are placed according to their distance from each class. This figure reveals the fact that *atheism* and *medicine* classes are more robust and homogeneous (concepts from corresponding documents are too specific and hardly appear in other classes' documents) whereas *graphics* and *autos* classes are less robust and show greater overlap (because they share more common concepts).

## 5 Experiments

We evaluated our method by comparing its effectiveness with two of the most standard and accurate document clustering techniques (Steinbach et al. 2000): Hierarchical Agglomerative Clustering (HAC) and k-Nearest Neibghbor (*k-NN*). *HAC* builds a corpus hierarchy from the individual documents by progressively merging clusters. The decision of which documents are merged to a cluster is taken using a metric about the distance (e.g. Euclidean). Moreover, the distance between clusters is measured in several ways e.g. the maximum (or the minimum or the mean) distance between the documents of each cluster. In the *k-NN* algorithm, each document is assigned to the cluster which is most common among its $k$ nearest documents-neighbors (a distance metric is required, e.g. Euclidean Distance). We make use of the CLUTO-2.0 Clustering Toolkit (Karypis 2002) to generate results for these methods.

### 5.1 Datasets

Three well-known datasets were used for the evaluation of the proposed method: 20-NG, Reuters-21578 (both available from the UCI Machine Learning Repository[1]) and and the Brown Corpus (available from International Computer Archive of Modern and Medieval English[2]).

20 Newsgroups (20-NG) (Lang 1995) contains approximately 20,000 documents of the full 20-newsgroup collection of USENET news group articles. Each newsgroup belongs to a different category, with varying overlap between them: some newsgroups are very related (e.g. comp.os.ms-windows.misc and comp.windows.x) and others are not related at all (e.g. sci.electronics and alt.atheism).

Reuters-21578 (Carnegie Group Inc. and Reuters Ltd. 1997) is the most widely used dataset for text categorization and clustering purposes. The collection's documents appeared on the Reuters newswire in 1987 and were indexed with categories by several people. We chose to use articles that are uniquely assigned to exactly one topic (removing non-labeled data and documents without body) ending with approximately 10,000 documents with more than 100 categories.

Brown Corpus (Francis and Kucera 1964) contains 500 documents published in 1961 representing written American English. Each document has more than 2,000 words and the corpus covers a range of 15 genres (such as books on religion, skills and hobbies, academic texts e.t.c.).

### 5.2 Evaluation criteria

In order to evaluate the clustering result, we adopt three quality measure families widely used in text clustering bibliography (Amigó et al. 2009), the *F1 measures* (macro and micro), the *Purity/Inverse Purity*, the *Entropy* and the *counting pairs based measures* R, J and FM.

F1-measures, Purity and Inverse Purity combine the *Precision* and *Recall* from the Information Retrieval field. Whereas micro-averaging gives equal weight to every document, macro-averaging gives equal weight to each topic. The higher the values of these measures the better the clustering is. The definition of these measures involves the calculation of the F1 value between each system-generated cluster and each manually labeled topic. The precision, recall and F1-measure of a cluster $m$ with respect to a class $l$ are defined as:

$$\begin{aligned} P &= Precision(l, m) = \frac{N_{l,m}}{N_m} \\ R &= Recall(l, m) = \frac{N_{l,m}}{N_l} \\ F1(l, m) &= \frac{2 \cdot P \cdot R}{P + R} \end{aligned} \tag{23}$$

[1]See http://kdd.ics.uci.edu/.

[2]See http://icame.uib.no/.

where:

$F1(l, m)$ is the F1-measure for class $l$ with respect to cluster $m$, $N_{l,m}$ is the number of members of class $l$ in cluster $m$, $N_l$ is the number of members of class $l$ and $N_m$ is the number of members of cluster $m$.

Then, the macro- and micro-averaged F1 measures are calculated as follows:

$$Macro - F1 = \frac{1}{|L|} \sum_{l \in L} F1(l, \sigma(l))$$

$$Micro - F1 = 2 \cdot \frac{micro\,P \cdot micro\,R}{micro\,P + micro\,R} \tag{24}$$

where:

$L$ is the set of classes, $\sigma(l) = \arg\max_m(F1(l, m))$ and

$$micro\,P = \frac{1}{|L|} \sum_{l \in L} \frac{N_{l,\sigma(l)}}{N_{\sigma(l)}}$$

$$micro\,R = \frac{1}{|L|} \sum_{l \in L} \frac{N_{l,\sigma(l)}}{N_l}$$

Purity is based on the precision measure. Each resulting cluster $m$ from a cluster set $C$ of the overall document set $D$ is treated as if it were the result of a query. Each set $l$ of documents of a class set $L$ is treated as if it were the desired set of documents for a query. Purity and Inverse Purity are defined as:

$$Purity(C, L) = \sum_{m \in C} \left( \frac{|C|}{|L|} \max_{l \in L} Precision(l, m) \right)$$

$$InversePurity(C, L) = \sum_{l \in L} \left( \frac{|L|}{|C|} \max_{m \in C} Precision(l, m) \right) \tag{25}$$

Purity measures the purity of the resulting clusters when evaluated against a pre-categorization, while Inverse Purity measures how stable the pre-defined categories are when split up into clusters.

Entropy measures how "good" a cluster is in terms of homogeneity. The higher the homogeneity of a cluster, the lower the entropy is, and vice versa. For every cluster $m$ the probability $p_{l,m}$ that a member of cluster $m$ belongs to class $l$ is computed. The entropy is then calculated using the standard equation:

$$E_m = -\sum_l p_{l,m} \cdot log(p_{l,m}) \tag{26}$$

where the sum is taken over all classes $l$. The total entropy for the final cluster set $C$ is calculated as the sum of entropies of each cluster by taking into account the size of each cluster:

$$E_C = -\sum_{m \in C} \left( \frac{N_m}{N} \cdot E_m \right) \tag{27}$$

where:

$C$ is the set of final clusters,
$N_m$ is the number of documents in cluster $m$,
$N$ is the total number of documents.

Finally, measures based on pairs consider statistics over pairs of items. Let $SS$ denote the number of pairs of items belonging to the same cluster and category, $DD$ the number of pairs belonging to different clusters and categories, $SD$ the number of pairs belonging to the same cluster but different categories and $DS$ the number of pairs belonging to the same category but different clusters. $SS$ and $DD$ are desired examples while $SD$ and $DS$ are undesired. Three commonly used metrics based on these enumerations are the *Rand Statistic*, the *Jaccard Coefficient* and the *Folkes and Mallows metric*:

$$R = \frac{SS + DD}{SS + SD + DS + DD} \tag{28}$$

$$J = \frac{SS}{SS + SD + DS} \tag{29}$$

$$FM = \sqrt{\frac{SS}{SS + SD} \frac{SS}{SS + DS}} \tag{30}$$

On the other hand, it is desirable to evaluate the clustering results in terms of the proposed SOM variation quality. To achieve this, we adopt three evaluation measures popular in SOM bibliography (Pölzlbauer 2004): the *quantization error*, the *topographic product* and the *SOM distortion*.

The *Quantization Error* ($QE$) is traditionally related to all forms of vector quantization and clustering algorithms. Thus, this measure completely disregards map topology and alignment but evaluates the quality of clusters without the need of a dataset. $QE$ is computed by determining the average distance of the sample vectors to the cluster centroids by which they are represented. In the case of the SOM, the cluster centroids are the prototypes. Typically, the quantization error for each neuron $i$ is defined by the following equation:

$$QE_i = \frac{1}{M_i} \sum_{x_j \in i} ||m_i - x_j|| \tag{31}$$

where:

$i$ refers to neuron $i$,
$M_i$ refers to all the documents that are represented by neuron $i$,
$x_j$ refers to the document $j$ vector,
$m_i$ refers to the neuron $i$ vector

After the computation of $QE$ for each neuron of the SOM, the average of these errors reflects the $QE$ of the whole SOM.

The *Topographic Product* is one of the oldest measures that quantify the topology preservation properties of the Self-Organizing Map. The result of the computation of the Topographic Product indicates whether the size of the map is appropriate

to fit onto the dataset. For computation of the Topographic Product, only the map's codebook is regarded. The main idea of this measure is to compare the neighborhood relation between two neurons with respect to their reference vectors on the one hand ($Q_1(j,k)$) and according to its position in the map on the other hand ($Q_2(j,k)$):

$$Q_1(j,k) = \frac{dist(m_j, m_{n_k^A(j)})}{dist(m_j, m_{n_k^V(j)})} \tag{32}$$

$$Q_2(j,k) = \frac{dist(r_j, r_{n_k^A(j)})}{dist(r_j, r_{n_k^V(j)})} \tag{33}$$

where:

| | |
|---|---|
| $j$ | refers to neuron $j$, |
| $m_j$ | refers to neuron's $j$ reference vector, |
| $r_j$ | refers to neuron's $j$ position vector, |
| $n_k^A(j)$ | refers to the $k$-nearest neighbor to $j$ in the input space $V$, |
| $n_k^V(j)$ | refers to the $k$-nearest neighbor to $j$ in the map space $A$, |
| $dist$ | is a distance metric e.g. Euclidean. |

The parameter $k$ must be selected by the user and controls up to which rank the calculation is performed (e.g. for $k = 4$, the 4 nearest neighbors are investigated).

Combining (32) and (33) we obtain a measure $Q_3$ of the topological relationship between neuron $j$ and its $k$-nearest neighbors:

$$Q_3(j,k) = \left( \prod_{l=1}^{k} Q_1(j,l) \cdot Q_2(j,l) \right)^{\frac{1}{2k}} \tag{34}$$

Then, to extend this measure to every neuron in the SOM and to all possible neighborhood orders, we define the topographic product $P$ as:

$$P = \frac{1}{N(N-1)} \sum_{j=1}^{N} \sum_{k=1}^{N-1} log(Q_3(j,k)) \tag{35}$$

The value of $P$ can then be interpreted easily: If $P << 0$, the map is too small (i.e. has too few map nodes); for $P >> 0$, the map is too big for the data space it represents.

Finally, the *Distortion Measure* is used to evaluate the quality of the SOM. It has been shown that if the neighborhood's kernel radius is made constant, then there exists a cost function that the SOM optimizes, which is expressed through the Distortion Measure. This function can be used to compute an error value for the whole map. Also, one of the major advantages of the measure is that the error can be decomposed in many ways per unit, (such that the error can be localized on the map lattice) or per component (variable dimension). It is formally defined as:

$$DM = \sum_{j=1}^{M} \sum_{i=1}^{N} h_{b_j,i} ||m_i - x_j|| \tag{36}$$

where:

$M$ is the number of documents,
$N$ is the number of neurons,
$m_i$ refers to neuron $i$ vector,
$x_j$ refers to document $j$ vector,
$b_j$ is the winner neuron for document vector $x_j$,
$h_{b_j,i}$ is the neighborhood function as defined by (19).

### 5.3 Parameter selection

For each document of each dataset we follow the procedure described in Section 3 and represent it using Wikipedia knowledge. For each Wikipedia concept that we map, we use (11) to compute its weight in the document and keep a global hash with its *Keyphraseness* value.

We experimented with various values for the $\alpha$, $\beta$ and $\gamma$ parameters of (11) in order to define the effect of *WFreq*, *LinkRank*, *OrderRank* and *ConceptSim* on document representation. Since there is no easy way of knowing beforehand the value for each one of the three parameters for a particular dataset, the only way to determine them is to empirically experiment with respect to some performance measure (Guyon and Elisseeff 2003). At first, we set aside an independent test set. The remaining data is used both for training and performing parameter selection. In our experiments, we performed $k$-fold cross-validation (with $k$ set to 5) on each one of the 3 datasets described above. Documents were first partitioned into 5 (nearly) equally sized segments. Subsequently, 5 iterations of learning and validation were performed such that within each iteration a different fold of the data is held-out for validation, while the remaining 4 folds were used for learning. Documents were commonly stratified prior to being split into 5 folds, so as to ensure that each fold is a good representative of the whole dataset and that all available data are used for cross-validation. Lastly, the independent test set which was initially held out, is used to assess the overall model's performance, again following the 5-fold cross-validation protocol.

The optimal values for the parameters of (11) are those which produce the best clustering results in terms of F-measure and Entropy values (in the training and validation sets) and are shown in Table 3 for each one of the three datasets used. *LinkRank* has the biggest effect on document representation (confirming the dense and semantically rich structure of Wikipedia), whereas *OrderRank* has the smallest. *ConceptSim* appears to be more important than *WFreq* except for the Brown dataset

**Table 3** Optimal parameters

| Optimal values and deviations | | | | |
|---|---|---|---|---|
| Parameter | | 20-NG | Reuters | Brown |
| Wfreq | a | $0.2520 \pm 0.0101$ | $0.2320 \pm 0.0031$ | $0.256 \pm 0.0051$ |
| LinkRank | b | $0.3800 \pm 0.0025$ | $0.4120 \pm 0.0036$ | $0.3920 \pm 0.0033$ |
| OrderRank | c | $0.1000 \pm 0.0000$ | $0.1000 \pm 0.0000$ | $0.1000 \pm 0.0000$ |
| ConceptSim | 1-a-b-c | $0.2680 \pm 0.1114$ | $0.2560 \pm 0.0034$ | $0.2520 \pm 0.0026$ |
| MinFreq | | 0.001–0.01 (large sets), 0.01–0.03 (small sets) | | |
| MinKeyph | | 0.5 | | |

(where probably due to the large size of documents the classic $tf - idf$ schema is more accurate).

After completing the concept vector space document representation and the concept weight computation, we proceed with the main clustering process as described in Section 4.2. We select the initial neurons by setting the *minimum keyphraseness threshold* (MinKeyph) and the *minimum global frequency* threshold (MinFreq) to values aiming to create descriptive labels for the final clusters. This is achieved through a large enough *Keyphraseness* threshold (which wipes out many general concepts) and through a low *frequency* threshold (depending on the documents available and given the fact that concepts are generally not simple words). Experiments show that a value for MinKeyph around 0.5 always yields good results in different datasets, provided that there are at least a few hundreds of documents available.

Numerous experiments showed that, if a dataset contains less than 5,000 documents, MinFreq should be set between 0.01 and 0.03, otherwise MinFreq should be set between 0.001 and 0.01. All optimal parameters are shown in Table 3.

As described in Section 4.2, DoSO utilizes ISOMAP projection in order to visualize the performed clustering. ISOMAP uses parameter $k$ for defining the nearest neighbors in the created 2D graph. $k$ corresponds to the number of (Euclidean) neighbors that ISOMAP will use as connections to the neighborhood graph constructed, and in our experiments was set to a value between 4 (to represent a rectangular grid) and 6 (to represent a hexagonal grid). Please bear in mind that after initialization and projection, we end up with a neuron topology (just as with the classic SOM algorithm) and at this step $k$ is used as a parameter for determining this pseudo-grid type.

### 5.4 Clustering results

The cluster discovery function was described in Section 4.2. The final clustering results in comparison to those of HAC and k-NN (for both the BOW space and concept space), for the 20-NG, Reuters and Brown datasets are shown in Table 4. Firstly, the effect of concept space introduction into the typical clustering algorithms (HAC and k-NN) was examined by comparing the results of the trivial BOW model (BOW) with (a) the concept space model using only simple *tf-idf* equations (CS-TFIDF) and (b) the concept space model through concept weighting using (11) (CS-WEIGHTED). Finally, the proposed clustering method, DoSO, (which uses the weighted concept space model) was also tested. To clearly demonstrate the impact of concept space model introduction, we highlight the best value among all clustering algorithms with bold and the best value for each baseline algorithm (HAC and k-NN) with italics.

As presented in Table 4, the introduction of concept space into HAC and k-NN algorithms yields better results in terms of F1, R, J and FM measures, when using the CS-WEIGTHED model (and not the CS-TFIDF model), probably due to the fact that concept space is utilized better through concept weighting due to the availability of external knowledge provided from Wikipedia and not just by counting occurrences. However, it appears that the BOW model leads to lower entropy values (more homogeneous clusters) for 20-NG and Reuters datasets (whereas in Brown dataset DoSO yields lower entropy value). This is probably due to the fact that Brown dataset's documents are much longer than those of 20-NG or Reuters,

**Table 4** DoSO clustering evaluation (extrinsic measures)

| Measures | | | | Set matching based | | Counting pairs based | | | Entropy based |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | Algorithm | Representation model | Vector dimension | F1-macro | F1-micro | R | J | FM | E |
| 20-Newsgroup | HAC | BOW | 61174 | 0.542 | 0.558 | 0.87 | 0.70 | 0.74 | *0.205* |
| | | CS-TFIDF | 22510 | 0.486 | 0.514 | 0.91 | 0.75 | *0.78* | 0.391 |
| | | CS-WEIGHTED | 22510 | *0.557* | *0.569* | *0.92* | *0.78* | 0.76 | 0.216 |
| | k-NN | BOW | 61174 | 0.604 | 0.612 | 0.95 | 0.73 | 0.75 | **0.114** |
| | | CS-TFIDF | 22510 | 0.473 | 0.481 | 0.92 | 0.71 | *0.78* | 0.314 |
| | | CS-WEIGHTED | 22510 | *0.613* | *0.627* | *0.95* | *0.79* | 0.75 | 0.199 |
| | DoSO | CS-WEIGHTED | 22510 | **0.799** | **0.831** | **0.96** | **0.81** | **0.84** | **0.114** |
| Reuters | HAC | BOW | 18985 | 0.417 | 0.437 | 0.65 | 0.56 | 0.62 | *0.196* |
| | | CS-TFIDF | 10714 | 0.402 | 0.440 | 0.67 | 0.61 | 0.68 | 0.283 |
| | | CS-WEIGHTED | 10714 | *0.424* | *0.471* | *0.74* | *0.63* | *0.69* | 0.207 |
| | k-NN | BOW | 18985 | 0.392 | 0.426 | 0.72 | 0.65 | 0.69 | ***0.092*** |
| | | CS-TFIDF | 10714 | 0.377 | 0.412 | 0.62 | 0.50 | 0.57 | 0.236 |
| | | CS-WEIGHTED | 10714 | *0.441* | *0.485* | *0.77* | *0.69* | *0.73* | 0.187 |
| | DoSO | CS-WEIGHTED | 10714 | **0.792** | **0.828** | **0.92** | **0.83** | **0.83** | 0.119 |
| Brown | HAC | BOW | 59601 | 0.479 | 0.484 | 0.87 | *0.69* | 0.64 | 0.231 |
| | | CS-TFIDF | 17880 | 0.454 | 0.470 | 0.90 | 0.66 | *0.65* | 0.240 |
| | | CS-WEIGHTED | 17880 | *0.494* | *0.498* | *0.91* | 0.62 | 0.61 | *0.162* |
| | k-NN | BOW | 59601 | 0.461 | 0.476 | *0.88* | 0.65 | 0.62 | 0.191 |
| | | CS-TFIDF | 17880 | 0.443 | 0.467 | 0.84 | 0.61 | 0.66 | 0.238 |
| | | CS-WEIGHTED | 17880 | *0.472* | *0.487* | 0.87 | *0.67* | *0.69* | *0.123* |
| | DoSO | CS-WEIGHTED | 17880 | **0.784** | **0.845** | **0.92** | **0.78** | **0.71** | **0.112** |

which leads to more accurate representations of documents in concept space. In addition, the Entropy measure favors clustering algorithms that produce clusters with relatively uniform cluster size (such as *k-NN*) and penalize algorithms which produce clusters of widely different sizes (Xiong et al. 2004) (such as DoSO algorithm). Finally, it must be noted that the introduction of concept space leads to lower time complexity, due to the fact that concept space is much more compact than the simple vector space. The reduction in vector size is seen in Table 4. For the proposed clustering method and for the weighting of concepts (11) the parameters of Table 3 were used. For the HAC method, the UPGMA variant was implemented, while in k-NN method $k$ was set to 8 and the similarity threshold to 0.25.

The performance of DoSO is also compared to the classic SOM algorithm which is applied to the three datasets both using the BOW model and the concept space model. The results are presented in Table 5 (bold values represent the best results obtained for each dataset). The four different SOM approaches (simple tf-idf model, concept space tf-idf model, concept space weighted model and DoSO) were compared using different map sizes (small, medium or large). The results are more or less independent of the map size. Table 5 shows the average values of the measures over all map sizes. DoSO performs better in terms of Topographic Product measure which means (according to Section 5.2) that the size of map used (i.e. the number of neurons selected) is suitable for the data. The classic SOM algorithm (using the BOW model) performs better than DoSO in some cases in terms of Quantization Error and SOM Distortion measures but the differences are considered to be small especially taking into account the runtime improvement (training of SOM) introduced by DoSO (due to the weight initialization described in Section 4.2).

Finally, we compare our approach with the current state of the art algorithms in document clustering which make use of external knowledge, in terms of F-score, purity and inverse purity. The only dataset used in common and hence allowing comparison with other clustering methods was a subset of Reuters dataset containing 1658 documents from 30 classes (classes chosen have more than 15 and less than 200 documents). The results are presented in Table 6 (previous results collected from the paper of Kiran and Shankar (2010) and (as previously) bold values represent the best

**Table 5** DoSO clustering evaluation (SOM quality measures)

| Dataset | Algorithm | Representation model | QE | Topographic product | SOM distortion |
|---|---|---|---|---|---|
| 20-Newsgroup | SOM | BOW | **0.1221** | 0.0010 | **0.3932** |
| | | CS-TFIDF | 1.2264 | −0.0104 | 1.6130 |
| | | CS-WEIGHTED | 1.1073 | 0.0023 | 1.2543 |
| | DoSO | CS-WEIGHTED | 0.1435 | **0.0007** | 0.7319 |
| Reuters | SOM | BOW | 0.2932 | 0.0044 | 1.3692 |
| | | CS-TFIDF | 0.3339 | −0.0081 | 1.6423 |
| | | CS-WEIGHTED | 0.3431 | 0.0076 | 1.5219 |
| | DoSO | CS-WEIGHTED | **0.1134** | **0.0027** | **1.0244** |
| Brown | SOM | BOW | 0.3993 | 0.0082 | **0.1423** |
| | | CS-TFIDF | 1.2342 | −0.0092 | 7.3219 |
| | | CS-WEIGHTED | 2.0081 | 0.0040 | 1.2108 |
| | DoSO | CS-WEIGHTED | **0.1723** | **0.0025** | 0.8492 |

**Table 6** Comparison of DoSO with other document clustering approaches using Reuters subset

| | F-measure | Purity | Inv. purity | Knowledge source |
|---|---|---|---|---|
| BOW | 0.618 | 0.603 | 0.544 | – |
| Gabrilovich and Markovitch (2006) | – | 0.605 | 0.548 | Wikipedia |
| Hotho et al. (2003) | – | 0.607 | 0.556 | WordNet |
| Hu et al. (2009) | – | 0.655 | 0.598 | Wikipedia |
| Huang et al. (2009) | 0.575 | 0.678 | 0.75 | Wikipedia |
| Kiran and Shankar (2010) | 0.732 | 0.684 | 0.778 | Wikipedia, WordNet, dmoz, social bookmark |
| DoSO | **0.799** | **0.781** | **0.817** | Wikipedia |

results obtained). DoSO performs better that previous methods in terms of all three measures used.

## 6 Conclusion

In this paper, we utilized a new document representation model based on Wikipedia knowledge in order to build a self-organizing map based approach for hierarchical document clustering. The proposed representation model (concept space) exploits Wikipedia textual content, link and category structure in order to create a rich and compact document representation, thus clearly improves the performance of baseline clustering algorithms (which use the classic BOW model). A document clustering process based on the idea of self organizing maps is carried out by assigning labels to the neurons of the proposed model. The clustering results are visualized efficiently and a hierarchical approach is finally obtained, leading to the construction of a tree structure. Experiments on three datasets (20-NG, Reuters, Brown) show that the introduction of concept space model (in general) yields better results for all algorithms (compared to those of BOW model) and that the proposed technique (DoSO) is more accurate than other methods, while providing descriptive clusters.

We are currently investigating ways of improving the proposed clustering technique. The basic point of further research would be the introduction of multiple assignments of clusters to each document. This is based on the idea that documents of a corpus do not exhibit a specific field-topic but most of the times combine topics, so a model that assigns each document to a single topic (i.e. cluster) is not ideally suited to capture the full meaning of the document.

## References

Alias-i (2008). LingPipe 4.1.0 (online). http://alias-i.com/lingpipe. Accessed 23 Jan 2012

Amigó, E., Gonzalo, J., Artiles, J., & Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval, 12*, 461–486.

Banerjee, S., Ramanathan, K., & Gupta, A. (2007). Clustering short texts using Wikipedia. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 787–788). New York, NY, U.S.A.: ACM.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., et al. (2009). DBpedia—A crystallization point for the Web of data. *Journal Web Semantics, 7*(3), 154–165.

Bloehdorn, S., Cimiano, P., & Hotho, A. (2006). Learning ontologies to improve text clustering and classification. In M. Spiliopoulou, R. Kruse, A. Nürnberger, C. Borgelt, & W. Gaul (Eds.), *From*

*data and information analysis to knowledge engineering: Proceedings of the 29th annual conference of the German classification society (GfKl 2005), 9–11 Mar 2005, Magdeburg, Germany*. *Studies in classification, data analysis, and knowledge organization* (Vol. 30, pp. 334–341). Berlin-Heidelberg, Germany: Springer.

Breaux, T. D., & Reed, J. W. (2005). Using ontology in hierarchical information clustering. In *HICSS '05: Proceedings of the proceedings of the 38th annual Hawaii international conference on system sciences (HICSS'05)—track 4* (p. 111.2). Washington, DC, U.S.A.: IEEE Computer Society.

Bunescu, R. C., & Pasca, M. (2007). Using encyclopedic knowledge for named entity disambiguation. In *EACL*. The Association for Computer Linguistics.

A. Carnegie Group Inc., & B. Reuters Ltd. (1997). *Reuters-21578 text categorization test collection*.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks, 4*(6), 759–771.

Chen, H., Schuffels, C., & Orwig, R. (1996). Internet categorization and search: A self-organizing approach. *Journal of Visual Communication and Image Representation, 7*(1), 88–102.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on Wikipedia data. In *Proc. 2007 joint conference on EMNLP and CNLL* (pp. 708–716).

Davison, M. L. (1983). *Multidimensional scaling*. New York: Wiley.

Demartines, P., & Herault, J. (1997). Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks, 8*(1), 148–154.

Francis, W. N., & Kucera, H. (1964). *Manual of information to accompany a standard corpus of present-day edited American english, for use with digital computers*. Providence, Rhode Island.

Fung, B. C. M., Wang, K., & Ester, M. (2003). Hierarchical document clustering using frequent itemsets. In *Proc. of the 3rd SIAM international conference on data mining (SDM)* (pp. 59–70). San Francisco, CA: SIAM.

Gabrilovich, E., & Markovitch, S. (2006). Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI'06: Proceedings of the 21st national conference on artificial intelligence* (pp. 1301–1306). Menlo Park, CA: AAAI Press.

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJCAI'07: Proceedings of the 20th international joint conference on artifical intelligence* (pp. 1606–1611). San Francisco, CA, U.S.A.: Morgan Kaufmann Publishers Inc.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.

Hammouda, K. M., & Kamel, M. S. (2004). Efficient phrase-based document indexing for Web document clustering. *IEEE Transactions on Knowledge and Data Engineering, 16*, 1279–1296.

He, J., Tan, A.-h., & Tan, C.-l. (2002). ART-C: A neural architecture for self-organization under constraints. In *In proceedings of international joint conference on neural networks (IJCNN)* (pp. 2550–2555).

Himberg, J. (2000). A SOM based cluster visualization and its application for false coloring. In *IJCNN '00: Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks (IJCNN'00)* (Vol. 3, p. 3587). Washington, DC, U.S.A.: IEEE Computer Society.

Hofmann, T. (1999). The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data. In *In IJCAI* (pp. 682–687).

Hotho, A., Staab, S., & Stumme, G. (2003). Wordnet improves text document clustering. In Y. Ding, K. van Rijsbergen, I. Ounis, & J. Jose (Eds.), *Proceedings of the semantic Web workshop of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval (SIGIR 2003), 1 Aug 2003, Toronto Canada*.

Hotho, A., & Stumme, G. (2002). Conceptual clustering of text clusters. In *Proceedings of FGML workshop* (pp. 37–45). Special Interest Group of German Informatics Society (FGML).

Hu, J., Fang, L., Cao, Y., Zeng, H.-J., Li, H., Yang, Q., et al. (2008). Enhancing text clustering by leveraging Wikipedia semantics. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 179–186). New York, NY, U.S.A.: ACM.

Hu, X., Zhang, X., Lu, C., Park, E. K., & Zhou, X. (2009). Exploiting Wikipedia as external knowledge for document clustering. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 389–396). New York, NY, U.S.A.: ACM.

Huang, A., Milne, D., Frank, E., & Witten, I. H. (2009). Clustering documents using a Wikipedia-based concept representation. In *Proceedings of the 13th Pacific–Asia Conference on advances in knowledge discovery and data mining. PAKDD '09* (pp. 628–636). Berlin, Heidelberg: Springer.

Jin, H., Wong, M.-L., & Leung, K. S. (2005). Scalable model-based clustering for large databases based on data summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(11), 1710–1719.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*, 11–21.

Junker, M., Sintek, M., & Rinck, M. (2000). Learning for text categorization and information extraction with ILP. *Learning Language in Logic*, 247–258.

Kangas, J., Kohonen, T., & Laaksonen, J. (1990). Variants of self-organizing maps. *IEEE Transactions on Neural Networks, 1*(1), 93–99.

Karypis, G. (2002). CLUTO—A clustering toolkit (Vol. 02–017). Technical Report.

Kiran, G. V. R., & Shankar, R. (2010). Enhancing document clustering using various external knowledge sources. In *Proceedings of the 15th Australasian document computing symposium*.

Kohonen, T. (1989). *Self-organization and associative memory* (3rd Edn.). New York, NY, U.S.A.: Springer New York, Inc.

Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., et al. (2000). Self organization of a massive document collection. *IEEE Transactions on Neural Networks, 11*(3), 574–585.

Kohonen, T., Schroeder, M. R., & Huang, T. S. (Eds.) (2001). *Self-organizing maps*. Secaucus, NJ, U.S.A.: Springer New York, Inc.

Kraaijveld, M. (1992). A non-linear projection method based on Kohonen's topology preserving maps. In *11th IAPR international conference on pattern recognition, 1992. Conference B: Pattern recognition methodology and systems, proceedings* (Vol. II, pp. 41 –45).

Lagus, K., Kaski, S., & Kohonen, T. (2004). Mining massive document collections by the WEBSOM method. *Informing Science, 163*(1–3), 135–156.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Proceedings of the international conference on machine learning*. Tahoe City, California, U.S.A.: Morgan Kaufmann.

Larsen, B., & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 16–22). New York, NY, U.S.A.: ACM.

Li, Y., Luk, W. P. R., Ho, K. S. E., & Chung, F. L. K. (2007). Improving weak ad-hoc queries using Wikipedia as external corpus. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 797–798). New York, NY, U.S.A.: ACM.

Lin, X., Soergel, D., & Marchionini, G. (1991). A self-organizing semantic map for information retrieval. In *SIGIR '91: Proceedings of the 14th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 262–269). New York, NY, U.S.A.: ACM.

Liu, X., Gong, Y., Xu, W., & Zhu, S. (2002). Document clustering with cluster refinement and model selection capabilities. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 191–198). New York, NY, U.S.A.: ACM.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics, 19*(2), 313–330.

Mendes, P., Jakob, M., Garca-Silva, A., & Bizer, C. (2011). Dbpedia spotlight: Shedding light on the Web of documents. In *In the proceedings of the 7th international conference on semantic systems (I-semantics)*.

Merkl, D. (1998). Text classification with self-organizing maps: Some lessons learned. *Neurocomputing, 21*(1–3), 61–77.

Merkl, D., & Rauber, A. (1997). Alternative ways for cluster visualization in self-organizing maps. In *In Proc. of the workshop on self-organizing maps (WSOM97)* (pp. 106–111).

Mihalcea, R., & Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. In *CIKM '07: Proceedings of the sixteenth ACM conference on conference on information and knowledge management* (pp. 233–242). New York, NY, U.S.A.: ACM.

Miikkulainen, R. (1990). Script recognition with hierarchical feature maps. *Connection Science, 2*, 83–101.

Milne, D., & Witten, I. H. (2008). Learning to link with Wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management. CIKM '08* (pp 509–518). New York, NY, U.S.A.: ACM.

Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(3), 301–312.

Moutarde, F., & Ultsch, A. (2005). U*F clustering: A new performant "cluster-mining" method based on segmentation of self-organizing maps. In *Workshop on self-organizing maps (WSOM'2005)*.

Navigli, R., & Ponzetto, S. P. (2010). Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics. ACL '10* (pp. 216–225). Stroudsburg, PA, U.S.A.: Association for Computational Linguistics.

Pampalk, E., Rauber, A., & Merkl, D. (2002). Using smoothed data histograms for cluster visualization in self-organizing maps. In *ICANN '02: Proceedings of the international conference on artificial neural networks* (pp. 871–876). London, U.K.: Springer.

Pölzlbauer, G. (2004). Survey and comparison of quality measures for self-organizing maps. In J. Paralič, G. Pölzlbauer, & A. Rauber (Eds.), *Proceedings of the fifth workshop on data analysis (WDA'04), Sliezsky dom, Vysoké Tatry, 24–27 June 2004* (pp. 67–82). Slovakia: Elfa Academic Press.

Pullwitt, D. (2002). Integrating contextual information to enhance som-based text document clustering. *Neural Networks, 15*(8–9), 1099–1106.

Ratinov, L., Roth, D., Downey, D., & Anderson, M. (2011). Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (Vol. 1, pp. 1375–1384). HLT '11. Stroudsburg, PA, U.S.A.: Association for Computational Linguistics.

Rauber, A. (1999). LabelSOM: On the labeling of self-organizing maps. In *International joint conference on neural networks, 1999. IJCNN '99* (Vol. 5, pp. 3527–3532).

Rauber, A., Merkl, D., & Dittenbach, M. (2002). The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks, 13*, 1331–1341.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*(5500), 2323–2326.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York, U.S.A.: McGraw-Hill.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers, 18*(5), 401–409.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing, Manchester, UK*.

Sedding, J., & Kazakov, D. (2004). Wordnet-based text document clustering. In *ROMAND '04: Proceedings of the 3rd workshop on robust methods in analysis of natural language data* (pp. 104–113). Morristown, NJ, U.S.A.: Association for Computational Linguistics.

Shehata, S., Karray, F., & Kamel, M. S. (2010). An efficient concept-based mining model for enhancing text clustering. *IEEE Transactions on Knowledge and Data Engineering, 22*, 1360–1371.

Slonim, N., Friedman, N., & Tishby, N. (2002). Unsupervised document classification using sequential information maximization. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 129–136). New York, NY, U.S.A.: ACM.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning, 34*(1–3), 233–272.

Spanakis, G., Siolas, G., & Stafylopatis, A. (2011). Exploiting Wikipedia knowledge for conceptual hierarchical clustering of documents. *The Computer Journal, Section C: Computational Intelligence*. doi:10.1093/comjnl/bxr024.

Stanford (2009). Named entity recognizer (online). http://www-nlp.stanford.edu/software/CRF-NER.shtml. Accessed 23 Jan 2012

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. In M. Grobelnik, D. Mladenic, & N. Milic-Frayling (Eds.), *KDD-2000 workshop on text mining, Boston, MA* (pp. 109–111).

Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., & Lakhal, L. (2002). Computing iceberg concept lattices with TITANIC. *Data & Knowledge Engineering, 42*(2), 189–222.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2008). Yago: A large ontology from Wikipedia and Wordnet. *Journal Web Semantics, 6*, 203–217.

Talavera, L., & Bejar, J. (2001). Generality-based conceptual clustering with probabilistic concepts. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(2), 196–206.

Tenenbaum, J. B., Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290*(5500), 2319–2323.

Toral, A., & Munoz, R. (2006). A proposal to automatically build and maintain gazetteers for named entity recognition by using Wikipedia. In *EACL*. The Association for Computer Linguistics.

Ultsch, A., & Siemon, H. P. (1990). Kohonen's self organizing feature maps for exploratory data analysis. In *Proceedings of international neural networks conference (INNC)* (pp. 305–308). Kluwer Academic Press.

Vesanto, J., & Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks, 11*(3), 586–600.

Vinokourov, A., & Girolami, M. (2002). A probabilistic framework for the hierarchic organisation and classification of document collections. *Journal of Intelligent Information Systems, 18*, 153–172.

Wang, P., & Domeniconi, C. (2008). Building semantic kernels for text classification using Wikipedia. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 713–721). New York, NY, U.S.A.: ACM.

Wang, P., Hu, J., Zeng, H.-J., & Chen, Z. (2009). Using Wikipedia knowledge to improve text classification. *Knowledge and Information Systems, 19*(3), 265–281.

Wang, B. B., Mckay, R. I. B., Abbass, H. A., & Barlow, M. (2003). A comparative study for domain ontology guided feature extraction. In *ACSC '03: Proceedings of the 26th Australasian computer science conference* (pp. 69–78). Darlinghurst, Australia, Australia: Australian Computer Society, Inc.

Wikipedia (2011). Wikipedia API (online). http://en.Wikipedia.org/w/api.php. Accessed 18 Oct 2011

Willett, P. (1988). Recent trends in hierarchic document clustering: A critical review. *Information Processing & Management, 24*(5), 577–597.

Xiong, H., Steinbach, M., Tan, P., & Kumar, V. (2004). HICAP: Hierarchical clustering with pattern preservation. In *Proceedings of SIAM international conference on data mining* (pp. 279–290). Philadelphia, PA: SIAM.

Xue, X.-B., & Zhou, Z.-H. (2009). Distributional features for text categorization. *IEEE Transactions on Knowledge and Data Engineering, 21*(3), 428–442.

Yin, H. (2002). ViSOM—A novel method for multivariate data projection and structure visualization. *IEEE Transactions on Neural Networks, 13*(1), 237–243.