Knowledge-Based Systems xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

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

A clustering technique for news articles using WordNet ☆

Christos Bouras *, Vassilis Tsogkas [1]

Computer Technology Institute and Press "Diophantus", Patras, Greece
Computer Engineering and Informatics Department, University of Patras, 26500, Rion, Patras, Greece

ARTICLE INFO

Article history:
Received 6 January 2012
Received in revised form 25 June 2012
Accepted 26 June 2012
Available online xxxx

Keywords:
News clustering
k-Means
W-k means
Cluster labeling
Partitional clustering

ABSTRACT

The Web is overcrowded with news articles, an overwhelming information source both with its amount and diversity. Document clustering is a powerful technique that has been widely used for organizing data into smaller and manageable information kernels. Several approaches have been proposed which, however, suffer from problems like synonymy, ambiguity and lack of a descriptive content marking of the generated clusters. In this work, we are investigating the application of a great spectrum of clustering algorithms, as well as similarity measures, to news articles that originate from the Web. Also, we are proposing the enhancement of standard k-means algorithm using the external knowledge from WordNet hypernyms in a twofold manner: enriching the "bag of words" used prior to the clustering process and assisting the label generation procedure following it. Furthermore, we are examining the effect that text preprocessing has on clustering. Operating on a corpus of news articles derived from major news portals, our comparison of the existing clustering methodologies revealed that k-means, gives better aggregate results when it comes to efficiency. This is amplified when the algorithm is accompanied with preliminary steps for data cleaning and normalizing, despite its simple nature. Moreover, the proposed WordNet-enabled W-k means clustering algorithm significantly improves standard k-means generating also useful and high quality cluster tags by using the presented cluster labeling process.

1. Introduction

News articles flood the Web every day from an extreme amount of major or minor news portals from around the globe. It is utterly impossible for a single individual to be able to keep track of an event, or a series of related events, from an unbiased and truly informative point of view. While the amount of online information sources is rapidly increasing, so does the available online news content. One of the most common approaches for organizing this immense amount of data is the use of clustering techniques. Object clustering refers to the process of partitioning a collection of objects into several sub-collections based on their similarity of contents. For the case of user clustering, each sub-collection is called a user cluster and includes users that have revealed similar appeals in their selections of text articles while browsing through a document collection. Clustering has been proven to be a useful technique for information retrieval by discovering interesting information kernels and distributions in the underlying data. In general, it helps constructing meaningful partitions of large sets of objects based on various methodologies and heuristics. It plays a crucial role in organizing large collections. For example (a) it can be used to structure query results, (b) form the basis for further processing of the organized topical groups using other information retrieval techniques such as summarization, or (c) within the scope of recommendation systems by affecting their performance as far as suggestions made towards the end users are concerned. Clustering has also been exploited within the scope of machine learning [2], as a time series mining task [17] which uses frequent itemsets to find association rules of items in large transactional databases.

Clustering of news articles can help by depicting the underneath content hierarchy of a huge amount of articles within the reach of a single individual. Consequently, it can provide information retrieval (IR) systems with the potential to alleviate users while browsing and detecting quickly the needed information.

However, there are several challenges that clustering techniques normally have to overcome. Among them is efficiency: generated clusters have to be well connected from a notional point of view, despite the diversity in content and size that the original documents might have. For example, it is frequent for some news articles to belong to the same notional cluster, even though they do not share common words. The vice-versa is also possible: news articles sharing common words, while being completely unrelated to each other. Ambiguity and synonymy are thus two of the major

☆ This manuscript is an extended version of the KES2010 conference paper named: "W-k means: Clustering News Articles using WordNet".

* Corresponding author at: Computer Technology Institute and Press "Diophantus", Patras, Greece. Tel.: +30 2610 960375; fax: +30 2610 969016.

E-mail addresses: bouras@cti.gr (C. Bouras), tsogkas@ceid.upatras.gr (V. Tsogkas).

[1] Tel.: +30 2610 996954.

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2012.06.015

problems that document clustering techniques regularly fail to tackle with. Furthermore, having IR systems simply generate clusters of documents is not enough per se. The reason is that it is virtually impossible for humans to conceptualize information by merely browsing through hundreds of documents belonging to the same cluster. However, assigning meaningful labels to the generated clusters can help users conveniently recognize the content of each generated set and thus easily analyze the results.

In this manuscript, we are describing a variety of document clustering techniques and evaluating their application on our data set: news articles originating from the Web. Our aim is to compare the resulting clusters and determine which technique is best fitted for the extreme amount and diversity of news articles that an indexing system needs to address. Furthermore we are presenting a novel methodological approach towards document clustering, and in particular, clustering of news articles deriving from the Web, that combines regular k-means with external information extracted from the WordNet database. Our approach combines keyword extraction and several information retrieval techniques. We are also incorporating the proposed algorithm in our existing system [5], evaluating the clustering results compared to regular k-means using a large pool of Web news articles existing in the system's database.

The rest of the manuscript is organized as follows: Section 2 gives a background of the related work regarding clustering methodologies as well as the use of the WordNet database on this field. In Section 3, we give a brief overview of our system which we are enhancing with clustering techniques. In Section 4 we describe the various clustering methodologies explored in this work, while in Section 5 we present the algorithmic approach of W-k means. In Section 6 we outline our experimental approach towards the clustering methodologies used and present our evaluation results. Section 7 concludes this manuscript with some remarks about the future work that is currently underway.

2. Related work

Clustering data in general has been heavily researched by the scientific community over the last 20 years. Especially for document clustering, a huge variety of techniques has been proposed. A major goal of document clustering is to improve the results of information retrieval systems in terms of precision/recall. This in turn leads to serving better filtered and adequate results to their users, helping in essence the decision making process.

2.1. Clustering methodologies

Two generic categories of the various clustering methods exist: agglomerative hierarchical and partitional. Typical hierarchical techniques [11] generate a series of partitions over the data, which may run from a single cluster containing all objects, to n clusters each containing a single object, and are widely visualized through a divisive (root to leaves) or agglomerative (leaves to root) tree structure. On the other hand, partitional algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering. For partitional techniques, a global criterion in most commonly used, the optimization of which drives the entire process producing thus a single-level division of the data. Given the number of desired clusters, let k, partitional algorithms find all k clusters of the data at once, such that the sum of distances over the items to their cluster centers is minimal. Moreover, for a clustering result to be accurate, besides the low intra-cluster distance, high inter-cluster distances, i.e. well separated clusters, is desired. A typical partitional algorithm is k-means which is based on the notion of the cluster center, a point in the data space, usually not existent in the data themselves, which represents a cluster.

We will now briefly elaborate more on the techniques that are applied within the scope of this paper to our clustering experimental approach.

2.1.1. Hierarchical clustering

Divisive hierarchical methodologies generate a nested sequence of partitions, with a single, all-inclusive cluster at the top and singleton clusters of individual points at the bottom [20]. The vice-versa procedure occurs with agglomerative methodologies: the algorithm starts by considering each data point as a cluster of its own and proceeds by merging together tree nodes that share a certain degree of similarity.

In the above sense, hierarchical techniques require a cluster similarity or distance measure, in order to successively split clusters or merge data points belonging to different clusters. Most commonly, a similarity (distance) matrix is computed whose ij_{th} element expresses the distance between the i_{th} and j_{th} cluster. This matrix is updated on each step, where subsequent nodes are created by pairwise joining (for agglomerative) or splitting (for divisive) of nodes until the process is complete. The result of the above techniques is a tree-like structure, a dendrogram displaying the merging process. The intermediate clusters that occur during the procedure can be taken by "cutting" the tree at the required precision level. The aforementioned procedure is deterministic, compared with the ones described in the following subsection for partitional techniques. However, as explained by Day and Edelsbrunner [10], sequential agglomerative hierarchical non-overlapping (SAHN) clustering methods, feature an average complexity of at least $O(n^2)$ and most commonly $O(n^3)$ – on the input size n – which in many cases is aversive for use with large datasets.

There are several flavors of hierarchical clustering techniques that we are evaluating in this manuscript. Their difference lies in how the distance between clusters is defined in terms of their members – articles. Typically, pairwise single, maximum, average, and centroid linkage distances between clusters are considered. For pairwise single linkage, the shortest among the pairwise distances of the clusters is considered as the inter-cluster distance, whereas for pairwise maximum linkage this is the longest among them. Moreover, for pairwise average linkage the mean of the pairwise distances is defined as the inter-cluster similarity (i.e. distance). Finally, for the centroid linkage, each cluster is represented by its centroid which is calculated on each step of the algorithm and the inter-cluster distance is the distance between the cluster centers.

2.1.2. Partitional clustering

Contrary to hierarchical clustering, partitional techniques produce a single-level division of the data. Given the number of desired clusters, let k, partitional algorithms find all k clusters of the data at once, such that the sum of distances over the items to their cluster centers is minimal. In addition, for a clustering result to be accurate, besides the low intra-cluster distance, high inter-cluster distances, i.e. well separated clusters, is desired. Typical partitional algorithms are: k-means, k-medians, and k-medoids. These algorithms are based on the notion of the cluster center, a point in the data space, usually not existent in the data themselves, which represents a cluster. Their difference consists in how the cluster center is defined in each case. Following, we will briefly describe each approach as well as some of their variations in the literature.

In k-means clustering, the cluster center is defined as the mean data vector averaged over all items in the cluster. In k-medians, instead of the mean, the median is calculated for each dimension in the data vector. Finally, in k-medoids the cluster center is defined as the item which has the smallest sum of distances to the other items in the cluster. k-Medoids has the advantage of better handling of the outliers existing in data, while it does not depend on the order in which the objects are examined. The family of k-means partitional clustering algorithms [26] usually tries to minimize the average squared distance between points in the same cluster, i.e. if $d_1, d_2, \ldots, d_n$ are the n documents and $c_1, c_2, \ldots, c_k$ are the k clusters centroids, k-means tries to minimize the global criterion function:

$$\sum_{\imath=1}^{k}\sum_{j=1}^{n} sim(d_j, c_i) \tag{1}$$

Typically, all those algorithms share the following Expectation Maximization (EM) steps [3]:

```
Algorithm 1: Basic k-means EM algorithm
1. Randomly Select $K$ points as the initial centroids
2. Assign all data to the closest centroid
3. Calculate the new centroids for each cluster
4. Repeat steps 2 and 3 until no reassignments of the centroids
   takes place.
```

The EM algorithm suffers from frequently converging to local minima (or maxima), due to the random choice of the initial centroids. Computing thus a refined starting condition can yield significant improvements [7]. For example k-means++ [4], selects a point x as an initial cluster center, using a probability that is proportional to the square of the distance between each successive choice and the previous ones and then proceeds as k-means. This heuristic offers a significant boost compared with regular k-means as far as error and execution time are concerned. Another approach commonly used is multiple executions of the k-means algorithm, with different starting conditions, and finally keeping the best result; if a specific cluster assignment appears to be repeating, it is likely to be the best.

Bisecting k-means [25] introduces an alternative approach: initially the whole data set is treated as one cluster. A cluster is selected for split into two at each step by using a criterion such as the cluster size or the overall similarity. The split of the selected cluster is done using regular k-means and the procedure completes when the desired number of clusters is created. Consequently, unlike regular k-means, which splits the whole data set into k cluster at each iteration step, its bisecting variation splits only one existing cluster into two sub-clusters. The selection of which cluster to split can be based on its size, or on the centroid's neighbors network. Surprisingly, bisecting k-means is reported with a performance that generally beats k-means and even hierarchical approaches, while keeping the complexity linear.

The low complexity is commonplace for all of the previously mentioned partitional algorithms and thus they are best suited for clustering large document databases, as it is the case of this paper. Especially for Algorithm 1, the average complexity is linear in all relevant factors: iterations, number of clusters and number of documents [3].

Many of the above methodologies have been bundled in software clustering packages, like Cluto [14] and SenseClusters [16]. Cluto provides three different classes of clustering algorithms that operate either directly in the object's feature space or in the object's similarity space. A key feature in most of Cluto's clustering algorithms is that they treat the clustering problem as an optimization process which seeks to maximize or minimize a particular clustering criterion function defined either globally or locally over the entire clustering solution space. Cluto has two execution modes, one that treats each object as a vector in a high-dimensional space and one that operates on the similarity space between the objects. Both of these modes compute the clustering solution using one of five different approaches: four of these approaches are partitional in nature, whereas the fifth approach is agglomerative. SenseClusters is a word sense discrimination system that takes a purely unsupervised clustering approach. It creates clusters made up of the contexts in which a given target word occurs. It uses no knowledge other than what is available in a raw unstructured corpus, and clusters instances of a given target word based only on their mutual contextual similarities. We are utilizing both of the above toolboxes at our evaluation stage.

2.2. WordNet

WordNet is one of the most widely used and largest lexical databases of English. It attempts to model the lexical knowledge of a native English speaker. Containing over 150,000 terms, it groups nouns, verbs, adjectives and adverbs into sets of synonyms called synsets. The synsets are organized into senses, giving thus the synonyms of each word, and also into hyponym/hypernym (i.e. Is-A), and meronym/holonym (i.e. Part-Of) relationships, providing a hierarchical tree-like structure for each term. The applications of WordNet to various IR techniques have been widely researched concerning finding the semantic similarity of retrieved terms [24], or their association with clustering techniques. For example in Chen et al. [9], they combine the WordNet knowledge with fuzzy association rules and in Sedding and Kazakov [19], they extend the bisecting k-means using WordNet; however, due to the fact that they choose WordNet hypernyms/synonyms in 'levels', they come to the conclusion that noise is degrading their clustering results. Compared to their approach, we believe that a valid weighing scheme for the WordNet hypernyms can prevent this problem. In [12] the authors explore the use of WordNet as a disambiguation tool by assigning the stemmed keywords to their lexical category. Their approach improves the efficiency of the applied clustering algorithms; however, it seems to overgeneralize the affected keywords. This is also the case for the study of Abdelmalek et al. [1], where the authors accept that the assignment of terms to concepts in ontology can be ambiguous and can lead to loss of information in their attempt to reduce dimensionality. Both of the aforementioned approaches do not take into consideration the WordNet hypernyms to actually enrich the list of keywords as we propose in this manuscript. Kiran et al. [15], propose a hierarchical clustering algorithm using closed frequent itemsets that use Wikipedia as an external knowledge to enhance the document representation.

Regarding cluster labeling, techniques frequently evaluate labels using information from the cluster themselves [22], while existing approaches that utilize other external databases, like Wikipedia [8] are only good for the labeling process and not the clustering one. Recently in [23], the authors propose an effective Fuzzy Frequent Itemset-based Document Clustering approach that combines fuzzy association rule mining with the background knowledge embedded in WordNet hypernyms for generating cluster labels; however, as the authors suggest, fuzzy association mining and the initial clustering stages are the two most time-consuming tasks, something that leads to high execution times in order to get the required cluster labels (even though it scales linearly as the amount of documents increases). In contrast, we are focusing on an approach that will generate the clusters as well as their labels reasonably fast.

3. Information flow

Our system, PeRSSonal [5], features a staged and modular approach for performing the various tasks concerning news articles

Please cite this article in press as: C. Bouras, V. Tsogkas, A clustering technique for news articles using WordNet, Knowl. Based Syst. (2012), http://dx.doi.org/10.1016/j.knosys.2012.06.015

that originate from the Web. The scope of the PeRSSonal system is the construction of a new generation Web service that unifies many Information Retrieval tasks under a common framework. It is delivering quality information, targeted to end users that do not want or do not have the time to engage to the tedious task of filtering information. PeRSSonal consists of several autonomous sub-modules, each one for a specific IR task. The flow of information as handled within our system is depicted in Fig. 1.

At its input stage, our system crawls and fetches news articles from major or minor news portals from around the world. This is an offline procedure and once articles as well as metadata information are fetched, they are stored in the centralized database from where they are picked up by the following procedures.

A key procedure of the system as a whole, which is probably as least as important as the clustering algorithm that follows it, is text preprocessing on the fetched article's content, that results into the extraction of the keywords each article consists of. Analyzed in [5], keyword extraction handles the cleaning of articles, the extraction of the nouns [6], the stemming as well as the stopword removal process. Following, it applies several heuristics to come up with a weighing scheme that appropriately weighs the keywords of each article based on information about the rest of the documents in our database. This weighting scheme takes into consideration: (a) the existence of a keyword in the title, (b) the frequency of a keyword in the article's body, (c) the noun tagging information, and (d) the existence of a keyword in the article's summary.

Next comes the pruning of words that appear with low frequency throughout the corpus and are unlikely to appear in more than a small number of articles. Keyword extraction in essence generates the term-frequency vector [18] for each article that is used by the information retrieval techniques that follow treating it as a 'bag of words' (words – frequencies).

Text summarization, categorization of the articles on a predetermined set of classes, as well as personalization of the results, are some additional steps deployed in order to extract useful information from the data [5]. It is this level of the system that we are enhancing in this paper with the application of document clustering algorithms, in order to generate better results that the system's users view. Following the retrieval techniques, information is transmitted back to the end user.

4. Clustering news articles

The overall clustering process as evaluated in this paper is depicted in Fig. 2.

The generated term – frequency vectors ('bag of words') for each article described in the previous section, which is a weighted scheme of stemmed nouns existing in the original text, is given as input to the clustering subsystem. At this level, we used a twofold implementation/evaluation. Firstly, by applying a variety of clustering algorithms and distance metrics, we try to determine whether preprocessing has an effect on the domain of clustering news articles and which approach yields the best results. Most importantly, we try to estimate the effect of noun identification and stemming on each clustering approach. Secondly, our aim towards increasing the efficiency of the used clustering algorithm is to enhance this 'bag of words' with the use of external databases, and in particular, WordNet (dashed box). This enhanced feature list feeds the k-means clustering procedure that follows, leading to our clustering implementation (W-k means). The generated clusters are finally forwarded for labeling, taking also advantage of the WordNet database. The labeling subprocess outputs suggested tags for the given cluster. Cluster assignments and labels are the output of the proposed approach.

An important aspect that has to do with news articles in general is their diversity and similarity at the same time. When fetching information from numerous news portals, it is normal to expect a certain degree of similarity, as far as the content is concerned, since a great amount of the published news articles is copied from other sources. However, it is important to be able to understand minor differences which may usually betray biases to certain opinions expressed in the articles. Moreover, when dealing with

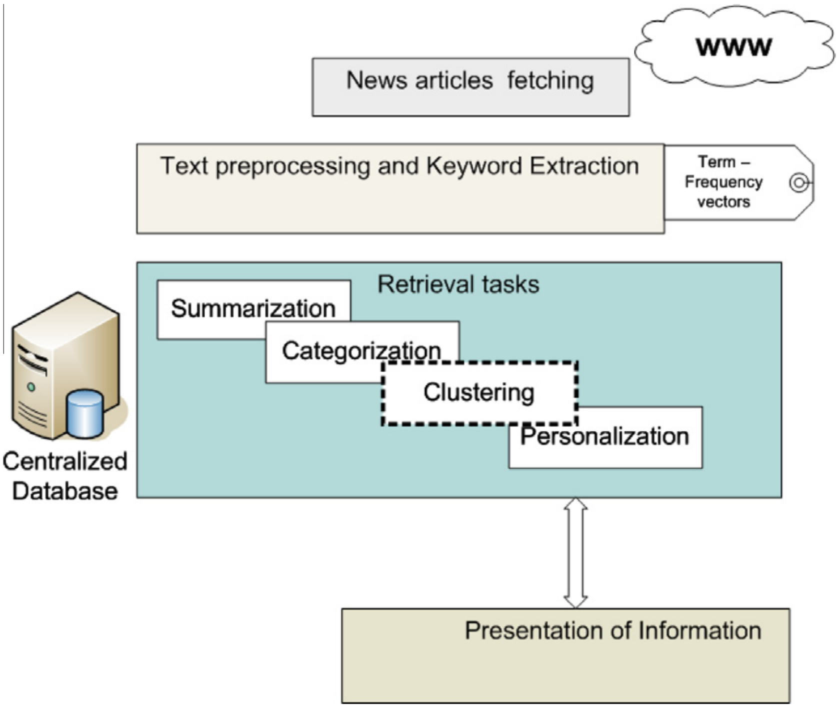

Fig. 1. Overview of the PeRSSonal's architecture.

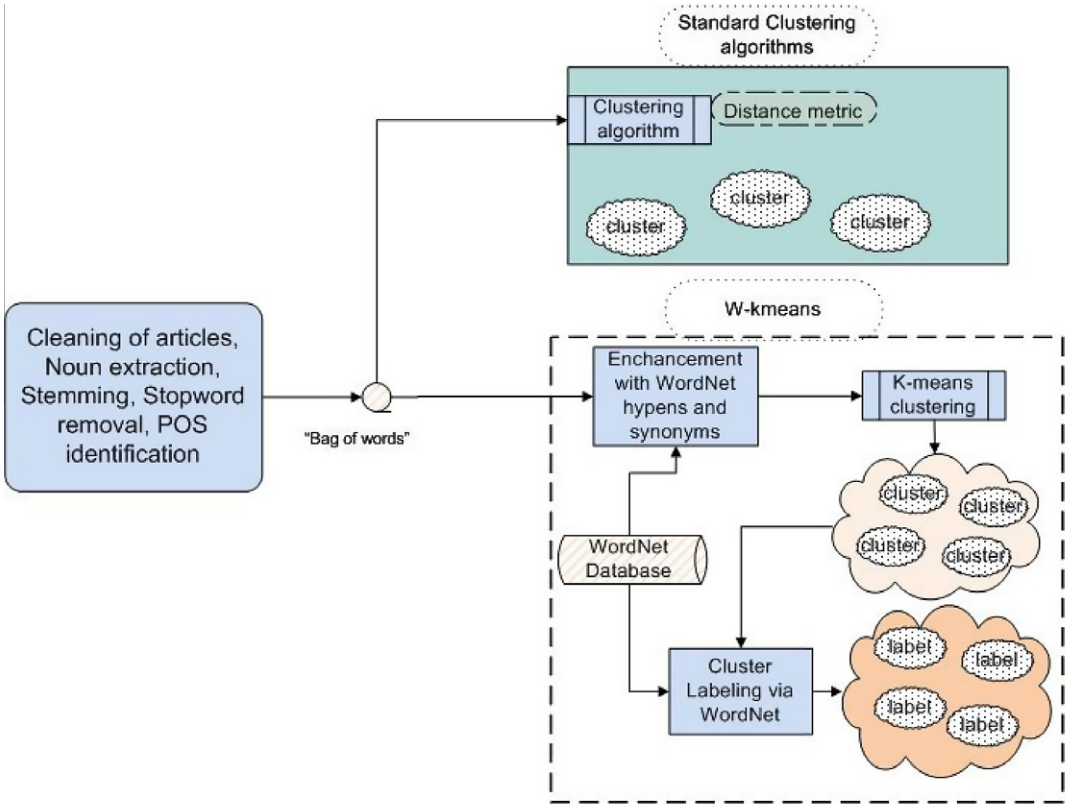
Standard Clustering algorithms
Clustering algorithm
Distance metric
cluster
cluster
cluster
W-kmeans
Cleaning of articles, Noun extraction, Stemming, Stopword removal, POS identification
"Bag of words"
Enchancement with WordNet hypens and synonyms
K-means clustering
cluster
cluster
cluster
cluster
WordNet Database
Cluster Labeling via WordNet
label
label
label
label

Fig. 2. Flow of information for evaluating the clustering methodologies.

documents, the amount of terms that the system can possibly come across is limitless (even more when multiple languages are taken into consideration), compared for example with gene-clustering. The applied algorithms, as well as the similarity measures used should take into consideration the above. Following, we describe the various similarity measures that are applied on high dimensionality sparse data within the scope of this paper.

All clustering methodologies described in Section 2 need to embed the documents to a suitable similarity space, thus share the notion of establishing the distance, i.e. similarity, between two data points, two clusters, or a data point and a cluster. In this paper, we are using the following distance functions for comparing the various methodologies:

- Euclidian, where the distance between two data points a and b is defined as:

$$d(a,b) = \frac{1}{n}\sum_{i=1}^{n}(a_i - b_i)^2 \tag{2}$$

n being the dimensionality of the data. The Euclidean distance takes the magnitude of the input data into account and consequently preserves more information about them.

- City-block:

$$d(a,b) = \frac{1}{n}\sum_{i=1}^{n}|a_i - b_i| \tag{3}$$

- Pearson correlation coefficient:

$$r(a,b) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{a_i - \bar{a}}{\sigma_\alpha}\right)\left(\frac{b_i - \bar{b}}{\sigma_b}\right) \tag{4}$$

in which $\bar{a}$ and $\bar{b}$ are the sample mean of a and b respectively, and σ_α, σ_b are the sample standard deviation of a and b. The Pearson correlation coefficient can be thought as a measure for how well a straight line can be fitted to a scatterplot of a and b. The Pearson correlation coefficient is either +1 or −1 for the points in the scatterplot that lie on a straight line. Note that the Pearson distance is thus defined as:

$$d(a,b) = 1 - r \tag{5}$$

- Cosine similarity:

$$d(a,b) = \cos(\theta) = \frac{a \bullet b}{|a||b|} \tag{6}$$

where the similarity between the two data points is viewed by means of their angle in the n-dimensional space.

- Spearman-rank correlation ρ,

which is a non-parametric measure that performs well against outliers. It originates from the Pearson correlation by replacing every data value with its rank having the values firstly ordered. Due to the diminishing of the data values, there is no weight information taking place to the distance calculation compared to the previous – parametric similarity measures. The Spearman-rank distance between two data points a and b is defined as:

$$d(a,b) = 1 - \rho \tag{7}$$

- Kendall's τ,

which is similar to the Spearman rank correlation, but using the relative ranks instead of the absolute ones. The Kendall's distance between two data points a and b is defined as:

$$d(a,b) = 1 - \tau \tag{8}$$

Once the distance measure is defined, each clustering algorithm proceeds by calculating the distance matrix containing all the distances between the items that are being clustered. From the above

Please cite this article in press as: C. Bouras, V. Tsogkas, A clustering technique for news articles using WordNet, Knowl. Based Syst. (2012), http://dx.doi.org/10.1016/j.knosys.2012.06.015

distance functions, only Euclidian and City-block distance are true metrics since they satisfy the triangle inequality.

5. Algorithm approach for W-k means

In this section we are presenting our algorithm approach for exploiting the WordNet database within the scope of k-means. The WordNet lexical reference system, organizes different linguistic relations into hierarchies. Most importantly, given any noun, verb, adjective and adverb, WordNet can provide results regarding hypernyms, hyponyms, meronyms or holonyms. Using these graph-like structures, we can search the WordNet database for all the hypernyms of a given set of words, then weigh them appropriately, and finally chose representative hypernyms that seem to extend the overall meaning of the set of given words. This intuitive approach, however, depends entirely on the weighing formula that will be used during the process. It is important that weighing only introduces "new knowledge" to the list of given words that will make the clustering result less fuzzy and more accurate.

5.1. Enriching articles using WordNet

Initially, for each given keyword of the article, we generate its graphs of hypernyms leading to the root hypernym (commonly being 'entity' for nouns). Following, we combine each individual hypernym graph to an aggregated one. There are practically two parameters that need to be taken into consideration for each hypernym of the aggregate tree-like structure in order to determine its importance: the depth and the frequency of appearance. For example, Fig. 3 depicts the aggregated hypernym graph for three terms: 'pie', 'apple', 'orange'.

It is observed that the higher (i.e. less deep, walking from the root node downwards) the hypernym is in the graph, the more generic it is. However, the lower the hypernym is in the graph, the less chances does it have to occur in many graph paths, i.e. its frequency of appearance is low. We can also see that each term might have multiple graph paths that lead from the term itself to the root, i.e. 'entity' node. For example in Fig. 3 the term 'apple' has three different paths: (i) apple – edible fruit – fruit, (ii) apple – edible fruit – produce, and (iii) apple – pome – fruit. Taking the previous observations into consideration, we can look up all the hypernyms of a set of given terms and then choose the best among them using a heuristic function. In order to maintain the specificity of a set of terms, while revealing their general topics, this function has to choose as low-level common hypernyms as possible. Within our work, we formulated our heuristic function as follows:

$$W(d,f) = 2 \cdot \frac{1}{1 + e^{-0.125\left(d^3 \frac{f}{TW}\right)}} - 0.5 \qquad (9)$$

where d stands for the node's depth in the graph (starting from root and moving downwards in Fig. 3), f is the frequency of appearance of the node to the multiple graph paths and TW is the number of total words that were used for generating the graph (i.e. total article's keywords, that is TW = 3 in the example depicted in Fig. 3). Function (9) is a sigmoid one in the weighted form of: $a \cdot sig(d,f) - b$. We determined the best suited values for a and b via a simple experiment. Using a corpus of 1000 pre-categorized news articles, we tried to determine the efficiency of the proposed W-k means algorithm via clustering these articles to the predetermined set of system categories. In this scenario, our clustering approach should make cluster assignments as close as possible to the categories of the articles. A variety of a, b combinations were used and the best overall result was achieved with $a = 2$ and $b = 0.5$. The steepness value of (9) is affected by both the frequency and the depth of the hypernym. We chose a sigmoid function after observing how the depth and frequency affect the generated clustering results: the importance (weight) of each hypernym exhibits a progression from small beginnings that accelerates and approaches a climax over time, a behavior that is affected by the two previously mentioned factors. For large depth – frequency combinations, the weight of the hypernym reaches closer and closer to 1 (neither f nor d can be negative), whereas for low depth – frequency combinations the weight is close to 0. A keyword having no hypernym or not being in WordNet is omitted both from the graph and the TW sum. Furthermore, a hypernym may have multiple paths to the root, but is counted only once for each given keyword. Note

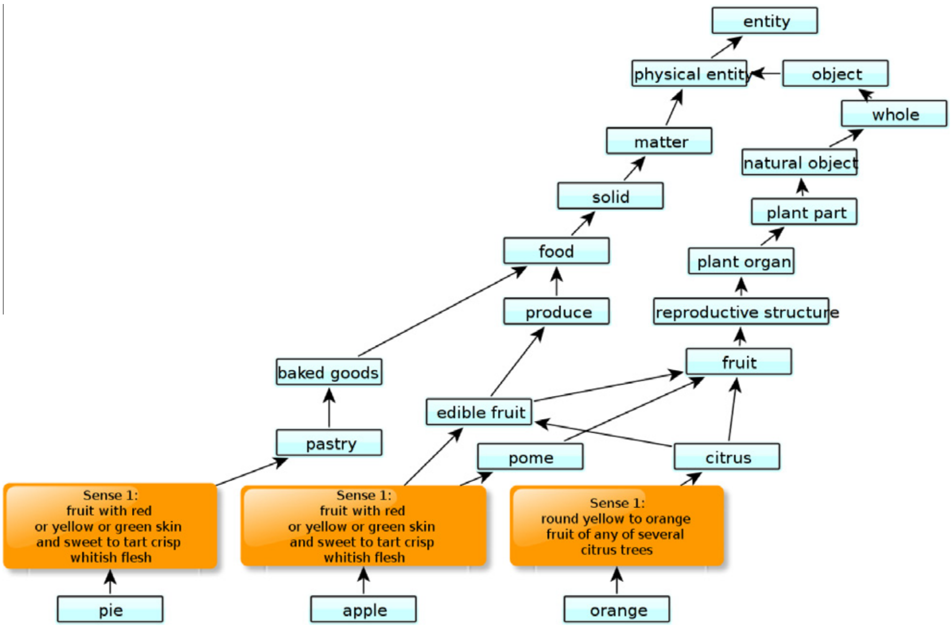
entity
physical entity
object
whole
matter
natural object
solid
plant part
food
plant organ
produce
reproductive structure
fruit
baked goods
edible fruit
pastry
pome
citrus
Sense 1: fruit with red or yellow or green skin and sweet to tart crisp whitish flesh
Sense 1: fruit with red or yellow or green skin and sweet to tart crisp whitish flesh
Sense 1: round yellow to orange fruit of any of several citrus trees
pie
apple
orange

Fig. 3. Aggregate hypernym graph for three words: 'pie', 'apple', and 'orange'.

also that the depth has a predominant role in the weighing process, much greater than frequency does. Frequency, however, acts as a selective factor when the graph expands with more and more keywords being added. We concluded to this weighing scheme after observations of hypernym graphs generated over hundreds of keywords because it scales well with real data. Given the aggregate hypernym graph in Fig. 3, we can compute the weight of the various hypernyms. For example for 'fruit': $d = 9$, $f = 2$, and $W = 0.9954$, whereas for 'edible fruit': $W = 0.8915$, and for 'food': $W = 0.6534$.

Algorithm 2: Enriching news articles using WordNet hypernyms

```
Algorithm wordnet_enrich
Input: article a
Output: enriched list of keywords
  total_hypen_tree = NULL
  kws = fetch 20% most frequent k/ws for a
    for each keyword kw in kws
      htree = wordnet_hypen_tree(kw)
        for each hypen h in htree
          if (h not in total_hypen_tree)
            h.frequency = 1
            total_hypen_tree ->append(h)
          else
            total_hypen_tree ->at(h)->freq++
    for each h in total_hypen_tree
      calculate_depth(h)
  sort_weights(total_hypen_tree)
  important_hypens = (kws ->size/
  4) * top(total_hypen_tree)
  return kws += important_hypens
```

The enriching algorithm using WordNet hypernyms, as outlined in Algorithm 2, operates on the articles keywords generating a hypernym graph for each. We use only 20% of the article's most important keywords reducing, thus, dimensionality and noise as explained in Bouras et al. [5]. Next, an aggregate graph is generated from which the weight of each hypernym is calculated using formula (9). The graph is sorted based on the nodes' weights and a list of the top keywords – hypernyms is returned, containing the suggested ones for enriching the article. In order to avoid dimensionality expansion and overgeneralization of the results, we take into consideration a total size of a quarter of the article's hypernyms for the enriching ones, which was observed to convey the best results with minimal overhead in computation time.

5.2. Labeling clusters using WordNet

In order to generate suggested labels for each resulting cluster, we are also utilizing the WordNet hypernyms information as presented in Algorithm 3. Cluster labeling operates on each cluster, fetching initially only 10% of the most important keywords belonging to each article of the cluster. We have found that this percentage is enough for the process to maintain a high quality level for the resulting labels by not introducing much noise. For each cluster's keyword we generate the hypernym graph and append it to the aggregate one. The resulting nodes are weighed, sorted and the top 5 hypernyms are returned as suggested labeling tags for the cluster. Since this is a labeling process, we believe that 5 keywords are usually enough to briefly convey the cluster's contents.

Algorithm 3: Labeling clusters using WordNet hypernyms

```
Algorithm wordnet_cl_labeling
Input: clusters
Output: cluster_labels
  for each cluster c
    total_hypen_tree = NULL
    for each article a in c
      cluster_kws += fetch 10% most frequent k/ws
  for a
    for each keyword kw in cluster_kws
      hypens_tree = wordnet_hypen_tree(kw)
      for each hypen h in hypens_tree
        if (h not in total_hypen_tree)
          h.frequency = 1
          total_hypen_tree->append_child(h)
        else
          total_hypen_tree->at(h)->frequency++
    for each hypen h in total_hypen_tree
      calculate_depth(h)
    sort_weights(total_hypen_tree)
    cluster_labels += 5 * top(total_hypen_tree)
  return cluster_labels
```

Using Algorithms 2 and 3, we can describe the algorithmic steps of W-k means as presented in Algorithm 4.

Algorithm 4: News article's clustering using W-k means

```
Algorithm W-k means
Input: articles, number of clusters
Output: cluster assignments
  for each article a
    fetch 20% most frequent k/ws for a
    wordnet_enrich(a)
    clusters = kmeans()
  return wordnet_cl_labeling (clusters)
```

6. Experimental procedure

In the current section we are presenting our experimental procedure and its results. Our analysis consists of: (a) evaluating known clustering methodologies and distance measures when applied within the domain of news articles, (b) evaluating our WordNet enabled k-means clustering and cluster labeling algorithm, and (c) comparing the proposed W-k means clustering results to those generated by two state of the art generic clustering toolboxes: Cluto [14] and SenseClusters [16].

6.1. Dataset and evaluation criteria

Within this frame we conducted a series of experiments on a predetermined set of news articles that are available in the system's database and have been offline analyzed as explained in Section 3. Our dataset consists of 10,000 randomly selected news articles originating from 20 major news portals, like bbc.com, cnn.com, reuters.com, etc. with a time span of six months. Those articles were evenly shared among the eight base categories that our system features. After the preprocessing procedure described in Section 3, and most notably stemming and noun identification, we have kept for each article its list of stemmed nouns. Notice that duplicate articles originating from different sources have been removed from the dataset based on their title and main body.

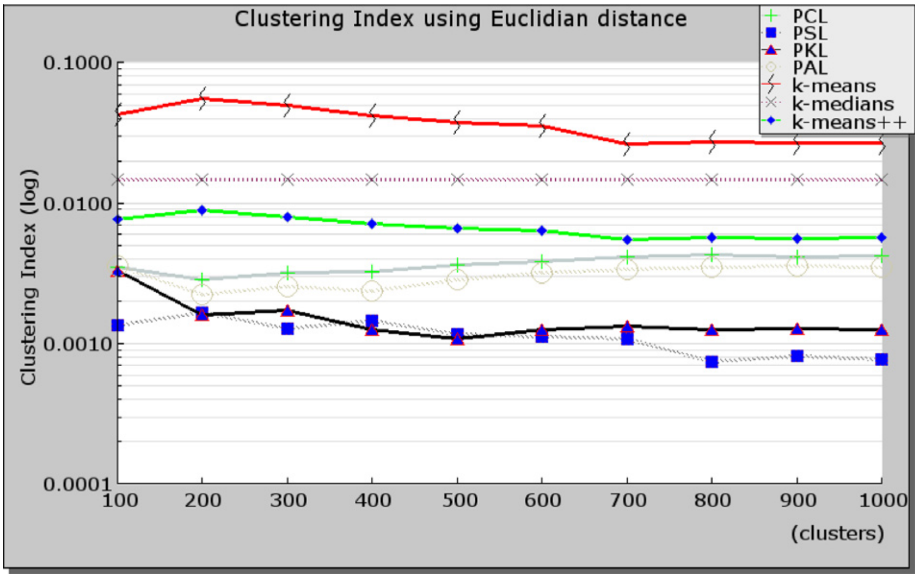

Fig. 4. Clustering results using the Euclidian distance.

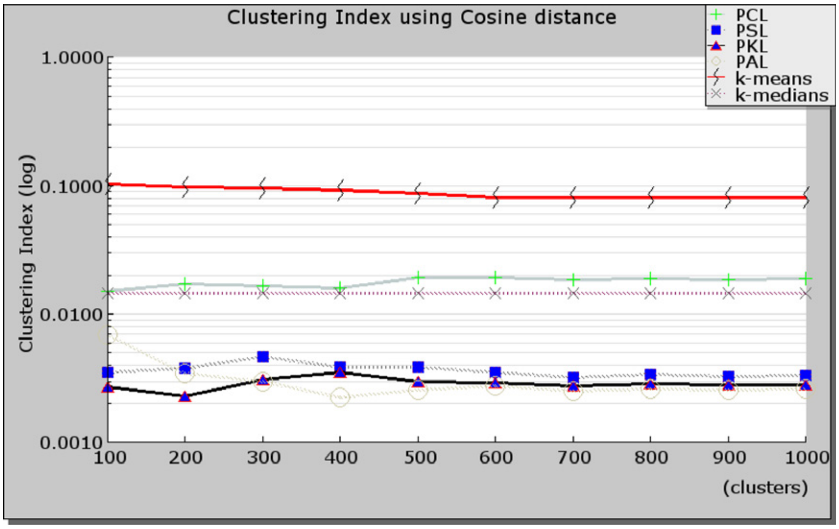

Fig. 5. Clustering results using the cosine distance.

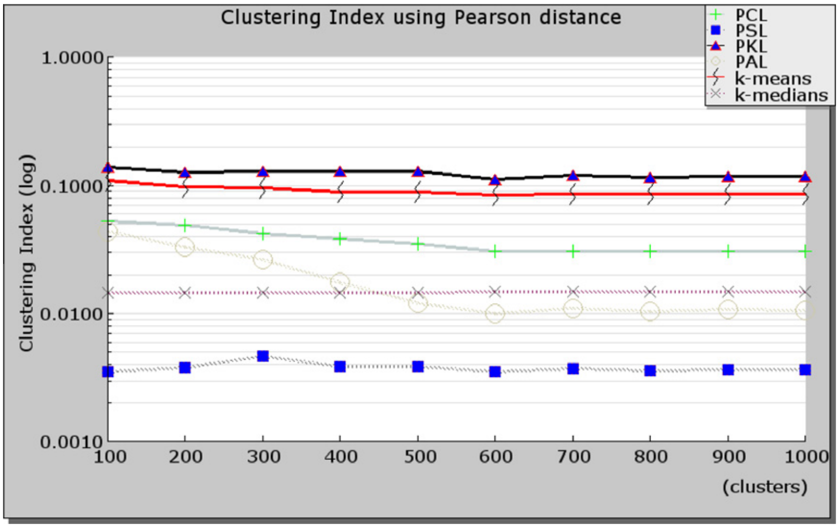

Fig. 6. Clustering results using the Pearson's distance.

Please cite this article in press as: C. Bouras, V. Tsogkas, A clustering technique for news articles using WordNet, Knowl. Based Syst. (2012), http://dx.doi.org/10.1016/j.knosys.2012.06.015

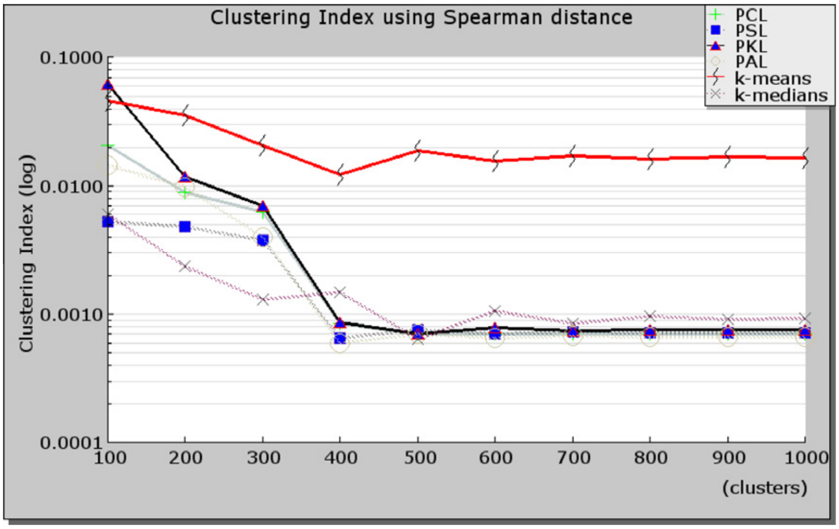

Fig. 7. Clustering results using the Spearman distance.

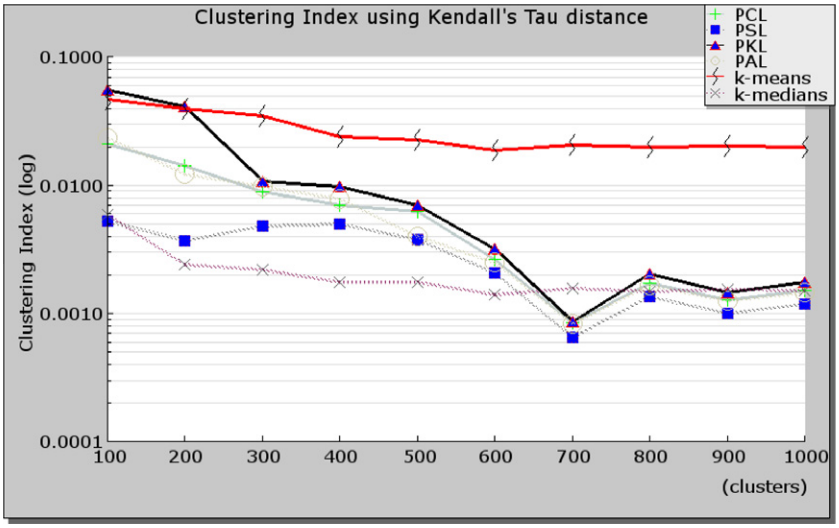

Fig. 8. Clustering results using the Kendal's tau distance.

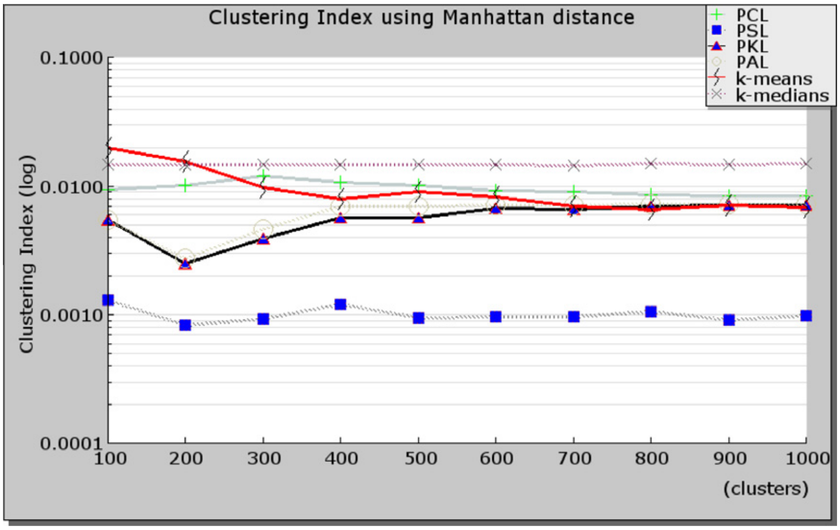

Fig. 9. Clustering results using the City-block distance.

Table 1
Hierarchical clustering notations.

Type of distance	Distance between two clusters
Pairwise maximum (complete) linkage	PCL
Pairwise single linkage	PSL
Pairwise centroid linkage	PKL
Pairwise average linkage	PAL

In order to determine the efficiency of each clustering method, we used the evaluative criteria of Clustering Index (CI) [21] and the F-measure. Intuitively, since the most efficient clusters are the ones containing articles close to each other within the cluster, while sharing a low similarity with articles belonging to different clusters, CI focuses on increasing the first measure (intra-cluster similarity) while decreasing the second (inter-cluster similarity). The Clustering Index of each pass is defined as:

$$\mathrm{CI} = \frac{\overline{\sigma}^2}{\overline{\sigma} + \overline{\delta}} \qquad (10)$$

where $\overline{\sigma}$ is the average intra-cluster similarity and $\overline{\delta}$ the average inter-cluster similarity.

The F-measure, as defined in Formula (11) is a weighed combination of the precision and recall metrics and is employed to evaluate the accuracy and efficiency of our recommendation system when using user profile clustering. We define a set to target articles, denote C, the system suggests and another set of articles, denote C', visited by the user after the recommendation process. Moreover, $r(c'_i, c_j)$is used to denote the number of documents both in the suggested and in the visited lists.

$$F(c'_i, c_j) = 2 \cdot \frac{r(c'_i, c_j)p(c'_i, c_j)}{r(c'_i, c_j) + p(c'_i, c_j)} \qquad (11)$$

where $r(c'_i, c_j) = \frac{doc(c'_i, c_j)}{doc(c'_i)}$ and $p(c'_i, c_j) = \frac{doc(c'_i, c_j)}{doc(c_j)}$

6.2. Evaluation of common clustering techniques

On the previously mentioned dataset we applied the following clustering methodologies: single, maximum, linkage and centroid linkage hierarchical clustering, as well as regular k-means, k-medians and k-means++. For those, we utilized the open source clustering library [13] as well as the k-means++ implementation [4]. Furthermore, for each of the above techniques, except k-means++ (which only supports Euclidian) we used the similarity measures described in Section 4, i.e. Euclidian distance, city-block distance, Pearson correlation coefficient, cosine similarity, Spearman-rank correlation and Kendall's τ. For partitional algorithms, we used a 10 pass scheme with different starting conditions in order to avoid phenomena of local minima for the distance measures.

Furthermore, for determining the similarity between two articles we used the distance vector which is produced using the respective similarity measure per case. The results for each clustering methodology and distance measure run for a number of clusters from 100 to 1000, are depicted in Figs. 4–9. The notions mentioned in the graphs are explained in Table 1.

From the above graphs, we can observe that k-means almost always outperforms any other clustering approach. Furthermore, cosine similarity and Euclidian distance proves better for k-means, since the clusters seem better connected, rather than with the city-block distance, which seems to be better fit to k-medians. Another observation is that the number of clusters directly affects the CI metric and that after a certain cluster threshold, each algorithm deteriorates in terms of CI. For example, the best CI for partitional algorithms is observed for k-means/cosine similarity and 100 clusters followed by k-means/Euclidian and 200 clusters. The best CI scores for hierarchical algorithms are observed for PKL and the Pearson's distance. Moreover, for most similarity measures we observed lower CI scores for hierarchical methodologies compared to partitional approaches. This originates from the manner that those algorithms operate when "cutting" the dendrogram: generation of

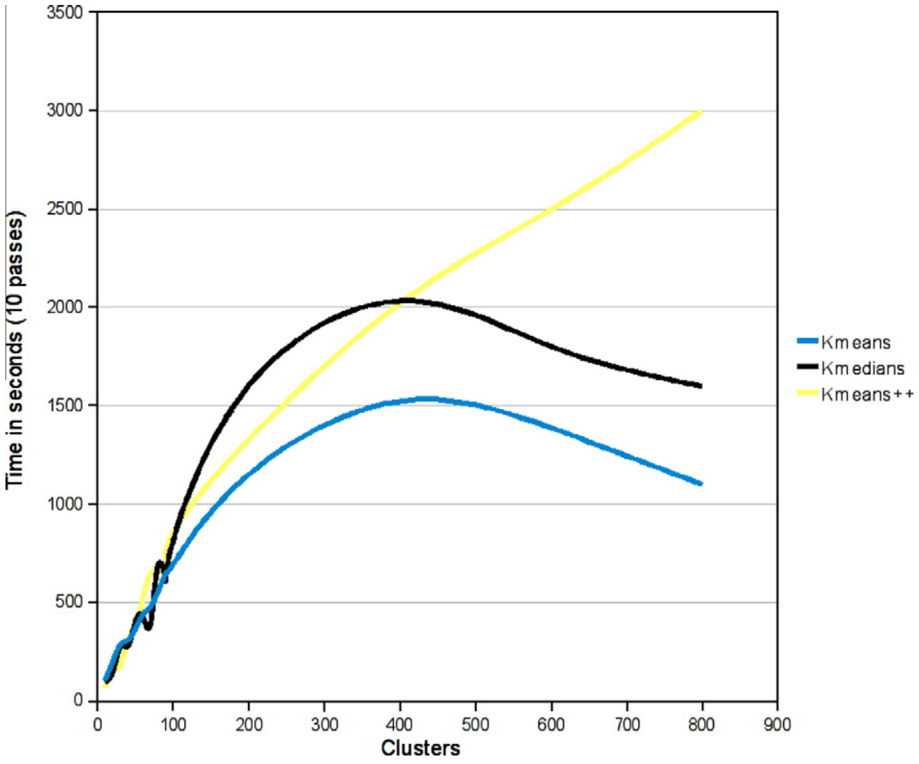

Fig. 10. Average intra-cluster sum of distances for partitional clustering.

Please cite this article in press as: C. Bouras, V. Tsogkas, A clustering technique for news articles using WordNet, Knowl. Based Syst. (2012), http://dx.doi.org/10.1016/j.knosys.2012.06.015

Table 2
The effect of preprocessing on the various clustering methodologies.

Clustering method	Percent of increase for CI when using stemming and noun identification (%)
PCL	5
PSL	6
PKL	6
PAL	5
k-Means	18
k-Medians	16
k-Means++	15

Table 3
Users' evaluation of the various clustering methodologies.

Clustering method	*F* result
PCL	0.42
PSL	0.42
PKL	0.43
PAL	0.41
k-Means	0.61
k-Medians	0.57
k-Means++	0.51

many singleton clusters and a few clusters containing many articles.

As far as partitional clustering is concerned, k-means outperforms k-medians and even k-means++ which seems to deteriorate sooner as the number of clusters increases. Moreover, as Fig. 10 shows, k-means++ is significantly slower than its counterparts given the number of clusters.

Following, we repeated the aforementioned experimentation omitting the steps of stemming and noun identification from the preprocessing procedure. The average modification of the CI results is presented in Table 2. Clearly, stemming and noun identification on the articles' keywords has a significantly beneficial effect for all methodologies, especially for k-means, partly explaining its superiority regarding CI results as presented earlier in Figs. 4–9.

Even though internal objective functions like CI are capable of giving a generic overview of the clustering process efficiency, an alternative approach is user-based evaluation. Based on this intuition, for our final set of experiments we tried to evaluate the generated clusters by using a group of 10 individuals. We requested that they grouped 50 random articles from the previous data set into 10 clusters according to their personal opinion. Afterwards we averaged their clustering selections and compared those results with the clustering passes of each of the various methodologies explained earlier using the Euclidian similarity distance. The evaluation metric at this case is the *F* measure, i.e. the weighed harmonic mean of the precision and recall observed between the users choices and the results generated by each clustering pass. The *F* results per clustering pass, depicted in Table 3 show that from a user based perspective, the resulting clusters produced by k-means are closer to what most of the users selected for the selected data set of articles. In order to determine the confidence level of the below *F* results, we also calculated the inter-annotator agreement. For this, we used the joint-probability of agreement, defined as the number of times that each of the 50 articles was assigned to each of the 10 clusters, divided by the total number of assignments. Using the above, we calculated the inter-annotator agreement to be on average 0.83 giving thus a high confidence level for the results presented in Table 3.

6.3. Evaluating W-k means

For our first experimentation set towards evaluating W-k means, we run both of the k-means and W-k means algorithms on the dataset and observed the CI scores over varying categories, number of articles and number of clusters. Note that given the experimental results of Section 6.2, comparing W-k means to the clustering approaches previously evaluated is redundant, since k-means with the cosine similarity metric has proven to be the best choice for our dataset. For the results presented in Fig. 11 and Fig. 12, the first graph (Fig. 11) gives the CI for the case of WordNet enriched executions of the k-means algorithm (W-k means), compared to the non-enriched ones (Fig. 12). It is clearly depicted that W-k means gives significantly improved clustering results when applied in our data set, regardless of the number of articles or the category they belong to. This provides a confirmation for the initial hypothesis that using outside features from the English language, apart from only textual – extracted features can be particularly useful. Another observation is that as the number of articles increases, the CI difference of W-k means compared to k-means

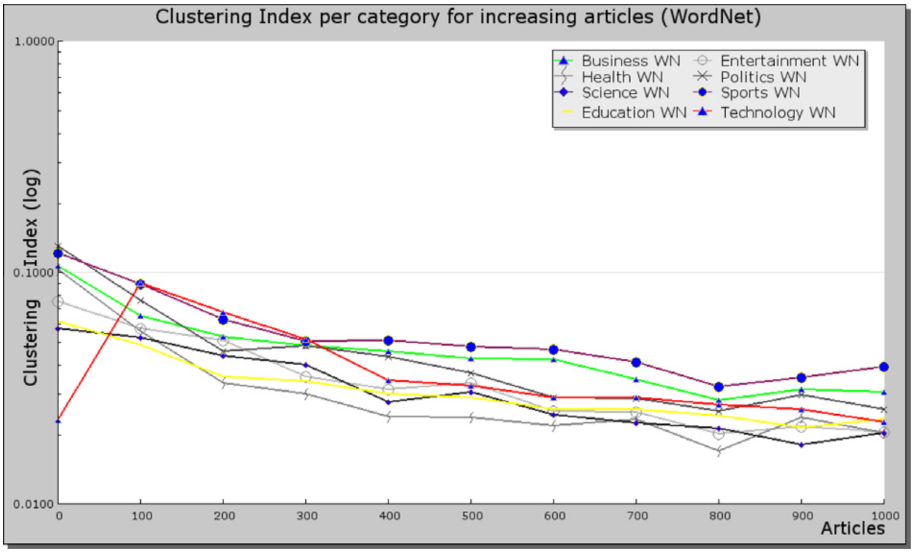

Fig. 11. Evaluating clustering over articles belonging to various categories (with WordNet use).

Please cite this article in press as: C. Bouras, V. Tsogkas, A clustering technique for news articles using WordNet, Knowl. Based Syst. (2012), http://dx.doi.org/10.1016/j.knosys.2012.06.015

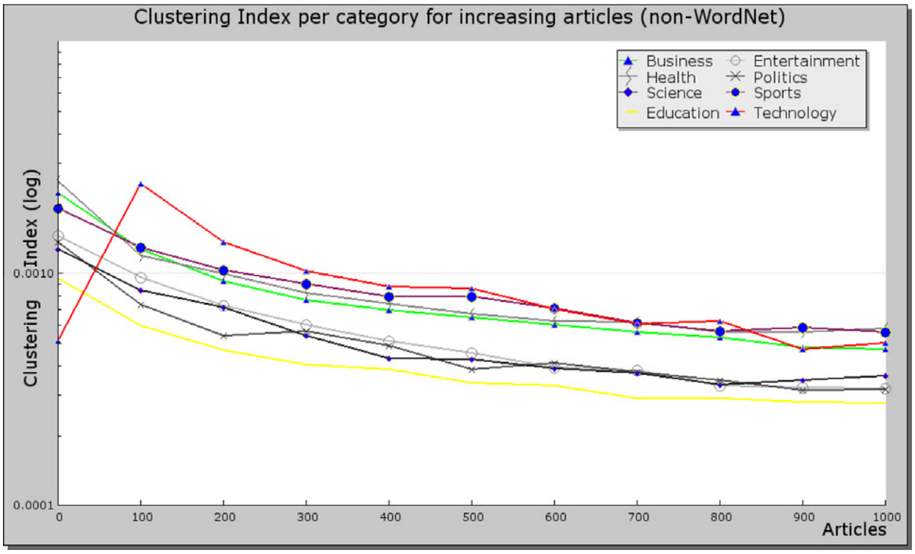

Fig. 12. Evaluating clustering over articles belonging to various categories (without WordNet use).

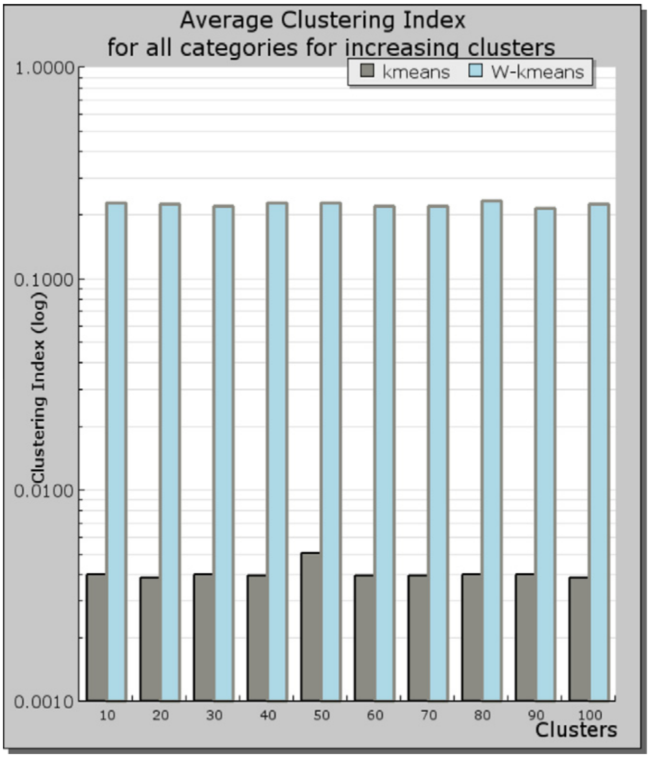

Fig. 13. Averaging clustering index over categories for various cluster numbers.

Table 4
Precision results for cluster labeling over various categories using W-k means.

Category	W-k means precision (%)
Business	85
Entertainment	78
Health	90
Politics	88
Science	65

Table 5
Clustering index comparison between Cluto, SenseCluster and W-k means.

Approach	CI	Execution time (s)
Cluto	0.85	204 s (average for five executions)
SenseCluster	0.56	302 s
W-k means	0.84	198 s

gets wider. We believe that this is because of the fact that while our experimentation data set grows larger, the probability of hypernyms occurring also increases. Therefore, our clustering approach has a better chance of selecting clusters with improved connectivity while at the same time keeping different clusters well separated from each other. Fig. 13 presents the CI results for a variety of cluster numbers as averaged over all the categories. The improvement, as before, is more than ten times over CI scores obtained with normal k-means (logarithmic scales in all Figs. 11–13). We also pinpointed that for the case of 50 clusters, the results are slightly improved over the rest of the cases which can be interpreted as a viable indication of the actual number of clusters our data set seems to have. Indeed, at a later stage we verified that the actual number of clusters in our dataset was 51. In order to determine this, we started with one cluster and then kept splitting clusters until the articles assigned to each cluster had a Gaussian distribution. This analysis, however, is beyond the scope of the current manuscript.

For our second experimentation set regarding W-k means, we evaluated the labeling results of the proposed algorithm. In order to do so, we applied W-k means over our data set using a total number of eight clusters. Since the articles of the data set are pre-categorized to one of the eight categories used, we compared the resulting cluster labels to aggregate lists created for each category containing: (a) the 10 most frequent keywords of each category and (b) the category name itself. Labels getting ‘close’ (i.e. synonyms or derivatives) to the contents of the aggregate list are considered as representative ones. In addition, the category’s aggregate list to which a cluster has the most labels belonging to is accepted as the representative category for this cluster. We evaluated the accuracy of the labeling process using the precision of the suggested cluster labels against the aggregate list of the category that the respective cluster belongs to. Precision for labeling i and its belonging category j is defined as:

Please cite this article in press as: C. Bouras, V. Tsogkas, A clustering technique for news articles using WordNet, Knowl. Based Syst. (2012), http://dx.doi.org/10.1016/j.knosys.2012.06.015

$$precision(label_i, category_j) = avg\_rank(i,j) \cdot \frac{a}{a+b} \tag{12}$$

where $avg_rank(i,j)$is the average rank that labeling i has in the aggregate list of category j, a is the number of terms labeling i has for category j and b is the number of terms that labeling i has but that are not in the jth's category aggregate list. The precision results per category presented in Table 4 show an overall precision rate of 75% for our labeling approach which would have been even better if the 'technology' and 'science' categories were not so closely related to each other.

For our final experimentation set, we compared the clustering results of the W-k means approach with the ones generated by two state of the art generic clustering toolboxes: Cluto [14] and SenseClusters [16]. We employed the same corpus and also kept clustering index as our evaluation criterion. The number of clusters in our dataset was assumed to be 50. For the case of Cluto, the clustering index of all five different approaches (four partitional and one hierarchical) was averaged, while for SenseCluster we used its hierarchical clustering representation. The CI values are aggregated over all the generated clusters by each methodology. The results depicted in Table 5, also capture the execution time needed for the aforementioned approaches in our dataset.

From the above results we see that despite its simplicity, W-k means produces clustering results that are almost as good as the ones generated by the Cluto toolkit and even better than the ones generated by the SenseCluster toolkit in terms of CI. Moreover, the execution time is significantly better for W-k means which can be explained by the simple nature of the algorithm.

7. Conclusion

Within the scope of our indexing system, we have presented our evaluation results comparing some of the best clustering options currently available, applying them to the domain of news articles that originate from the Web. From the plethora of similarity measures that have been used, the appliance of Euclidian and cosine k-means produced the best results based not only on the internal CI function, but also on a real users' experimentation. More specifically, we have found that hierarchical clustering techniques resulted generally in worse CI scores, while partitional clustering, even though non-deterministic, can provide exceptional results. Another important finding is that preprocessing of the articles via stemming and noun identification can improve significantly the clustering results by a factor of 5–15% depending on the clustering algorithm.

We have also presented a novel algorithmic approach towards enhancing the k-means algorithm using knowledge from an external database, WordNet, in a twofold manner. W-k means firstly enriches the clustering process itself by utilizing hypernyms and secondly, generates useful labels for the resulting clusters. We have measured a 10-times improvement over the standard k-means algorithm in terms of high intra-cluster similarity and low inter-cluster similarity. Furthermore, the resulting labels are with high precision the correct ones as compared with their category tagging counterparts.

8. Future work

For the future, we will be evaluating W-k means with regards to time efficiency using more clustering algorithms and larger document sets. We are also planning on determining how well our approach scales with increasing numbers of articles as is the case with online indexing services. Moreover, we will be researching towards using the clustering kernel for clustering system users based on their dynamic profiles, and we will proceed with evaluating more extensively the clustering module with user feedback.

Acknowledgements

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) – Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

References

[1] A.A. Abdelmalek, E. Zakaria, S. Michel, Evaluation of text clustering methods using WordNet, The International Arab Journal of Information Technology 7 (4) (2010).

[2] A.Y. Al-Omary, M.S. Jamil, A new approach of clustering based machine-learning algorithm, Knowledge-Based Systems 19 (4) (2006) 248–258.

[3] D. Arthur, S. Vassilvitskii, On the Worst Case Complexity of the k-means Method, Technical Report, Stanford, 2005.

[4] D. Arthur, S. Vassilvitskii, k-Means++: the advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2007, pp. 1027–1035.

[5] C. Bouras, V. Poulopoulos, V. Tsogkas, PeRSSonal's core functionality evaluation: enhancing text labeling through personalized summaries, Data and Knowledge Engineering Journal 64 (1) (2008) 330–345 (Elsevier Science).

[6] C. Bouras, V. Tsogkas, Improving text summarization using noun retrieval techniques, Lecture Notes in Computer Science, Knowledge-Based Intelligent Information and Engineering Systems 5178 (2008) 593–600.

[7] P.S. Bradley, U. Fayyad, Refining initial points for k-means clustering, in: Proceedings of the 15th International Conference on Machine Learning, 1998, pp. 91–99.

[8] D. Carmel, H. Roitman, N. Zwerdling, Enhancing cluster labeling using wikipedia, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development Information Retrieval, 2009, pp. 139–146.

[9] C.L. Chen, S. Frank, C. Tseng, T. Liang, An integration of fuzzy association rules and WordNet for document clustering, Lecture Notes in Computer Science in Advances in Knowledge Discovery and Data Mining (2009) 147–159.

[10] W.H.E. Day, H. Edelsbrunner, Efficient algorithms for agglomerative hierarchical clustering methods, Journal of Classification (1984) 7–24.

[11] A. El-Hamdouchi, P. Willett, Comparison of hierarchic agglomerative clustering methods for document retrieval, The Computer Journal 32 (1989) 220–227.

[12] T.F. Gharib, M.M. Fouad, M.M. Aref, Fuzzy document clustering approach using WordNet lexical categories, in: Khaled Elleithy (Ed.), Advanced Techniques in Computing Sciences and Software Engineering, Springer Science+Business, Media B.V., 2010, p. 181. ISBN 978-90-481-3659-9.

[13] M. Hoon, S. Imoto, S. Miyano, The C Clustering Library, Institute of Medical Science, Human Genome Center, University of Tokyo, 2003.

[14] G. Karypis, {CLUTO} a clustering toolkit, Technical Report 02-017, Dept. of Computer Science, University of Minnesota, 2002. <http://www.cs.umn.edu~cluto>.

[15] G.V.R. Kiran, R. Shankar, V. Pudi, Frequent itemset based hierarchical document clustering using Wikipedia as external knowledge, in: KES'10 Proceedings of the 14th International Conference on Knowledge-Based and Intelligent Information And Engineering Systems: Part II, 2010, pp. 11–20.

[16] A. Purandare, T. Pedersen, SenseClusters – finding clusters that represent word senses. in: Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04), San Jose, USA, 2004, pp. 25–29.

[17] S. Rodpongpun, V. Niennattrakul, C.A. Ratanamahatana, Selective subsequence time series clustering, Knowledge Based Systems (2012), http://dx.doi.org/10.1016/j.knosys.2012.04.022.

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

[19] J. Sedding, D. Kazakov, WordNet-based text document clustering, in: Proceedings of COLING-Workshop on Robust Methods in Analysis of Natural Language Data, 2004.

[20] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, Department of Computer Science and Engineering, University of Minnesota, 2000.

[21] J. Taeho, L. Malrey, The evaluation measure of text clustering for the variable number of clusters, Advances in Neural Networks 4492 (2007) 871–879. ISNN 2007.

Please cite this article in press as: C. Bouras, V. Tsogkas, A clustering technique for news articles using WordNet, Knowl. Based Syst. (2012), http://dx.doi.org/10.1016/j.knosys.2012.06.015

[22] P. Treeratpituk, J. Callan, Automatically labeling hierarchical clusters, in: Proceedings of the 2006 International Conference on Digital Government Research, San Diego, California, 2006 May 21–24.

[23] Y.H. Tseng, Generic title labeling for clustered documents. Expert Systems with Applications 37 (3) (2009) 2247–2254 (Elsevier).

[24] G. Varelas, E. Voutsakis, P. Raftopoulou, E. Petrakis, E. Milios, Semantic similarity methods in wordNet and their application to information retrieval on the web, in: Workshop on Web Information and Data Management, Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management, 2005, pp. 10–16.

[25] L. Yanjun, C. Soon, Parallel bisecting k-means with prediction clustering algorithm, The Journal of Supercomputing 39 (2007) 19–37.

[26] Y. Zhao, G. Karypi, Empirical and theoretical comparisons of selected criterion functions for document clustering, Machine Learning 55 (3) (2004) 311–331.

Please cite this article in press as: C. Bouras, V. Tsogkas, A clustering technique for news articles using WordNet, Knowl. Based Syst. (2012), http://dx.doi.org/10.1016/j.knosys.2012.06.015