W-kmeans: Clustering News Articles Using WordNet

Christos Bouras[1,2] and Vassilis Tsogkas[1]

[1] Computer Engineering and Informatics Department, University of Patras, Greece
[2] Research Academic Computer Technology Institute
N. Kazantzaki, Panepistimioupoli Patras, 26500 Greece
bouras@cti.gr, tsogkas@ceid.upatras.gr

Abstract. 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 suffering however from problems like synonymy, ambiguity and lack of a descriptive content marking of the generated clusters. We are proposing the enhancement of standard kmeans 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. Our experimentation revealed a significant improvement over standard kmeans for a corpus of news articles derived from major news portals. Moreover, the cluster labeling process generates useful and of high quality cluster tags.

Keywords: News clustering, k-means, Cluster Labeling, Partitional Clustering.

1 Introduction

While the amount of online information sources is rapidly increasing, so does the available online news content. One of the commonest approaches for organizing this immense amount of data is the use of clustering techniques. 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 vise-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 problems that document clustering techniques regularly fail to tackle.

Furthermore, having IR systems simply generate clusters of documents is not enough per se. The reason is that it's virtually impossible for humans to conceptualize information by merely browsing though 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.

Two generic categories of the various clustering methods exist: agglomerative hierarchical and partitional. Typical hierarchical techniques generate a series of partitions

R. Setchi et al. (Eds.): KES 2010, Part III, LNAI 6278, pp. 379–388, 2010.

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 tree-like structure. On the other hand, partitional algorithms typically determine all clusters at once. For partitional techniques, a global criterion is 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.

The family of k-means partitional clustering algorithms [1] usually tries to minimize the average squared distance between points in the same cluster, i.e. if d_1, d_2,…, d_n are the n documents and c_1, c_2,…, c_k are the k clusters centroids, k-means tries to minimize the global criterion function:

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

Several improvements have been proposed over this simple scheme, like bisecting k-means [2], k-means++ [3] and many more.

WordNet is one of the most widely used thesauri for 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 [4], or their association with clustering techniques. For example in [5] they combine the WordNet knowledge with fuzzy association rules and in [7], they extend the bisecting k-means using WordNet; their methodology however is rather unclear.

Regarding cluster labeling, techniques frequently evaluate labels using information from the cluster themselves [8], while existing approaches that utilize other external databases, like Wikipedia [6] are only good for the labeling process and not the clustering one. Recently in [9], WordNet hypernyms were used for the labeling process; we found however that their weighting scheme didn't scale well with the number of documents.

In this paper we are presenting a novel algorithmic approach towards document clustering, and in particular, clustering of news articles deriving from the Web, that combines regular kmeans with external information extracted from the WordNet database. We are also incorporating the proposed algorithm in our existing system [10], evaluating the clustering results compared to regular kmeans using a large pool of Web news articles existing in the system's database.

2 Information Flow

The flow of information as handled by our approach 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 meta-data 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 to the extraction of the keywords each article consists of. Analyzed in [10], keyword extraction handles the cleaning of articles, the extraction of the nouns [11], the stemming as well as the stopword removal process. Following, it applies several heuristics to come up with a weighting scheme that appropriately weights the keywords of each article based on information about the rest of the documents in our database. Pruning of words, appearing with low frequency throughout the corpus, which are unlikely to appear in more than small number of articles, comes next. Keyword extraction, utilizing the vector space model [12], generates the term-frequency vector, describing each article that will be used by the clustering approach technique that follows, as a 'bag of words' (words – frequencies).

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 kmeans clustering procedure that follows. In this work, clustering is achieved via regular kmeans using the cosine similarity distance measure:

$$d(a,b) = \cos(\theta) = \frac{a \cdot b}{|a||b|} \tag{2}$$

Where |α|, |b| are the lengths of the vectors α, b respectively and the similarity between the two data points is viewed by means of their angle in the n-dimensional space. It is important to note however that the clustering process is independent of the rest of the steps, meaning that it can easily be replaced by any other clustering approach operating on a word-level of the input documents.

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, as well as labels are the output of the proposed approach.

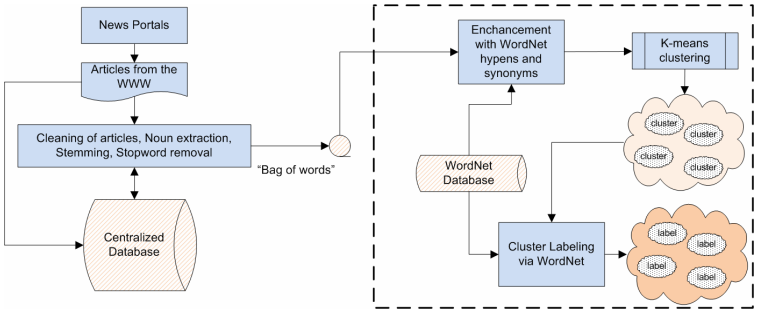

Fig. 1. Information flow for clustering news articles

3 Algorithm Approach

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 weight 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 weighting formula that will be used during the process. It is important that weighting only introduces "new knowledge" to the list of given words that will make the clustering result less fuzzy and more accurate.

3.1 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. 2 depicts the aggregated hypernym graph for three terms: 'pie', 'apple', 'orange'.

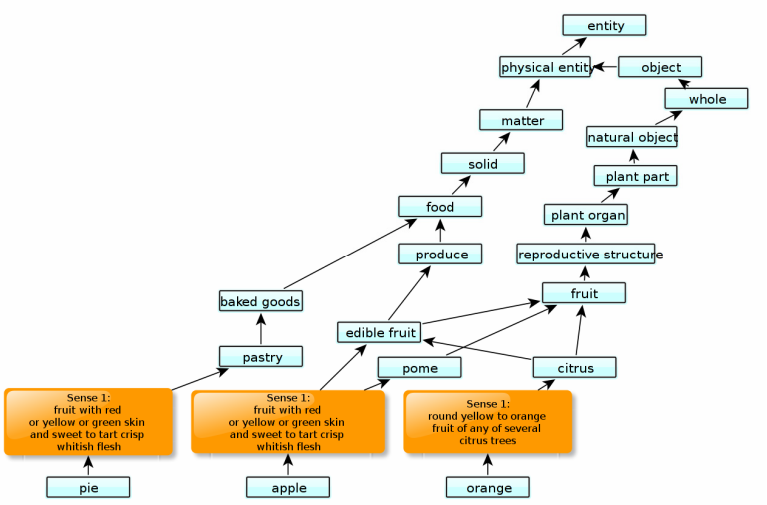

Fig. 2. Aggregate hypernym graph for three words: 'pie', 'apple', 'orange'

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 it has to occur in many graph paths, i.e. its frequency of appearance is low. In our approach, those two contradicting parameters are weighted using the formula (3).

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

where d stands for the node's depth in the graph (starting from root and moving downwards), 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). Function (3) is a sigmoid one with a steepness value including both the frequency and the depth of the hypernym. 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 also that the depth has a predominant role in the weighting 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 weighting 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. 2, we can compute the weight of the various hypernyms. For example for 'fruit': d = 9, f = 2 and W = 0.9954, where for 'edible fruit': W = 0.8915, and for 'food': W = 0.6534.

The enriching algorithm using WordNet hypernyms, as outlined in Algorithm 1, 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 [10]. Following, an aggregate graph is generated from which the weight of each hypernym is calculated using formula (3). 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. We take into consideration a total size of a quarter of the article's hypernyms for the enriching ones.

```
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)
   weight = 2 ((1/(1+ exp(-0.0125 * (h->depth ^3 * h->freq/
   kws_in_wn->size)))) - 0.5))
 sort_weights(total_hypen_tree)
 important_hypens = (kws ->size/4)*top(total_hypen_tree)
 return kws += important_hypens
```

Alg. 1. Enriching news articles using WordNet hypernyms

3.2 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 2. Cluster labeling operates on each cluster, fetching initially 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 weighted, sorted and the top 5 hypernyms are returned as suggested labeling tags for the cluster. Using Algorithm 1 and 2, we can describe the algorithmic steps of W-kmeans as presented in Algorithm 3.

```
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)
   weight = 2 ((1/(1+ exp(-0.0125 * (h->depth ^3 * h->frequency/
   kws_in_wordnet ->size)))) - 0.5))
  sort_weights(total_hypen_tree)
  cluster_labels+=5*top(total_hypen_tree)
 return cluster_labels
```

Alg. 2. Labeling clusters using WordNet hypernyms

```
Algorithm W-kmeans
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)
```

Alg. 3. News article's clustering using W-kmeans

4 Experimental Procedure

For our experiments we used a set of 8000 news articles obtained from major news portals like BBC, CNN, etc. over a period of 2 months. Those articles were evenly shared among the 8 base categories that our system features. In order to determine the

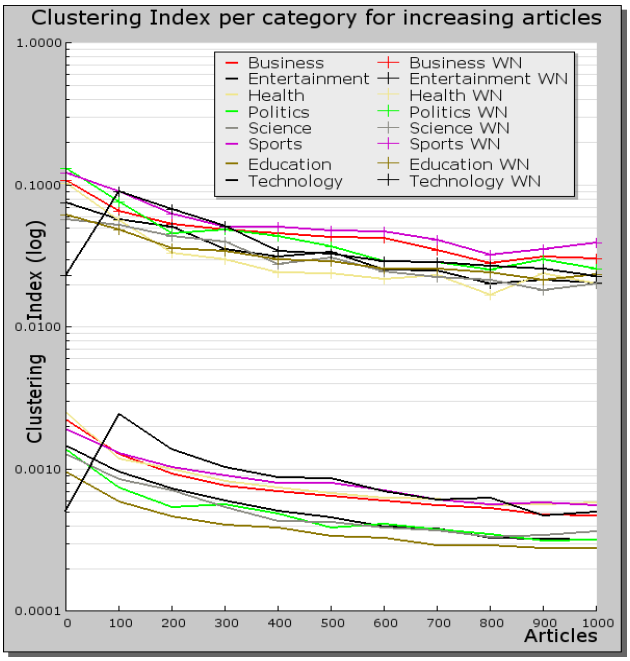

Fig. 3. Evaluating W-kmeans clustering over articles belong to various categories

efficiency of each clustering method, we used the evaluative criterion of Clustering Index (CI) as explained in [13], defined as: $CI=\overline{\sigma}^2/(\overline{\sigma}+\overline{\delta})$, where $\overline{\sigma}$ is the average intra-cluster similarity and $\overline{\delta}$ is the average inter-cluster similarity. For our first experimentation set, we run both of the kmeans and W-kmeans algorithms on the dataset and observed the CI scores over varying categories, number of articles and number of clusters. For the results presented in Fig. 3, the top set of lines gives the CI for the case of WordNet enriched executions of the kmeans algorithm, compared to the non enriched ones (bottom set). It is clearly depicted that the quality of the kmeans algorithm has improved significantly when applied in our data set regardless the number of articles or the category they belong.

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 increase, the CI difference of W-kmeans compared to kmeans 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. Fig. 4 presents the CI results for a variety of cluster numbers as averaged over all the categories (i.e. over all 8000 articles). The improvement, as before, is more than ten times over CI scores obtained with normal k-means (logarithmic scales in both Fig. 3 and 4). 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.

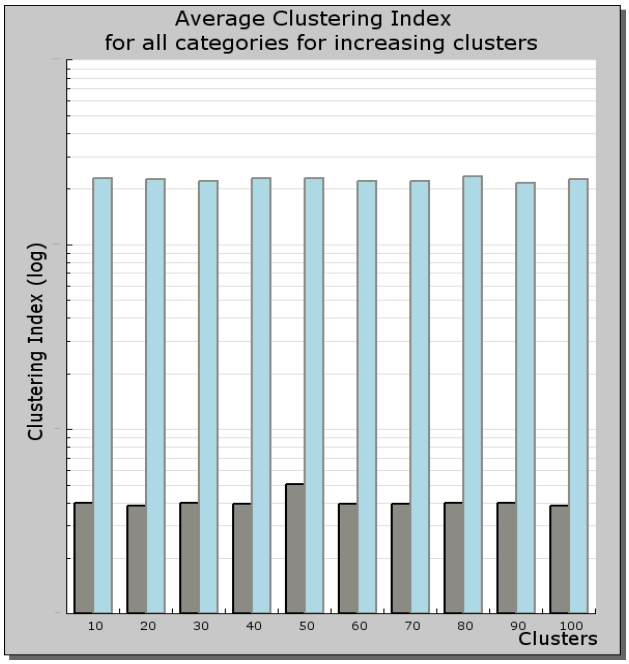

Fig. 4. Averaging Clustering Index over categories for various cluster numbers

For our second experimentation set, we evaluated the labeling results of the proposed algorithm. In order to do so, we applied W-kmeans over our data set using a total number of 8 clusters. Since the articles of the data set are pre-categorized to one of the 8 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 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:

$$precision\ (label_i, category_j) = avg\_rank\ (i, j) \cdot \frac{a}{a+b} \tag{4}$$

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 are not in the j[th]'s category aggregate list. The precision results per category presented in Table 1 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.

Table 1. Precision results for cluster labeling over various categories using W-kmeans

Category	W-kmeans Precision
Business	85%
Entertainment	78%
Health	90%
Politics	88%
Science	65%

5 Conclusion

We have presented a novel algorithmic approach towards enhancing the kmeans algorithm using knowledge from an external database, WordNet, in a twofold manner. W-kmeans 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 kmeans 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. As a future enhancement, we will be evaluating W-kmeans with regards to time efficiency using more clustering algorithms and larger document sets

References

[1] Zhao., Y., Karypi, G.: Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering. Machine Learning 55(3), 311–331 (2004)
[2] Yanjun, L., Soon, C.: Parallel bisecting k-means with prediction clustering algorithm. The Journal of Supercomputing 39, 19–37 (2007)
[3] Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proc. of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027–1035 (2007)
[4] Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E., Milios, E.: 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, pp. 10–16 (2005)
[5] Chen, C.-L., Frank, S., Tseng, C., Liang, T.: An integration of fuzzy association rules and wordNet for document clustering. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 147–159. Springer, Heidelberg (2009)
[6] Carmel, D., Roitman, H., Zwerdling, N.: Enhancing cluster labeling using wikipedia. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development ininformation retrieval, pp. 139–146 (2009)
[7] Sedding, J., Kazakov, D.: WordNet-based text document clustering. In: Proc. of COLING-Workshop on Robust Methods in Analysis of Natural Language Data (2004)
[8] Treeratpituk, P., Callan, J.: Automatically labeling hierarchical clusters. In: Proceedings of the 2006 international conference on Digital government research, San Diego, California, May 21-24 (2006)

[9] Tseng, Y.H.: Generic title labeling for clustered documents. In: Expert Systems With Applications, vol. 37(3), pp. 2247–2254. Elsevier, Amsterdam (2009)
[10] Bouras, C., Poulopoulos, V., Tsogkas, V.: PeRSSonal's core functionality evaluation: Enhancing text labeling through personalized summaries. Data and Knowledge Engineering Journal, Elsevier Science 64(1), 330–345 (2008)
[11] Bouras, C., Tsogkas, V.: Improving text summarization using noun retrieval techniques. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS (LNAI), vol. 5178, pp. 593–600. Springer, Heidelberg (2008)
[12] Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM 18(11), 613–620 (1975)
[13] Taeho, J., Malrey, L.: The Evaluation Measure of Text Clustering for the Variable Number of Clusters. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4492, pp. 871–879. Springer, Heidelberg (2007)