Enhancing News Articles Clustering using Word N-Grams

Christos Bouras[1,2] and Vassilis Tsogkas[1]
[1]*Computer Engineering and Informatics Department, University of Patras, Patras, Greece*
[2]*Computer Technology Institute and Press "Diophantus", Patras, Greece*
bouras@cti.gr, tsogkas@ceid.upatras.gr

Keywords: n-Grams, Clustering, k-Means, w-Kmeans, Text Preprocessing.

Abstract: In this work we explore the possible enhancement of the document clustering results, and in particular clustering of news articles from the web, when using word-based n-grams during the keyword extraction phase. We present and evaluate a weighting approach that combines clustering of news articles derived from the web using n-grams, extracted from the articles at an offline stage. We compared this technique with the single minded bag-of-words representation that our clustering algorithm, W-kmeans, previously used. Our experimentation revealed that via tuning of the weighting parameters between keyword and n-grams, as well as the n itself, a significant improvement regarding the clustering results metrics can be achieved. This reflects more coherent clusters and better overall clustering performance.

1 INTRODUCTION

Clustering, i.e. the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups, plays a central role in data analysis. It can give useful information about the structure of the underlying data. By discovering interesting information kernels and distributions, it has proven to be a key technique for the information retrieval (IR) and decision making process. Clustering also plays a crucial role in organizing large collections and can be used a) 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.

An n-gram is defined either as a textual sequence of length n, or similarly, as a sequence of n adjacent 'textual units', both extracted from a particular document. A 'textual unit' can be identified at a byte, character or word level depending on the context of interest. In this work, we are dealing with word n-grams which can be conceptualized by placing a small sliding window over a sentence of a given text, in which only n words are visible at a given time (Robertson and Willett, 1998). At each position of the window, the sequence of words inside it is recorded. In some schemes, the window may be slid more than one word after each n-gram is recorded. The simplest n-gram is the so-called unigram, where n=1, which falls back to the single minded bag-of-words (BOW) representation. Typically, n is a fixed number, highly dependent on the particular corpus of documents and the queries made against that corpus. Each of the n-grams is a coordinate in a vector which represents the text under study and the frequency that this n-gram appears in the text can be the number of this coordinate. We can hence use this representation for text comparison, and as such, it can find many uses in various information retrieval (IR) tasks, including but not limited to item clustering.

The use of n-gram probability distribution and n-gram models in NLP is a relatively simple idea, but it has been found to be effective in many applications. For example, character level n-gram language models can be easily applied to any language, and even non language sequences such as DNA and music. They are also widely used in text compression, e.g. the PPM model (Cleary, Bell and Witten, 1990), and have recently been found to be effective in text mining problems as well (Mahoui et al., 1999). In the domain of language independent text categorization, Damerau, Apte and Weiss (1994), have used word-based language modeling techniques for both English and German with good results. N-grams analysis has proven of great significance in many areas of natural language processing and text mining, such as text parsing or information retrieval applications. Typical examples include a) searching and categorization of similar documents (Cavnar and Trenkle, 1994) where the authors present a character n-gram-based approach to text categorization. They conclude that by using character n-grams, they are able to reliably categorize documents in a wide range of classification tasks, b) identification of reused, duplicated or plagiarized text (Barron-Cedeno and Rosso, 2009), c) malicious code detection (Abou-Assaleh et al., 2004) and d) a variety of linguistic tasks such as speech recognition (Jurafsky and James, 2000). The intuition behind all of the afore-mentioned approaches is common: phrases as a whole should carry more information than the sum of their individual components, extraction of which should lead to better textual representations and results.

Another aspect of n-gram analysis that should also be stressed out, is that infrequent n-grams are uninteresting, thus one only needs to keep track of n-grams that occur with frequency above a certain threshold – say at least m times. Furthermore, determining the correct value of n, i.e. the size of the sliding window that is to be used, when using word based n-gram analysis, is an area of experimentation on each particular domain of knowledge. For example, on the domain of plagiarism detection, Barron-Cedeno and, Rosso (2009), explain that low values for n appear to give the best results with a specific precision-recall trade-off. Values larger than 4, diminish the performance of the approach. A similar result is also pointed out by Furnkranz (1998), where word sequences of length 2 or 3 prove to be the most useful since larger sequences reduce classification performance.

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 article clustering that we are interested in, each sub-collection is called an article cluster and includes news articles that convey similarities between them and dissimilarities with others belonging to different news article clusters. Two generic categories of the various clustering methods exist: hierarchical and partitional. Typical hierarchical techniques 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 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. 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 (Zhao and Karypi, 2004) usually tries to minimize the average squared distance between points in the same cluster, i.e. if d1, d2,…, dn are the n documents and c1, c2,…, ck are the k clusters centroids.

In our previous work (Bouras and Tsogkas, 2010), we proposed a new clustering method, called W-kmeans, which improves the traditional k-means algorithm by enriching its input with WordNet hypernyms. The WordNet lexical reference system, organizes different linguistic relations into hierarchies/hypernyms (Is-a relation) and W-kmeans uses them as a preprocessing stage before the regular k-means algorithm. We extended this algorithm to the domain of user clustering (Bouras and Tsogkas, 2011), where we investigated how user clustering alone can affect the recommender's performance.

In this paper, we are focusing on incorporating n-gram extraction into our keyword extraction mechanism, which operates on news articles originating from the web, and explore the effect that this integration has on the article clustering process of W-kmeans. Moreover, the W-kmeans algorithm is adjusted so that it utilizes the n-grams information too, besides mere keywords. Furthermore, we are experimenting with different values of the n parameter so as to determine which gives the best results for our clustering approach as well as the target corpus (i.e. news articles from the web).

The rest of the paper is structured as follows: Section 2 gives an overview of the information flow that is followed within our approach. In section 3, the weighting approach for combining news articles clustering with n-grams extraction is presented. Next, in section 4, we outline our experimental framework as well as its results. Finally, section 5 gives some concluding remarks as well as some future work pointers.

2 FLOW OF INFORMATION

In Figure 1 we present the information flow of the suggested approach. Initially, at its input stage, our system fetches news articles generated by news portals from around the Web. 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 pro-

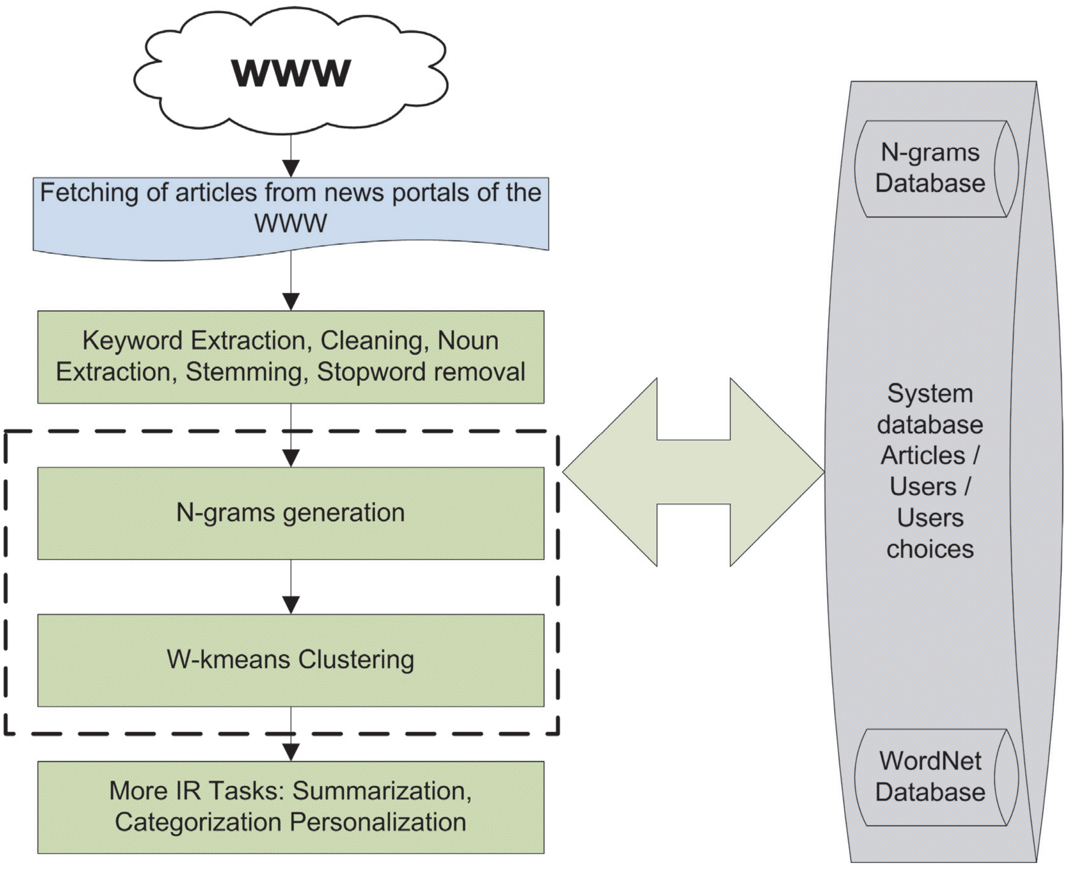
WWW
Fetching of articles from news portals of the WWW
Keyword Extraction, Cleaning, Noun Extraction, Stemming, Stopword removal
N-grams generation
W-kmeans Clustering
More IR Tasks: Summarization, Categorization Personalization
N-grams Database
System database Articles / Users / Users choices
WordNet Database

Figure 1: Flow of information.

cedures that follow.

A key process of the system as a whole, and probably as important as the IR algorithms that follow it, is text preprocessing that is applied on the fetched article's content and which results to the extraction of the keywords that each article consists of. At this analysis level, we apply some typical dimensionality reduction techniques, which include: stemming, stopword removal and filtering of low frequency words. In addition to the above, we also utilize:

- Feature selection / reduction where we attempt to select a subset of features that are most useful for the IR tasks that follow. This is achieved a) via POS tagging and noun extraction, and b) by pruning of words, appearing with low frequency throughout the corpus, which are unlikely to appear in more than a small number of articles.
- Feature generation / extraction where new features, are sought for representation. In our case, this is already achieved in a twofold manner: a) via the extraction of the text's nouns (through the process of POS tagging) and b) via WordNet generated hypernyms. In particular, for W-kmeans we enrich the article's text with new words utilizing the WordNet tree-like structure of hypernyms.

Keyword extraction, by using the vector space model, generates the term-frequency vector, describing each article as a 'bag of words' (words – frequencies vector) to the key information retrieval techniques that follow: article categorization, summarization and clustering. Previously, (Bouras and Tsogkas, 2010) we had successfully enhanced the efficiency of this 'bag of words' with the use of an external database, WordNet, in order to improve the results of the clustering algorithm.

Following keyword and n-gram extraction, we proceed with the document clustering procedure using our developed k-means variant, called W-kmeans. W-kmeans is a novel approach that extends the standard k-means algorithm using the external knowledge from WordNet hypernyms for enriching the "bag of words" used by the clustering k-means core process. Given the number of desired clusters, let k, partitional algorithms like W-kmeans, 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, are desired. It is important to note, however, that this clustering process is independent from the rest of the steps, meaning that it can easily be replaced by any other clustering methodology that works on a keyword level.

The contribution of the present work is highlighted within the dashed box of Figure 1. In essence, we are enhancing the feature generation process or the preprocessing stage by also extracting n-grams from the text and indexing them into the database. The process is similar to that of keyword extraction (which could actually be thought of as the trivial case of n-gram extraction with n=1): for each article and for values of n from 2 to 6, we identify the word n-grams of the input text and properly index them into our database. In this scenario, the overall similarity, either between two articles or between an article and a class, i.e. cluster, cannot be conveyed only in terms of keyword frequency / inverse document frequency metric (kf-idf), but rather as a combination of kf-idf and its n-gram counterpart metric, let's call it gf-idf, which will be described in detail in section 3. Note also that stemming is not being applied on the extracted n-grams since stemming techniques can be used in word-based systems but not in n-gram-based systems.

Summarizing, in the current work, W-kmeans operates using a) keywords, b) enriched hypernyms from keywords and c) previously extracted n-grams. The effect that these three aspects of the preprocessing steps have on the clustering process will be presented in sections 3 and 4.

3 N-GRAM WEIGHTING ANALYSIS

When keyword extraction completes its operation on each news article that is fetched from the web, a list of stemmed keywords is generated and stored in the database. For example, let's consider a given article that belongs to the domain or cosmology (categorized as 'science' by our classifier) and for which our preprocessing mechanism detects 18 valid keywords, as depicted in Table 1.

In the above list, the existing keywords are stemmed nouns and are presented in decreasing order given they absolute frequency of appearance in the text. Given the above data as well as the recorded data of the rest of the keywords in the database, one could easily determine the tf-idf weights of these keywords.

Table 1: Stemmed keywords with accompanying frequencies as extracted from an article.

ID	Keyword	Frequency
1	year	6
2	cosm	4
3	radiat	4
4	profess	4
5	mass	3
6	intens	3
7	event	3
8	Neuhauser	2
…	…	…
18	burst	2

In addition to the extracted keywords, our approach also extracts n-grams from the articles, with $2 < n <= 6$ and with frequency of appearance $fr > 1$. For example and for the same article, the extracted n-grams are depicted in Table 2. Note that given the nature of n-grams regarding their meaning and use in natural language, we have no choice but to take into consideration stopwords, too.

From the n-gram list of Table 2, one could infer that some n-grams like: "light years away" and "Professor Neuhauset" could be considered as good representatives of the particular domain to which this article belongs. Intuitively, the cluster that this article belongs to, should have those n-gram weights boosted.

Table 2: Top n-grams, with 2<n<6 as extracted from the same article.

ID	n-gram	Frequency
1	light years	4
2	the most	3
3	the past 3000	3
4	to have	3
5	light years away	3
…	…	…
24	Professor Neuhauser	2

Once we have the above vectors, we can come up with a weighting scheme utilizing both the individual keyword and the n-gram information. In this path, we are enhancing the keyword weighing equation described in Bouras and Tsogkas (2008), where the score of sentence a, Sa is given from equation (1).

$$Sa = \sum w_{k,i}(k_1 + k_2)k_3 * k_4 \qquad (1)$$

where k_1 expresses the impact of each particular keyword *i* that appears in the article's body, given its relative frequency compared with the total number of appearances in the database (tf-idf), k_2 expresses the impact of keywords appearing also in the article's title and factors k_3 and k_4 express the impact of the categorization and summarization subprocesses (as depicted in Figure 1) respectively (for a more in depth analysis of the factors k_1-k_4, we suggest that the user studies: Bouras and Tsogkas (2008)). The previously described weight was previously also utilized in our W-kmeans algorithm (Bouras and Tsogkas, 2010).

Enriching this weighting notion, we can assign weights to n-grams that are in each text, too by utilizing their 'tf-idf' statistics. More specifically, for each n-gram *j*, its weight could be expressed by its tf-idf frequency – let's call it *gf-idf* (gram frequency / inverse document frequency). This weight could be written as shown in equation (2)

$$W_{ngi} = gf - idfi = freq_j * \log\frac{N}{M} \qquad (2)$$

where *N* is the total number of articles in the database and *M* is the total number of articles containing the n-gram *i*.

Combining (1) and (2), we could express the total weight of each sentence, given its keywords and n-grams as in equation (3).

$$Si = A*(\sum w_{k,i}(k1+k2)k3*k4) + B*\sum w_{ngj} \qquad (3)$$

We can control and normalize the effect that the keywords and n-grams have on the clustering algorithm in a linear way by using the two parameters A, B, mentioned in (3), so that:

$$W'_{kwi} = W_{kwi} * A \qquad (4)$$

$$W'_{ngi} = W_{ngi} * B \qquad (5)$$

and:

$$A + B = 1 \qquad (6)$$

Determining the appropriate weights A and B is a matter of experimentation for the given dataset. As previously stated however, our analysis focuses only on the domain of news articles originating from the web so the follow-up weights and experimentation are based on this fact.

Summarizing the weighting heuristics described in this section, we could propose the algorithmic steps given in Algorithm 1 for clustering news articles using n-grams.

```
Algorithm clustering_based_on_ngrams
Input: articles, number of clusters, X
Output: cluster assignments
for each article a
   KW=fetch 20%most frequent k/ws of a
   NG = n_grams_extract(a, 1<n<X)
   WN = wordnet_enrich(a)
   clusters = kmeans(KW, NG, WN)
   // Weighting is on a sentence
  //level, based on (3)
   return clusters
```

Algorithm 2. Clustering news articles based on keywords, enriched hypernyms and n-grams

4 EVALUATION

For our evaluation, we tried to determine the effect of n-grams extraction and usage within the context of news article clustering via a series of offline experiments. In particular, we wanted to find out the improvement (if any) that the proposed approach would have on the efficiency of the W-kmeans algorithm. For extracting the n-grams from the news article's body, we utilized the n-gram extraction toolset from Zhang (2013) applying it on the documents of the used corpus.

Our corpus consists of 10000 news articles obtained from major news portals like BBC, CNN, etc. over a period of 5 months. Those articles were evenly shared among the 8 base categories featured by our system in order to avoid any biasese towards a specific news articles class. In order to determine the efficiency of the proposed approach in terms of our applied W-kmeans clustering method, we used the evaluative criterion of Clustering Index (CI), defined in Taeho and Malrey, (2007) as:

$$CI = \bar{\sigma}^2 / (\bar{\sigma} + \bar{\delta}) \qquad (7)$$

where $\bar{\sigma}$ is the average intra-cluster similarity and $\bar{\delta}$ is the average inter-cluster similarity. 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). As a result, this index can depict the coherence of the generated clusters.

For our first experiment, we tried to determine the best value for n, i.e. up to what size should the words window be when capturing grams. For this experiment, we arbitrarily set A=B=0.5 (i.e. giving the same weighting significance to both keywords

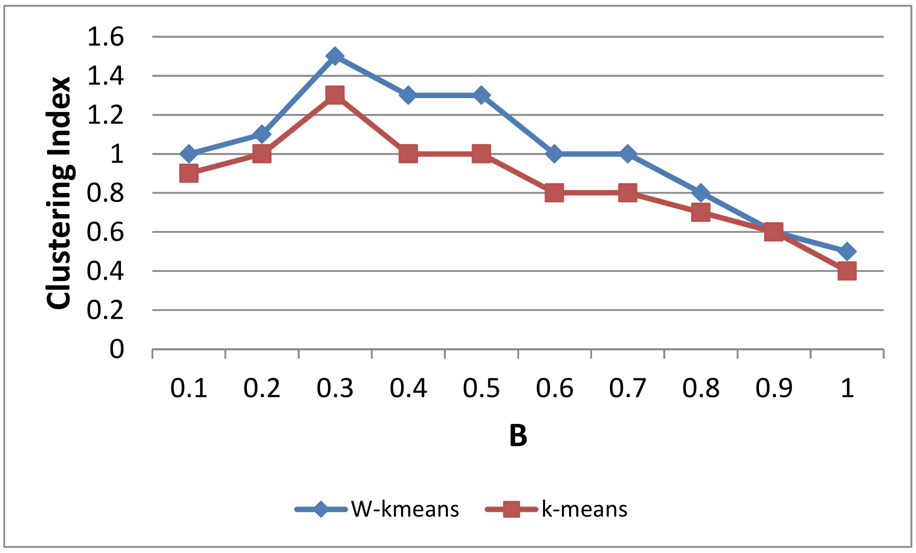

Figure 3: Performance results for various B values. Comparison of W-kmeans via regular k-means.

and n-grams) and tried different values of n where 2<=n<=6 as shown in Figure 2. For each n value, we repeated the clustering process 10 times with different starting cluster centres (10-pass experiment).

The results depicted in Figure 2, show that when n=3, i.e. we keep both 2-grams and 3-grams for weighting the sentences, the W-kmeans algorithm's performance is increased by an average of 0.3 regarding the clustering index of the generated clusters. This result is in accordance with the findings of Furnkranz (1998). For n=4, we still see a performance increase compared to the case where n-grams are not taken into consideration (i.e. n=1 in the graph of Figure 2).

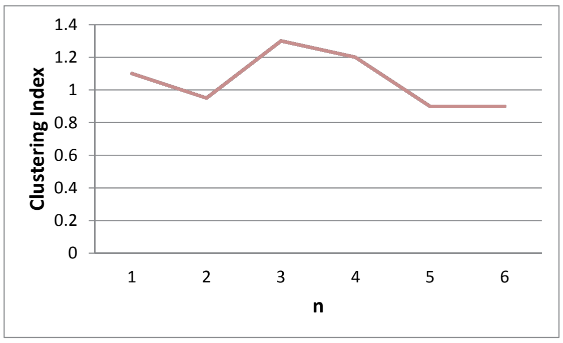

Figure 2: The effect of the parameter n when considering n-grams for W-kmeans clustering.

Increasing even further the window size seems to have a negative impact on the overall clustering index results. This can be explained as follows: larger window sizes mean that n-grams that randomly appear together in larger sequences are weighted much more that they should, a situation that probably has a negative impact on the overall weighting, given their random nature. Another interesting finding of this experiment is that for n=2, the results get slightly worse than when not using n-grams at all. A possible cause for that might be small statistical anomalies in the underlying data and also some n-grams extraction errors that we observed using the library from Zhang (2013) that are mainly due to the small window size.

For our second experiment, we tried to determine the best values for the keyword and n-gram weighting factors A and B. As such, we set n=3 based on our previous result, and run a 10 pass of our W-kmeans algorithm as well as of the standard k-means algorithm, with increasing values of B (given that from (6): A=1-B), while recording the CI scores of all the clustering passes. The average CI results are depicted in Figure 3.

As illustrated in Figure 3, the best CI results are obtained when B=0.3, i.e. when the n-gram weighting contributes by a factor of 30% while the rest is contributed by regular BOW weighting. What is more interesting, is that the performance deteriorates quickly as B increases and reaches the worst value of around 0.5 when only n-grams are taken into consideration.

This can be explained by the fact that not all the articles have outstanding and/or frequent n-grams, and as such, the more we take into consideration n-grams more than regular keywords, the less the performance will get on average. For example, a situation that an article does not have n-grams with frequency of at least 2 is commonplace for small arti cles.

Another observation we can easily make is that W-kmeans outperforms regular k-means significantly even when n-grams are thrown into the weighting equation. As a matter of fact, the results were consistently in favor on W-kmeans for any value of B that we tried. This is a good indication that the WordNet heuristic that W-kmeans applies pays off at any configuration. Nevertheless, we could not explain the almost identical CI results we got on average when B=0.9. Our point of view is that this is probably due to the underlying data and statistic anomalies of the used articles but further investigation and experiment should help us determine the cause.

5 CONCLUSIONS AND FUTURE WORK

In this paper we analyzed the implications of word n-gram extraction and use within the scope of our WordNet-enabled clustering algorithm, W-kmeans. We presented a means of utilizing both the classical BOW representation and the n-gram expansion in a tunable fashion, via the parameters A and B.

Our experimentation towards determining the best value for the n parameter revealed similar results with the existing literature, i.e. using 2-grams and 3-grams for the weighting process seems to yield better performance in terms of CI when it comes to clustering news articles from the web. Moreover, we saw that it is not enough to simply include n-grams into the equation. The weighting, given to the n-grams compared to regular keywords, as conveyed by the parameters A and B, is thus of great importance, not previously explored and an area that we feel is domain-specific and open for further experimentation. To summarize, the best results were obtained when n-grams affected the weighting process by around 30% and this was true for both the W-kmeans and the regular k-means clustering algorithms. Extending on the experimental results of our previous work (Bouras and Tsogkas 2010), we again came to the conclusion that W-kmeans outperforms regular k-kmeans, even with the n-gram enrichment process that this work describes in place.

For the future, we are planning on enriching the various components of our system with various and improved techniques, particularly for keyword extraction / enrichment and categorization. Moreover, we would like to execute a larger experiment for validating the presented in this work results; with a wider range and randomly selected news articles, we should be able to pin down the best values for n and B regarding the domain of clustering news articles from the web. We will also be focusing on creating suitable communication channels for delivering the article recommendations to the user's desktop or handheld device.

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

Abou-Assaleh T, Cercone N, Keselj V, Sweidan R, 2004, Detection of new malicious code using n-grams signatures. *In: Second annual conference on privacy, security and trust*. Fredericton, NB, Canada, pp 193–196.

Barron-Cedeno, A., Rosso. P, 2009. On automatic plagiarism detection based on n-grams comparisons. *In Proceedings of the European Conference on Information Retrieval, ECIR-2009*, pages 696–700.

Bouras, C., Tsogkas, V., 2010, W-kmeans: Clustering News Articles Using WordNet. *In Proceedings of KES (3).* pp. 379-388.

Bouras, C., Tsogkas, V., 2011, Clustering user preferences using W-kmeans. *In proceedings of SITIS 2011*, pp. 75 – 82.

Bouras, C., Poulopoulos, V., Tsogkas, V., 2008, PeRSSonal's core functionality evaluation: Enhancing text labeling through personalized summaries. *Data and Knowledge Engineering Journal, Elsevier Science*, Vol. 64, Issue 1, pp. 330 – 345.

Cavnar, W. Trenkle, J. 1994. N-gram-based text categorization. *In Proceedings of SDAIR-94.*

Cleary, J., Bell, T., and Witten, I., 1990. Text Compression. Prentice Hall.

Damerau, F., Apte, C. Weiss. S., 1994. Toward language independent automated learning of text categorization models. *In Proceedings SIGIR-94.*

Furnkranz, J., 1998, A study using n-grams features for text categorization. *Technical Report OEFAI-TR-98-30, Austrian Research Institute for Artificial Intelli* *gence.*

Jurafsky, D. James. H. M. 2000. Speech and Language Processing. *Prentice-Hall, Inc.*

Mahoui, M., Witten, I., Bray, Z. Teahan, W., 1999. Text mining: A new frontier for lossless compression. *In Proceedings of the IEEE Data Compression Conference (DCC).*

Moore, R., Lopes, J., 1999. Paper templates. In *TEMPLATE'06, 1st International Conference on Template Production*. SciTePress.

Robertson, A.M. Willett, P., 1999. Applications of n-grams in textual information systems*, Journal of Documentation*, 54(1), 48-69.

Smith, J., 1998. *The book*, The publishing company. London, 2nd edition.

Taeho, J. Malrey, L., 2007. The Evaluation Measure of Text Clustering for the Variable Number of Clusters. *Advances in Neural Networks* ISNN 2007 Volume 4492 pp. 871 - 879.

Zhao, Y. Karypi, G., 2004, Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering. *In Machine Learning*, v.55 n.3, p.311 – 331.

Zhang, L., N-Gram Extraction Tools, accessed: 1-1-2013, *http://homepages.inf.ed.ac.uk/lzhang10/ngram.html*