Clustering Documents using the 3-Gram Graph Representation Model

John Violos
Dept of Electrical and Computer Engineering
National Technical University of Athens,
9, Heroon Polytechniou Str, 15773, Zografos, Greece
+30 2107722568
violos@mail.ntua.gr

Konstantinos Tserpes
Dept of Informatics and Telematics
Harokopio University of Athens
9, Omirou Str., 17778, Tavros, Greece
+302109549413
tserpes@hua.gr

Athanasios Papaoikonomou
Dept of Electrical and Computer Engineering
National Technical University of Athens,
9, Heroon Polytechniou Str, 15773, Zografos, Greece
+30 2107722568
tpap@mail.ntua.gr

Magdalini Kardara
Dept of Electrical and Computer Engineering
National Technical University of Athens,
9, Heroon Polytechniou Str, 15773, Zografos, Greece
+30 2107722568
nkardara@mail.ntua.gr

Theodora Varvarigou
Dept of Electrical and Computer Engineering
National Technical University of Athens,
9, Heroon Polytechniou Str, 15773, Zografos, Greece
+30 2107722484
dora@telecom.ntua.gr

ABSTRACT

In this paper we illustrate an innovative clustering method of documents using the 3-Gram graphs representation model and deducing the problem of document clustering to graph partitioning. For the latter we employ the kernel k-means algorithm. We evaluated the proposed method using the Test Collections of Reuters-21578, and compared the results using the Latent Dirichlet Allocation (LDA) Algorithm. The results are encouraging demonstrating that the 3-Gram graph method has much better Recall and F_1 score but worse Precision. Further changes that will further improve the results are identified.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Clustering; H.2.8 [Database Applications]: Data mining; E.1 [Data Structures]: graphs and networks; G.2.2 [Graph Theory]: graph algorithms

General Terms

Algorithms.

Keywords

N-Gram graph, graph partitioning, graph Comparison, Text Clustering,

1. INTRODUCTION

The volume of documents available in the web dictates the need for developing mechanisms that automatically categorize them based on the topics they are related to. The applications that would benefit from such a tool are numerous, however the task itself is extremely challenging because of the need to use content to derive –unknown so far- categories which are often contextually defined. This task is largely known as "document clustering" and several approaches exist for tackling it. In this work, we focus on a novel such approach which yields encouraging results. This approach relies on the innovative combination of a document representation model and a clustering algorithm.

PCI2014, October 02 - 04 2014, Athens, Greece

The suggested document representation model is an NLP method called N-Gram graphs originally introduced by Giannakopoulos et al. [8]. An N-Gram graph models the frequency that two N-Grams (consecutive sequences of characters) appear in adjacent positions in the document. N-Grams are denoted as nodes, whereas adjacency is denoted with an edge, the weight of which corresponds to adjacency frequency. The transformation of a model to a graph is expected to capture notions such as styles of the use of language which often encompass valuable contextual information. At the same time, the use of N-Grams instead of words as the core elements of the model, allows for a better resilience in grammatical and syntactical errors, neologisms, abbreviations, language switch and other such phenomena [7] which are commonly used to documents in the web (non-curated text).

The identification of the commonalities in the documents can now shift from purely content comparison to graph comparison investing in the latent contextual features incorporated in the graph model. Thus, the problem of text clustering is deduced to graph partitioning. For that purpose we employ the kernel k-means algorithm [5] which effectively splits the document set graph to k subgraphs with the objective to maximize the sum of weight edges inside the clusters and minimize the sum of weight edges among the clusters. By comparing each individual document's graph model to the k subgraphs and calculating a similarity score we can place the documents to the group to which it is more similar.

To further back the claim that the combination of N-Gram graphs and kernel k-means achieves an effective and comparatively better than the current art document clustering, we perform a set of experiments. Using a dataset obtained by Reuters (Test Collections of Reuters-21578), we perform and present how the proposed mechanism outperforms other similar solutions.

In what follows the document is structured as such: The technologies upon which this research had relied upon are presented in Section 2. In Section 3 provides details about the representation model and the clustering algorithm. Section 4 demonstrates the experimental results and the comparison with other document clustering methods. Finally, in Section 5, we list the main conclusions regarding this work.

2. RELATED WORK

Our method is based on the use of N-Grams graphs [8] which can represent any kind of text as a graph. Different assumptions may result in different graph representation models. One differentiation results from the use of unweighted graphs in which an edge

depicts the occurrence of adjacency between two N-Grams in the document. Other variations may exist with the use of directed graphs, though –again- in this work we employ undirected graphs. Further variations are created if one considers the generation of N-Grams, e.g. using a sliding window of n characters with step one or with step n. Since we are interested for all the possible combinations of N-Grams in a document we go with the first approach. For each node we make N edges which join the N following nodes. For example if we use 3-Gram graphs for each node we make three edges which join the current node with the next three nodes. We repeat this procedure from the first text characters until the end.

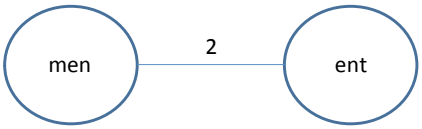

Figure 1: A part of the 3-Gram graph generated by the phrase: A complementary document. Notice the two **appearances of the sequence "ment" which contains two 3-Grams: "men" and "ent"**

Once we have the document models for each document combined in one main graph, we employ a graph partitioning algorithm to identify the clusters. Typically the graph partitioning problems are NP-hard. That's why we use heuristic and approximation algorithms. It is important to state that even though there are many options for heuristic algorithms not all of them solve the problem efficiently.

A well-known graph partition algorithm is the Kernighan–Lin algorithm [11] which exchanges nodes between the partitions using a metric of internal and external cost. Newman and Girman [12] suggest the use of betweeness as a metric to make the partitions. This algorithm performed well for –the rather limited number of- less than 10000 nodes.

K-Means [10] is a popular algorithm which partitions n observations into k clusters. In this method each observation belongs to the cluster with the nearest mean. The observations are d-dimensional real vectors. K-Means algorithm has a limitation: it cannot partition efficiently non-linear observations. Consequently in many occasions it is not best suited for real life datasets.

An extension of the K-Means algorithm which efficiently tackles the previous limitation using a kernel function is the weighted Kernel K-Means Clustering [6]. Dhillon et al. have stated that the general weighted kernel k-means objective is mathematically equivalent to a weighted graph partitioning objective [5] and have proposed algorithms which can partition a graph using Kernel based methods [4]. We opt to employ a kernel based algorithm for our method as the most appropriate solution for the scale and requirements of the investigated problem.

To estimate the similarity between two N-Gram graphs we used the Containment Similarity [1] criterion which expresses the proportion of edges of the first graph that are shared with the second graph. We chose this metric because in the stage of graph comparison we use unweighted graphs and it includes fewer computations than the other metrics. The number of comparisons is large in the proposed methods raising an issue of effectiveness.

Many solutions have been proposed from domains like Information Retrieval and Data mining in order to automatically categorize documents, with the most well-known and efficient possibly being PLSA [9] and LDA [3]. PLSA is based on the co-occurrence of words in the same document and using a statistical technique makes some latent variables which represent the observed words. LDA is a more accurate method than PLSA. LDA uses Dirichlet distribution and produces a reasonable mixture of topics for each document. In the end of our evaluation we also used the LDA clustering method to the same data set in order to compare the results of our 3-Gram graphs to LDA.

There are specific extrinsic clustering evaluation metrics [2] like Precision BCubed, Recall BCubed and F-measure which are used in order to estimate the quality of text clustering algorithms. These methods make the comparison between a clustering system and a gold standard which we have.

3. MODEL & APPROACH

3.1 Document Representation Model

An n-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram"; size 3 is a "trigram". Larger sizes are sometimes referred to by the value of n, e.g., "four-gram", "five-gram", and so on. In this model we used all the contiguous sequence of three letters which can be derived from any set of documents we want to cluster. We used these trigrams as the core components of the model, i.e. the 3-Grams graph.

The n-gram graph is a graph G = $\{V^G, E^G\}$ which has n-grams as its vertices $v^G \in V^G$, and the edges $e^G \in E^G$ connecting the n-grams indicate proximity of the corresponding vertex n-grams. If the n-grams are found to be neighbors more than once in the original text, which is usually the case, one could assign a distribution of distances to each edge to use the distance information. The formal definition is as follows.

Definition 3.1. If S = $\{S_1, S_2, \ldots\}$, $S_k \neq S_l$, for k ≠ l , k, l ∈ ℕ is the set of distinct n-grams extracted from a text T^l , and S_l is the ith extracted n-gram, then G is a graph where there is a bijection (one-to-one and onto) function f : S → V.

After we construct the N-grams nodes from the text document we join some nodes of the graph to make the edges. Each N-Gram node has an edge with the following D_{win} Ngrams nodes which exist in the text. As example if we have D_{win}=3 and N=3 the node "exa" from the text "example" is joined with the nodes "amp", "mpl", "ple". A future research is to experiment with other parameters and threshold D_{win} for creating the edges.

The edges E are assigned weights of $c_{i,j}$, where $c_{i,j}$ is the number of times a given pair S_i, S_j of n-grams happen to be neighbors within some distance D_{win} (in characters for character n-grams) of each other or within a function of the given distance D_{win}. Since probably not all distances are of importance, and thus two n-grams neighboring by 150 characters probably have no actual relation, we take into account only a window around S_i in the original text to determine which S_j deserves our attention. The vertices v_i, v_j corresponding to n-grams S_i, S_j that are located within this parameter distance D_{win} are connected by a corresponding edge e ≡ $\{v_i , v_j\}$.

Following this method of representation, we have reached a point where we have kept some information for a determined n-gram length n and parameter distance D_{win}. It is nontrivial, though, to

choose a single {n, D_{win}} pair that can be optimal for n-gram extraction, independent of the text. In this research the combination n=3 and $D_{win} = 3$ was used in order to balance the size of the resulting graphs based on the number of documents at hand.

3.2 Partitioning Algorithm

The approach is initiated with the transformation of a set of documents to their respective 3-Gram graphs. A new graph is then created by merging the initial set. E.g. for a set of 100 documents 101 3-Gram graphs are created in total: 100 for each document and one more by concatenating these hundred graphs.

Every node represents a 3-Gram and any directed edge represents that the starting node (3-Gram) is in front of the ending node (3-Gram) in a framework of three characters. At this point of the algorithm the graph of every single document is directed and unweighted. That is, it doesn't matter how many times two 3-Grams are adjacent in a document. The reason for this selection is that documents contribute equally to the algorithm. Taking into consideration the length of a document (which most likely will yield bigger weights) would result in an uneven contribution in the definition of clusters from different documents which would lead to wrong conclusions since a cluster may be equally defined by a long or short document. On the other hand the 3-Gram graph which combines all the documents is a weighted graph because the edge weights are needed in order to partition it as it is explained in the following paragraphs.

Having a 3-Gram graph which represent the set of all the documents the algorithm partitions it in k partitions. Each partition has some principal nodes, as well as all the directed adjacent edges of principal nodes and the ending nodes of these edges. To illustrate this point if one partition would have the principal nodes {A, B, C} it would also include all the edges which have as staring node the nodes A, B and C so it would have the edges {(A,B), (A,D), (A,E), (A,C)} consequently this partition would include the nodes: {A, B, C, D, E} and the edges: {(A,B), (A,D), (A,E), (A,C)}

Our aim is to Cluster large volumes of textual documents based on their topics. A graph partitioning algorithm is used just as a component of the method we propose. We use the graph partitioning algorithm to construct the N-gram graph which corresponds and represents each cluster. These cluster N-gram graphs are compared with the document N-gram graphs in order to cluster the documents in the right topic clusters.

To partition the 3-Gram graph which represents the set of all the documents the algorithm of Kernel based graph Clustering was used (3.1)

Weighted Kernel k-means(K, k, w, t_{max}, $\{\pi_c^{(0)}\}_{c=1}^k$, $\{\pi_c^{(0)}\}_{c=1}^k$)

Input: K: kernel matrix, k: number of clusters, w: weights for each point, t_{max}: optional maximum number of iterations, $\{\pi_c^{(0)}\}_{c=1}^k$: optional initial clustering

Output: $\{\pi_c\}_{c=1}^k$: final clustering of the points

1. If no initial clustering is given, initialize the k clusters $\pi_1^{(0)}, \dots, \pi_k^{(0)}$ randomly. Set t = 0.

2. For each row i of K and every cluster c, compute

$$d(i, m_c) = K_{ii} - \frac{2 \cdot \sum_{j \in \pi_c^{(t)}} w_j \cdot K_{ij}}{\sum_{j \in \pi_c^{(t)}} w_j} + \frac{\sum_{j \in \pi_c^{(t)}} w_j \cdot w_l \cdot K_{ij}}{(\sum_{j \in \pi_c^{(t)}} w_j)^2}$$

3. Find $c^*(i) = argmin_c d(i, m_c)$, resolving ties arbitrarily. Compute the updated clusters as

$$\pi_c^{t+1} = \{i : c^*(i) = c\}$$

4. If not converged or t_{max}, > t, set t = t + 1 and go to Step 3; Otherwise, stop and output final clusters $\{\pi_c^{(t+1)}\}_{c=1}^k$

Algorithm (3.1) Kernel based graph Clustering

A kernel matrix can be calculated using one of the following objectives of Table 1.

Table 1: Common graph clustering objectives with corresponding weights and kernels given the affinity matrix A

Objective	Node Weights	Kernel Matrix
Ratio Association	$1 \forall nodes$	K = σI + A
Ratio Cut	$1 \forall nodes$	K = σI –D + A
Normalized Cut	Deg. of node	K=$\sigma D^{-1} + D^{-1}AD^{-1}$ A

The affinity matrix A is |V|×|V| whose entries represent the weights of the edges (an entry of A is 0 if there is no edge between the corresponding vertices).

The Algorithm 3.1 calculates for each node i and each cluster m_c the cost $d(i, m_c)$ and clusters the node i to the cluster which produce the minimum cost we can make repeat these steps until no one node would change cluster or for a specific number of iterations t_{max}.

Eventually we have the k n-gram graphs which represent the k clusters. From now on the weights of the graphs will not be used and for each cluster will have an unweighted directed 3-gram graph which will be compared with the unweighted directed 3-gram graph of each document.

For the graph comparison we used the Containment Similarity (3.1), which expresses the proportion of edges of graph G_{ti} that are shared with graph G_{Tp} . Assuming that G is an n-gram graph, e is an n-gram graph edge and that for the function μ(e,G) it stands that μ(e,G) = 1, if and only if e ∈ G, and 0 otherwise, then:

$$CS(G_{ti}, G_{Tp}) = \sum_{e \in G_{t_i}} \frac{\mu(e, G_{T_p})}{\min(|G_{t_i}|, |G_{t_p}|)} \quad (3.1)$$

Where |G| denotes the size of the n-gram graph G, in our model the number of edges the graph contains.
As G_{t_i} we use the cluster 3-Gram graph and as G_{t_p} the document 3-Gram graph.
The Containment Similarity gives the proportion of a document to belong in any cluster. So we have a soft membership for each document. If we want a document to belong in only one cluster then we choose the cluster that produces the highest Containment Similarity. In case we want a document to belong in m number of documents we choose the m clusters which have the higher Containment Similarity.

4. EVALUATION

We used the Reuters-21578 Text Categorization Test Collection as the evaluation dataset to examine the performance of our model. This document set appeared on the Reuters newswire in 1987. The documents were assembled and indexed with categories by personnel from Reuters Ltd. This collection includes many documents which belong to zero up to twenty-nine categories. A clean up of the dataset resulted in the discard of documents that do not belong to any category as well as many documents which their interpretation was particularly difficult. In total 18457 documents belonging to 428 categories were used.

Some categories were empty so they weren't used as well. The size of the remaining categories ranged significantly, however this does not affect the efficiency of the proposed method. This is a property which enables the method to be employed even for text from microblogging services like tweets and even newspaper articles. This document set had a variety of small, medium and large documents but neither huge nor tiny. All documents were written in the English language.

The complete method was implemented using Java SE. The program received the Reuters dataset as input and made the 3-Gram graph of all documents. Afterwards it partitioned the graph in 428 partitions. Each set had some nodes and all their adjacent edges. Very few sets were empty and didn't have a 3-Gram graph to represent them. This issue slightly affected the results because there are some categories which have very few documents. Thence the 3-Gram graphs for each document was created and each was compared to the 428 3-Grams graphs which represent the clusters/partitions. A soft membership was considered for each document to all the clusters, i.e. a probability for each document to belong to any specific cluster was estimated. A document was included in a cluster if its graph presented high similarity to a cluster's/partition's graph.

The same dataset was used for a parallel thread of experiments, this time using LDA in order to compare the results and have a measure of performance of our proposed algorithm. LDA is one of the most popular documents clustering method. The Machine Learning for Language Toolkit (Mallet) was used which is a Java-based package and implements LDA.

As stated above, for comparing the two approaches we used three metrics that are commonly employed in order to estimate the precision of a clustering method, namely Precision, Recall and F-measure. In the evaluation of the method we used retrieval / classification measures because our aim is to evaluate the document clustering in each topic cluster. Each is defined as presented below.

$$\text{Precision} = Avg_e[Avg_{e',C(e)\cap C(e')\neq 0}[Multiplicity\ precision(e,e')]] \quad (4.1)$$

$$\text{Recall} = Avg_e[Avg_{e',L(e)\cap L(e')\neq 0}[Multiplicity\ recall(e,e')]] \quad (4.2)$$

$$\text{Multiplicity Precision}(e,e') = \frac{Min(|C(e)\cap C(e')|,|L(e)\cap L(e')|)}{|C(e)\cap C(e')|} \quad (4.3)$$

$$\text{Multiplicity Recall}(e,e') = \frac{Min(|C(e)\cap C(e')|,|L(e)\cap L(e')|)}{|L(e)\cap L(e')|} \quad (4.4)$$

$$\text{F-measure} = \frac{2\cdot Recall(L,C)\cdot Precision(L,C)}{Recall(L,C)+Precision(L,C)} \quad (4.5)$$

where e and e' are two documents, L(e) denotes the set of categories (derived from the gold standard) and C(e) the set of clusters associated to e. The item precision represents how many items in the same cluster belong to its category. Symmetrically, the recall metric is associated to one item represents how many items from its category appear in its cluster. F-measure is a metric which combines Precision and Recall.

4.1 Results

The results indicated that the 3-Gram graph method has much better Recall but worse Precision. The reason is that 3-Gram graph can recognize the clusters which have many documents and classify many documents to them. On the other hand LDA makes many small clusters which have few documents and does not recognize the big clusters. As example LDA makes a cluster containing four documents, two of them is correct that clustered together so we have Precision is 2/4 =50% but the gold standard contains 100 documents in the same cluster so the Recall is 2/100= 2% Table 2 sums up the results of the comparison between 3-Gram graph to gold standard and LDA to gold standard using the equations (4.1) to (4.5)

Table 2: Evaluation of experimental results using three metrics for both methods.

	Precision	Recall	F-measure
3-Gram graph	0.2870524	0.2045877	0.2419
LDA	0.5757706	0.025551	0.0498

In many cases in which documents were clustered wrongly this could have been fixed if they had been clustered according to the second or the third higher comparison value. This means that 3-Gram graph can recognize in most of the times the right cluster of a document in its top cluster results in contrast to LDA which made only small clusters which were like broken parts of the gold standard clusters

It is very positive for our method that Precision and Recall are close and that F-measure of 3-Gram graph clustering is much higher than LDA. LDA could understand if some documents had many common words so it clustered them together but as we can see this was not enough to have high Recall. 3-Gram graph clustering has a different criterion for each cluster and document had a representative graph. If the cluster graph and document graph are similar then we expect that the documents can be similar too. The experimental results saw this criterion is more accurate.

We executed the java program in a general-purpose computer for less than four hours. We made only 400 iterations of the K-Means clustering algorithm (3.1) because we observed that it converged quickly. There are many suggestions about the starting partition of the 3-Gram graph which can accelerate the converge we just used a variation of the coarsening method of A Fast Kernel based Multilevel Algorithm for graph Clustering [7]. Methods to make better starting partitions can be researched in future work.

Precision is more important than recall in this setting. It is better to have pure clusters. That is to say it is preferable to make clusters which have only the right clustered documents than the discovered clusters to compound all the right documents together with wrong clustered documents. So future research will be made to improve the precision.

Using another kernel function or partition method of the 3-Grams graph it may product better results. In the future we will try many more candidate methods. Moreover we can try to remove all the stop words and make stemming before the graph construction. Instead of 3-Grams we can use 4-Grams, this will have as a result a bigger graph which represent all the documents so the partition time will last more time but if we have a bigger graph then we would have more nodes and edges to represent each cluster and have better accuracy.

5. CONCLUSIONS

Clustering documents using the 3-Gram graph representation model is an innovative method which can categorize efficiently thousands of documents and can estimate the extent to which a document belongs to any cluster. It is language independent and incorporates state of the art techniques of information retrieval and data mining. It can have better results than other popular document clustering methods according to our experiments and we hope that further research will meliorate it even more.

6. ACKNOWLEDGMENTS

This work has been supported by the SUPER project and has been partly funded by the EU Seventh Framework Programme, theme SEC-2013.6.1-1: The impact of social media in emergencies – Capability Project under Contract No. 606853.

7. REFERENCES

[1] Aisopos, F. et al. 2012. Content vs. Context for Sentiment Analysis: A Comparative Analysis over Microblogs. *Proceedings of the 23rd ACM Conference on Hypertext and Social Media* (New York, NY, USA, 2012), 187–196.

[2] Amigó, E. et al. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*. 12, 4 (Aug. 2009), 461–486.

[3] Blei, D.M. et al. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3, (Mar. 2003), 993–1022.

[4] Dhillon, I. et al. 2005. A Fast Kernel-based Multilevel Algorithm for Graph Clustering. *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (New York, NY, USA, 2005), 629–634.

[5] Dhillon, I. et al. 2004. *A unified view of kernel k-means, spectral clustering and graph cuts*.

[6] Dhillon, I.S. et al. 2004. Kernel K-means: Spectral Clustering and Normalized Cuts. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2004), 551–556.

[7] Giannakopoulos, G. et al. 2012. Representation models for text classification: a comparative analysis over three web document types. *2nd International Conference on Web Intelligence, Mining and Semantics, WIMS '12, Craiova, Romania, June 6-8, 2012* (2012), 13.

[8] Giannakopoulos, G. et al. 2008. Summarization System Evaluation Revisited: N-gram Graphs. *ACM Trans. Speech Lang. Process.* 5, 3 (Oct. 2008), 5:1–5:39.

[9] Hofmann, T. 1999. Probabilistic Latent Semantic Analysis. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (San Francisco, CA, USA, 1999), 289–296.

[10] Kanungo, T. et al. 2002. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 24, 7 (Jul. 2002), 881–892.

[11] Kernighan, B.W. and Lin, S. 1970. An Efficient Heuristic Procedure for Partitioning Graphs. *Bell System Technical Journal*. 49, 2 (Feb. 1970), 291–307.

[12] Newman, M.E.J. and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical Review E*. 69, 2 (Feb. 2004), 026113.