Evaluation of Text Clustering Algorithms with N-Gram-Based Document Fingerprints

Javier Parapar and Álvaro Barreiro

IRLab, Computer Science Department
University of A Coruña, Spain
{javierparapar,barreiro}@udc.es

Abstract. This paper presents a new approach designed to reduce the computational load of the existing clustering algorithms by trimming down the documents size using fingerprinting methods. Thorough evaluation was performed over three different collections and considering four different metrics. The presented approach to document clustering achieved good values of effectiveness with considerable save in memory space and computation time.

1 Introduction and Motivation

Document's fingerprint could be defined as *an abstraction of the original document that usually implies a reduction in terms of size*. In the other hand data clustering consists on the partition of the input data collection in sets that ideally share common properties. This paper studies the effect of using the documents fingerprints as input to the clustering algorithms to achieve a better computational behaviour.

Clustering has a long history in Information Retrieval (IR)[1], but only recently Liu and Croft in [2] have demonstrated that cluster-based retrieval can also significantly outperform traditional document-based retrieval effectiveness. Other successful applications of clustering algorithms are: document browsing, search results presentation or document summarisation.

Our approach tries to be useful in operational systems where the computing time is a critical point and the use of clustering techniques can significantly improve the quality of the outputs of different tasks as the above exposed.

Next, section 2 introduces the background in clustering and document representation. Section 3 presents the proposed approach. In section 4 is explained the evaluation methodology and results are reported in section 5. Finally conclusions and future work are presented in section 6.

2 Background

Traditional clustering algorithms could be classified in two groups: partitional like all the k-means [3] clustering family and hierarchical algorithms both agglomerative (single-link, complete-link, average-link, etc.) and divisive, although

last ones had not too much impact. The main problem of applying clustering techniques in order to improve the performance of real IR systems is the computational cost. Traditionally the clustering algorithms have a high computational complexity in terms of space and time [4]. Hierarchical methods are typically $O(n^2 \log(n))$ (n is the number of documents in the collection) in time and $O(n^2)$ in space. K-means is $O(k \times i \times n)$ in time and $O(k + n)$ in space, where k is the number of clusters and i the number of iterations, but the quality of the clustering results is quite dependent of the initial random cluster seeding. Meanwhile hybrid approaches as Buckshot [5] are $O(k \times i \times n + n \log(n))$ and $O(ni)$ in time and space respectively, but these methods become inefficient in practice. So when the input is a huge documents collection, e.g. the web, the spent time and the needed memory space to compute the clusters are not admissible.

Although data clustering using document fingerprints was not too much explored, we want to remark two works. Broder et al. in [6] explored the use of a kind of fingerprints called *shingles* to perform syntactic clustering of web documents efficiently. Puppin and Silvestri [7] evaluated the use of the shingle based method described by Broder et al. with a classical k-means algorithm in order to get an efficient and suitable collection partition in the context of a distributed retrieval model.

In this paper we propose the use of winnowing fingerprints for the clustering task. Winnowing fingerprints were introduced by Schleimer et al. in [8], the algorithm was presented with the objective of plagiarism detection, but the fingerprint construction guarantees also a set of theoretical properties in terms of fingerprint density and sub-string matching detection. Recently Parapar and Barreiro [9] presented the use of these fingerprints in the context of clustering.

In order to compare the performance of the selected fingerprint method we chose three other document representations: term frequency, mutual information and a designed fixed-size fingerprint that we coined as *n-fingerprint*.

Term Frequency (TF). The term frequency representation of a document d in a collection of m terms was computed as follows:

$$TF(d) = [tf(d, t_1); tf(d, t_2); tf(d, t_3); \ldots; tf(d, t_m)]$$

where $tf(d, t_m)$ is the relative frequency of the term t_m in the document d. In order to compute the similarity between two term frequency vectors standard cosine distance was used.

Mutual Information (MI). The mutual information vector of a document d in collection of m terms and D documents was computed as follows:

$$MI(d) = [mi(d, t_1); mi(d, t_2); mi(d, t_3); \ldots; mi(d, t_m)]$$

where

$$mi(d, t) = \log \frac{\frac{tf(d,t)}{N}}{\frac{\sum_i^D tf(d_i,t)}{N} \times \frac{\sum_j^m tf(d,t_j)}{N}} \tag{1}$$

$N = \sum_i \sum_j tf(d_i, t_j)$ and $tf(d, t)$ is the frequency of the term t in the document d. The cosine distance was also used as similarity measure.

N-Fingerprint (NFP). The idea behind this representation is to construct a reduced fixed-size fingerprint representation that enables very fast cluster-

ing but with more valuable information that simple MD5 fingerprints [10]. N-fingerprints are representations of the documents as n-gram frequency deviations from the standard frequency in a given language. For a given n the N-fingerprint of a document d was computed as follows:

$$NFP(d) = [nfp(d, n_1); nfp(d, n_2); nfp(d, n_3); \ldots; nfp(d, n_m)]$$

where

$$nfp(d, n_i) = reff(n_i) - f_d(n_i) \tag{2}$$

Each of the elements of the n-fingerprint of a document represents the deviation of frequency of the corresponding n-gram n_i, where $f_d(n_i)$ is the frequency of the n-gram n_i in the document d and $reff(n_i)$ is the standard frequency of the n-gram n_i in the given language. In order to compute document similarity was also used the cosine distance.

3 Clustering with Winnowing Fingerprints

The idea presented in [8] was to introduce a new kind of fingerprinting able to detect local matches (partial copies) in the task of plagiarism and version detection. We are going to use the winnowing algorithm [8] quite straightforward with some minor changes that are presented next.

Winnowing Algorithm. One of the advantages of the winnowing algorithm for hashing selection is the trade-off between the fingerprint length and the shortest matching string to be detected, establishing theoretical guarantees. Let t the threshold that guarantees *(a)* that any match of strings *longer or equal than t is detected*, and let n another threshold that guarantees *(b)* that any match of strings *shorter than n is not detected*. The parameters t and n are chosen by the user being the n value the n-gram size. Given any list of hashes $h_1, h_2, ...h_k$, if $k > t - n$ then at least one of the $h_{1<i<k}$ should be chosen in order to guarantee the detection of all matches longer or equal than t. To achieve this the next selection process was proposed.

Let $w = t - n + 1$ be the window size and let be $h_1, h_2, ...h_k$ the whole sequence of hashes result of hashing all the n-grams in which the document text was decomposed. Each position $1 \leq i \leq k - w + 1$ defines the start of a window (sub-list) of hashes $h_i, h_{i+1}, ...h_{i+w-1}$, therefore in order to guarantee the condition *(a)* is necessary and sufficient to select one hash value for every window to compose the fingerprint. The condition *(b)* is guaranteed by the fact of choosing n-grams of size n (see in [8] the proof). In order to achieve this, the process defined by the authors was the next:

In each window select the minimum hash value. If there is more than one hash with the minimum value select the rightmost occurrence. Now save all selected hashes as the fingerprints of the document.

Let's see an example[1] in figure 1. First, the document text is preprocessed (2). After this, the string is decomposed in n-grams (3). For each n-gram of the text a hash value is computed (4). Over the list of hash values the list of

[1] The Police: De Do Do Do, De Da Da Da. In: Zenyatta Mondatta, 1980.

Fig. 1. Example of the construction of winnowing fingerprint ($w = 4$, $n = 4$, $t = 7$)

1. `De do do do, de da da da`
2. `dedododododedadada`
3. `dedo edod dodo odod dodo odod dode`
   `oded deda edad dada adad dada`
4. 59 62 39 67 39 67 29 57 45 48 53 46 53
5. (59,62,**39**,67) (62,39,67,**39**) (39,67,39,67) (67,39,67,**29**) (39,67,29,57) (67,29,57,45) (29,57,45,48) (57,**45**,48,53) (45,48,53,46) (48,53,**46**,53)
6. 39 39 29 45 46

moving windows is constructed (5), in each window is selected the minimum value, if any tie occurs the rightmost hash is selected. The list of selected hashes is retained in a multi-set as document representation. Some remarks: in the selection process, a hash occurrence (the same hash value and position) can not be selected more than once (see in step 5 for example that in the third window the second apparition of `39` was not chosen because is the same hash chosen in the previous window). This last condition produces that the number of selected hashes can be smaller than the number of windows, but it still maintains the desired guarantees. In our case the positions of the hashes are just used in the selection process, they are not recorded in the document representation because in the clustering process positional information was not used.

Expected Density and Hashing Function. It can be proved [8] that, using a non-collision hash function, the selection method guarantees an expected fingerprint density $d = \frac{2}{w+1}$ (the fraction of hashed selected from among all the computed ones). This density represents then a trade-off between t and the fingerprint size, i.e., short document fingerprints will only guarantee the detection of long string matches between documents. Also this density property allows the adaptation of the winnowing fingerprint size to the clustering domain, the collection size and the nature of the documents.

Another key point is the election of the hashing function. In our case the fingerprint memory use is a hot point so the reduction of the hashes size was very desirable. We chose a 32 bits hashed function based on a hash table of n-grams, this particular function avoids collisions in our collections and set-ups and preserves the theoretical guarantees of the algorithm while allowing us a considerable memory save. In [8] it also was considered an efficient hashing for large n-gram sizes computation based on rolling functions [11] with the advantage of computing them incrementally.

Similarity Measures. After applying the winnowing algorithm each document is a multi-set of hash values. Each multi-set $W(d)$ has different size depending on the document d size and could have repeated hash values. For this reason each document is now represented as a multi-set.

In order to cluster the collection we have to adopt a suitable similarity measure. For this purpose we chose to adapt the Jaccard Coefficient computed as showed in equation 3 to multi-sets.

$$Sim(d_i, dj) = \frac{|W(d_i) \cap W(d_j)|}{|W(d_i) \cup W(d_j)|} \tag{3}$$

The intersection and union were defined as: $W(d_i) \cup W(d_j)$ is the multi-set composed of every element of $W(d_i)$ and $W(d_j)$ repeated son many times as its maximum presence in $W(d_i)$ or $W(d_j)$. $W(d_i) \cap W(d_j)$ is the multi-set composed of those hashes present in $W(d_i)$ and $W(d_j)$ repeated as many times as in the multi-set where they have lower cardinality. For further reference we shall also define here the join of $W(d_i) \uplus W(d_j)$ as the multi-set composed of all the elements of $W(d_i)$ and all the elements of $W(d_j)$.

Another problem was the computation of the cluster centroid when working with multisets. Centroid computation over sets is not trivial. One alternative could be using the union of the cluster documents as cluster centroid, but it will have a very high cardinality and will not consider the frequency. Thus, when the cluster contains documents with very few elements in common the centroid will not be representative. On the other hand intersection is not suitable neither because it is probable to obtain the empty set when working with several documents.

In this work we have devised a new approach to centroid computation with multi-sets. Because the same problems will happen using the union and the intersection of multi-sets as in the case of the centroid over sets, our approach is inspired in the work of Giannotti et al.[12] but in our case over multi-sets. For each cluster $C = \{W(d_1), W(d_2)...W(d_n)\}$ the multi-set representing its centroid $centroid(C)$ is computed as in eq. 4.

$$centroid_\gamma(C) = \bigcup f(C_\gamma, h) \mid h \in \biguplus_{i=1..n} W(d_i), \frac{hf(h,C)}{n} \geq \gamma \tag{4}$$

where $hf(h, C)$ is the number of occurrences of hash h in the join of all the documents of the cluster C, and $f(C_\gamma, h)$ is a function that for a hash h returns a multi-set composed of h repeated so many times as $(\frac{hf(h,C)}{n})/\gamma$.

4 Evaluation

In order to assess the outcomes of clustering based on winnowing fingerprints, evaluation of the efficiency and effectiveness was carried out. The comparison was made between the use of the whole documents with different representations: term frequency (TF) and mutual information (MI); n-fingerprints (NFP) and winnowing fingerprints (WFP). For all the representations the same preprocessing of the input documents text was performed: lowercasing of- the texts, symbol removal, stopword removal and no stemming avoiding in this way any advantage of the fingerprint methods.

Test Data. The evaluation was done with three different collections widely used in text classification and clustering. The documents of these three datasets can be split in different classes, these classes are the ones that the effectiveness measures will take as an input to compare with the clustering output. The collections and the splits used are:

- Reuters-21578 V.12 is divided in 92 non disjointed classes. The 2742 documents that having the attribute LEWISSPLIT = TEST and the *BODY* element, were assigned at least to one topic were selected.

- WebKB contains web pages collected from the computer science departments of four different universities classified in seven different categories. For our evaluation we only used the 4,199 documents from four of the categories: "course","faculty", "project" and "student".
- 20News-18828 is composed of 18,828 documents divided on 20 different newsgroups, but some of them are close related areas and usually also it is considered as divided in 6 macro-topics according to the subject matter ("computers", "sports", "for-sale", "politics","science", "religion"), this was the selected division for our evaluation process.

Clustering Methods. Different document representations were assessed with traditional clustering algorithms: agglomerative (average-link) and partitional (batch k-means). We studied the behaviour, in terms of effectiveness and efficiency in order to assess the degradation introduced by the use of fingerprints as document representations.

Effectiveness Measures. The effectiveness of the clustering algorithms was assessed using four external criteria of cluster quality. All the metrics are based on comparing the clustering outcomes with another manually done split (answer key that defines the classes) of the collection that is used as a judgement criteria. Tree edit (TE) [13] is a measure based on an edit distance, it computes the distance from the clustering results to the manual split and how good is this distance respect to the one between having each document in a cluster, i.e. no clustering done, and the manual classification. Purity (P) [14] is a precision metric that also measures how well the clustering results match the manual split in average. F-measure (F) [14] is centred on the best match between the target class and the resulting cluster, and entropy (E) [14] is computed as an average of the entropy of each cluster, being in that way an average measure of the order/randomness.

Methodology. The evaluation was carried out as follows: Both clustering algorithms were run over the three collections with the four different representations. The computation times were tracked for every run, including I/O times of the collection to memory, representation computation and clustering. Experiments where executed in an Intel Quad-Core Q6600 2.4 GHz with 2GB of RAM.

In the case of k-means algorithm 100 runs were repeated, to deal with the randomness of the initial seeding, and the best values are reported. For each collection, in both algorithms, the k values were used corresponding with the classes in the manual splits: Reuters $k = 92$, WebKB $k = 4$ and 20News $k = 6$. In the case of n-fingerprints n was fixed to 1 because it reported the better result and for winnowing the parameters that establish the theoretical guarantees were fixed to $w = 25$ and n=5 and γ was tuned to 0.3 in the Reuters collection.

5 Results

Attending to the results (see table 1) the first fact to remark is that for every collection the results of using winnowing fingerprint are better than the results

using n-fingerprint. This was an expected point because n-fingerprints are fixed-size vectors (in our experimentation we used 1-fingerprints, which implies 26 elements).

Table 1. Results for the Reuters (1) and WebKB (2) and 20News (3) collections

(1)

Algorithm		TF	MI	WFP	NFP
K-means	TE:	0.60	0.61	0.57	0.46
	P:	0.78	0.72	0.69	0.59
	F:	0.33	0.29	0.34	0.16
	E:	0.011	0.015	0.014	0.013
	Times:	42.4s	25.6s	18.3s	10.4s
Average-link	TE:	0.53	0.58	0.56	0.38
	P:	0.77	0.80	0.69	0.57
	F:	0.73	0.69	0.63	0.40
	E:	0.010	0.012	0.013	0.018
	Times:	2441.7s	643.8s	367.1s	293.6s

(2)

Algorithm		TF	MI	WFP	NFP
K-means	TE:	0.62	0.67	0.53	0.45
	P:	0.62	0.67	0.53	0.45
	F:	0.57	0.67	0.52	0.41
	E:	0.100	0.091	0.103	0.106
	Times:	114.2s	112.6s	20.4s	8.3s
Average-link	TE:	0.39	0.39	0.38	0.39
	P:	0.39	0.39	0.39	0.39
	F:	0.43	0.43	0.43	0.43
	E:	0.110	0.011	0.110	0.110
	Times:	386.6m	142.2m	236.1m	19.52m

(3)

Algorithm		TF	MI	WFP	NFP
K-means	TE:	0.45	0.69	0.35	0.32
	P:	0.45	0.69	0.35	0.32
	F:	0.47	0.64	0.32	0.27
	E:	0.092	0.034	0.105	0.109
	Times:	18.6m	23.4m	59.8s	21.2s
Average-link	TE:	0.26	0.27	0.26	0.25
	P:	0.26	0.26	0.26	0.25
	F:	0.32	0.32	0.32	0.32
	E:	0.114	0.114	0.114	0.115
	Times:	3323.5m	5618.3m	2260.2m	532.25m

In the Reuters collection with average-link for the tree edit distance the winnowing fingerprint even outperformed the term frequency representation. In the other collections the average-link results for the given k are poor for every representation. Also with the k-means algorithm the F-measure reported better values for winnowing than for term frequency. For the other measures and collections the values of winnowing show some degradation comparing with the term-frequency representation. Comparing winnowing fingerprint with mutual information representation we obtained that MI clearly outperforms WFP and also TF. The best behaviour of mutual information in terms of effectiveness is explained by the fact that it uses more information and collection statistics for its computation, this also produces that the computation of MI vectors is slower than TF and even more than WFP. This fact can be cleared observed in the computation times of the 20News where the computation of the MI vectors for the 18,828 documents produced a drastic degradation of the computational performance of MI. Interestingly in the small collections the computation times under MI are lower than the ones with TF, this is explained by the fact that with k-means richer MI representation produces faster algorithm convergence, and with average-link the TF produces a worse balancing in the dendogram implying a lot more operations to calculate document-document similarities.

6 Conclusions and Future Work

We have implemented two traditional clustering algorithms with document representation based on winnowing fingerprints. We have adapted the similarity measures for working with multi-sets and designed a new way of centroid computation. We have compared the performance of winnowing fingerprints with

term frequency and mutual information and n-fingerprints with four different metrics and with three different collections. The achieved results show that further evaluation of the presented approach in tasks like cluster based retrieval or clustering of web results should be performed.

Acknowledgements: This work was co-funded by Ministerio de Ciencia e Innovación, FEDER and Xunta de Galicia under projects TIN2008-06566-C04-04 and 07SIN005206PR

References

1. Rijsbergen, C.V.: Information Retrieval. Butterworths, London (1979)
2. Liu, X., Croft, W.B.: Cluster-based retrieval using language models. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and Development in Information Retrieval, NY, USA, ACM Press (2004) 186–193
3. McQueen, J.: Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability **1** (1967) 281–297
4. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys **31**(3) (1999) 264–323
5. Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/gather: a cluster-based approach to browsing large document collections. In: Proceedings of the 15th annual international ACM SIGIR conference on Research and Development in Information Retrieval, NY, USA, ACM Press (1992) 318–329
6. Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G.: Syntactic clustering of the web. In: Selected papers from the sixth international conference on World Wide Web, Essex, UK, Elsevier Science Publishers Ltd. (1997) 1157–1166
7. Puppin, D., Silvestri, F.: The query-vector document model. In: Proceedings of the 15th ACM international conference on Information and Knowledge Management, NY, USA, ACM Press (2006) 880–881
8. Schleimer, S., Wilkerson, D.S., Aiken, A.: Winnowing: local algorithms for document fingerprinting. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of Data, NY, USA, ACM Press (2003) 76–85
9. Parapar, J., Álvaro Barreiro: Winnowing-based text clustering. In: Proceeding of the 17th ACM conference on Information and Knowledge Management, NY, USA, ACM (2008) 1353–1354
10. Rivest, R.L.: The MD5 message digest algorithm. RFC 1321 (Apr 1992)
11. Karp, R.M., Rabin, M.O.: Efficient randomized pattern-matching algorithms. IBM Journal of Research and Development **31**(2) (1987) 249–260
12. Giannotti, F., Gozzi, C., Manco, G.: Characterizing web user accesses: A transactional approach to web log clustering. In: Proceedings of the International Conference on Information Technology: Coding and Computing, Washington, DC, USA, IEEE Computer Society (2002) 312–317
13. Pantel, P., Lin, D.: Document clustering with committees. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and Development in Information Retrieval, NY, USA, ACM Press (2002) 199–206
14. Rosell, M., Kann, V., Litton, J.E.: Comparing comparisons: Document clustering evaluation using two manual classifications. In: Proceedings of the International Conference on Natural Language Processing. (2004)