BUAP: Performance of K-Star at the INEX'09 Clustering Task

David Pinto[1], Mireya Tovar[1], Darnes Vilariño[1], Beatriz Beltrán[1], Héctor Jiménez-Salazar[2], and Basilia Campos[1]

[1] Faculty of Computer Science
B. Autonomous University of Puebla, Mexico
[2] Department of Information Technologies
Autonomous Metropolitan University, Mexico
{dpinto,mtovar,darnes,bbeltran}@cs.buap.mx, hgimenezs@gmail.com

Abstract. The aim of this paper is to use unsupervised classification techniques in order to group the documents of a given huge collection into clusters. We approached this challenge by using a simple clustering algorithm (K-Star) in a recursive clustering process over subsets of the complete collection.

The presented approach is a scalable algorithm which may automatically discover the number of clusters. The obtained results outperformed different baselines presented in the INEX 2009 clustering task.

1 Introduction

The INEX 2009 clustering task was presented with the purpose of being an evaluation forum for providing a platform to measure the performance of clustering methods over a real-world and high-volume Wikipedia collection.

Clustering analysis refers to the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait, often proximity, according to some defined distance measure [1,2,3].

Clustering methods are usually classified with respect to their underlying algorithmic approaches. Hierarchical, iterative (or partitional) and density-based are some possible categories belonging to this taxonomy. In Figure 1 we can see the taxonomy presented in [4].

Hierarchical algorithms find successive clusters using previously established ones, whereas partitional algorithms determine all clusters at once. Hierarchical algorithms can be agglomerative ("bottom-up") or divisive ("top-down"); agglomerative algorithms begin with each element as a separate cluster and merge the obtained clusters into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. Iterative algorithms start with some initial clusters (their number either being unknown in advance or given a priori) and intend to successively improve the existing cluster set by changing their "representatives" ("centers of gravity" or "centroids"), like in K-Means [3] or by iterative node-exchanging (like in [5]).

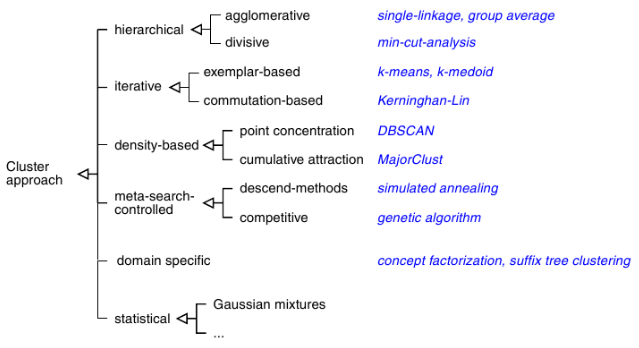

Fig. 1. A taxonomy of clustering methods as presented in [4] (Reproduced with permission of the author)

In this paper we report the obtained results when the K-Star clustering method was applied to the INEX2009_SUBSET collection. Therefore, the complete description of this clustering method is given in the following section.

2 The K-Star Clustering Method

K-Star [6] is a hierarchical agglomerative clustering method which automatically reveals the number of clusters, unknown in advance. As the most of the clustering methods, it requires a similarity matrix of the documents to be clustered (corpus). The K-Star clustering method follows as shown in Algorithm 1.

Algorithm 1. Algorithm of the K-Star clustering method

```
   Input: A $n \times n$ similarity matrix $\varphi(d_i, d_j)$
   Output: A set of clusters $\{C_1, C_2, \cdots\}$
1  Cluster=1;
2  ClusteredSet = $\emptyset$;
3  while $|ClusteredSet| < n$ do
4      $C_{Cluster} = C_{Cluster} \bigcup \arg\max_{\{d_i,d_j\}} \varphi(d_i, d_j)$;
5      ClusteredSet = ClusteredSet $\bigcup \{d_i, d_j\}$;
6      foreach $d_k \notin ClusteredSet$ do
7          if $\varphi(d_k, d_i) > \tau$ then
8              $C_{Cluster} = C_{Cluster} \bigcup \{d_k\}$;
9              ClusteredSet = ClusteredSet $\bigcup \{d_k\}$;
10         end
11     end
12     Cluster = Cluster + 1;
13 end
14 return $C_1, C_2, \cdots$
```

In this work, we have used a canonic threshold τ defined as the average of the values in the similarity matrix. The rationale of the similarity matrix construction is described in the following section.

3 Construction of the Similarity Matrix

We assume that the complete document clustering task may be carried out by executing at least the following three steps: (1) document representation; (2) calculus of a similarity matrix which represents the similarity degree among all the documents of the collection; and (3) clustering of the documents. In particular, the construction of the similarity matrix was carried out by means of the TF-IDF measure which is described into detail as follows.

The Term Frequency and Inverse Document Frequency (tf-idf) is a statistical measure of weight often used in natural language processing to determine how important a term is in a given corpus, by using a vectorial representation. The importance of each term increases proportionally to the number of times this term appears in the document (frequency), but is offset by the frequency of the term in the corpus. In this document, we will refer to the tf-idf as the complete similarity process of using the tf-idf weight and a special similarity measure proposed by Salton [7] for the Vector Space Model, which is based on the use of the cosine among vectors representing the documents.

The tf component of the formula is calculated by the normalized frequency of the term, whereas the idf is obtained by dividing the number of documents in the corpus by the number of documents which contain the term, and then taking the logarithm of that quotient. Given a corpus D and a document d_j ($d_j \in D$), the tf-idf value for a term t_i in d_j is obtained by the product between the normalized frequency of the term t_i in the document d_j (tf_{ij}) and the inverse document frequency of the term in the corpus ($idf(t_i)$) as follows:

$$tf_{ij} = \frac{tf(t_i, d_j)}{\sum_{k=1}^{|d_j|} tf(t_k, d_j)} \tag{1}$$

$$idf(t_i) = log\left(\frac{|D|}{|d : t_i \in d, d \in D|}\right) \tag{2}$$

$$tf\text{-}idf = tf_{ij} * idf(t_i) \tag{3}$$

Each document can be represented by a vector where each entry corresponds to the tf-idf value obtained by each vocabulary term of the given document. Thus, given two documents in vectorial representation, d_i and d_j, it is possible to calculate the cosine of the angle between these two vectors as follows:

$$\mathrm{Cos}_\theta(\overrightarrow{d_i}, \overrightarrow{d_j}) = \frac{\overrightarrow{d_i} \cdot \overrightarrow{d_j}}{\left\|\overrightarrow{d_i}\right\| \left\|\overrightarrow{d_j}\right\|}$$

The similarity matrix is then constructed on the basis of the above formulae, i.e., for each possible pair of documents, we need to calculate how similar they are by using the cosine measure. Once the similarity matrix is calculated, we may proceed with the clustering step. The complete description of the implemented clustering approach is given in the following section.

4 Description of the Approach

We have approached a clustering technique by partitioning the complete document collection. The divide-and-conquer approach described here was motivated by the time and space complexity needed in order to cluster huge volumenes of data. Instead of constructing one similarity matriz of high dimensionality for a document collection D (54,575 $\times$ 54,575), we constructed m similarity matrices of low dimensionality ($\frac{|D|}{m} \times \frac{|D|}{m}$). The proposed method actually allow the clustering process to be performed in a considerable lower amount of time in comparison with the traditional approach. The process followed is presented in Algorithm 2, whereas a scheme of the same process is given in Figure 2.

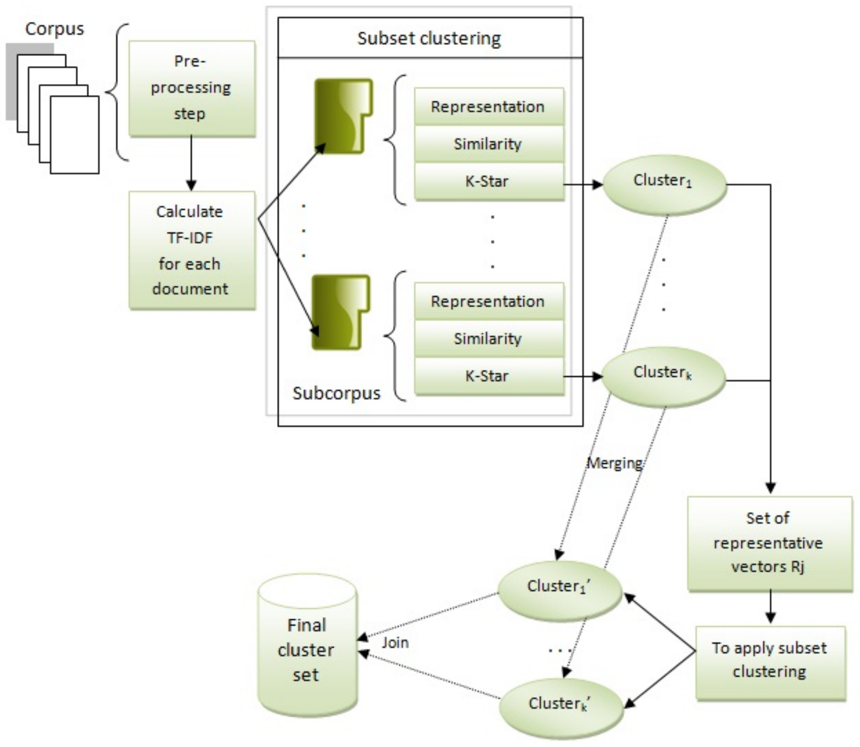

Fig. 2. Two step approach of BUAP Team at the INEX 2009 clustering task

Algorithm 2. Algorithm used for clustering the INEX2009_SUBSET with K-Star

```
   Input: A document collection $D$
   Output: A set of clusters $C_{i,j}$
1  Compute a global dictionary which contains all unique terms found in a
   collection of documents and their respective frequency of occurrence;
2  Eliminate all those terms whose frequency is lower than $\beta$;
3  Represent each document according to TF-IDF;
4  Split $D$ into $m$ subsets $D_i$ made of $\frac{|D|}{m}$ documents;
5  foreach $D_i \in D$ such as $D_i = \{d_{i,1}, d_{i,2}, d_{i,3}, \cdots d_{i,\frac{|D|}{m}}\}$ do
6      Calculate the similarity matrix $M_i$ of $D_i$ using the cosine measure;
7      Apply the K-Star clustering method to $M_i$ in order to discover $k$ clusters
       $(\{C_{i,1}, C_{i,2}, \cdots, C_{i,k}\})$;
8  end
9  $Loop = 1$;
10 while $Loop \leq MAX\_ITERATIONS$ do
11     Select a random representative document $d_{i,j}$ for each cluster $C_{i,j}$ obtained;
12     Let $D'$ be the set of documents $d_{i,j}$, i.e., only those that represent each
       cluster obtained;
13     Calculate the similarity matrix $M_i'$ of $D_i'$ using the cosine measure;
14     Apply the K-Star clustering method to $M_i'$ in order to discover $k$ clusters
       $(\{C'_{i,1}, C'_{i,2}, \cdots, C'_{i,k}\})$;
15     Let $C_{i,j} = C_{i,j} \bigcup C_{i,j'}$, where $d_{i,j} \in C'_{i,r}$ and $d_{i,j'} \in C'_{i,r}$ with $1 \leq r \leq k$ and
       $j <> j'$;
16     $Loop = Loop + 1$;
17 end
18 return (The set of clusters discovered)$C_{i,j}$
```

The obtained results are presented and discussed in the following section.

5 Experimental Results

The clustering task of INEX 2009 evaluated unsupervised machine learning solutions against the ground truth categories by using standard evaluation criteria such as Purity, Entropy and F-score.

Even if the complete description of the dataset used in the clustering task of INEX 2009 is given in the track overview paper, we may describe general features of this corpus.

The INEX XML Wikipedia collection used in the experiments is a subset of the complet corpus. This subset contains 54,575 documents pre-processed in order to provide various representations of the documents. The aim of this pre-processing was to enable the participation of different teams with minimal overheads in data-preparation the collection. It was provided, for instance, a bag-of-words representation of terms and frequent phrases in a document, frequencies of various XML structures in the form of tags, trees, links, named entities, etc.

Table 1. Purity of the INEX09_SUBSET clustering evaluation using 73,944 YAGO categories (all YAGO categories)

Run ID	*Description*	*Clusters*	*Micro Purity*	*Macro Purity*
24	random-54575	54575	1.0	1.0
11	ground truth for 73,944 YAGO categories	73944	0.9999	1.0
67	KStar method Bigrams	35569	0.6812	0.8968
65	KStar method Unigram Stems	7019	0.1894	0.6399
66	KStar method Stems Bigrams	6961	0.1883	0.6350

Table 2. Purity of the INEX09_SUBSET clustering evaluation using 12,803 YAGO categories (containing ≥ 5 documents)

Run ID	*Description*	*Clusters*	*Micro Purity*	*Macro Purity*
12	ground truth for 12,804 YAGO categories	12804	1.0	1.0
24	random-54575	54575	1.0	1.0
67	KStar method Bigrams	35569	0.6964	0.9006
65	KStar method Unigram Stems	7019	0.2076	0.6410
66	KStar method Stems Bigrams	6961	0.2074	0.6407

Table 3. Entropy of the INEX09_SUBSET clustering evaluation using 73,944 YAGO categories (all YAGO categories)

Run ID	*Description*	*Clusters*	*Micro Entropy*	*Macro Entropy*
66	KStar method Stems Bigrams	6961	0.1883	0.6350
65	KStar method Unigram Stems	7019	0.1894	0.6399
11	ground truth for 73,944 YAGO categories	73944	0.9999	1.0
67	KStar method Bigrams	35569	0.6812	0.8968
24	random-54575	54575	1.0	1.0

Table 4. Entropy of the INEX09_SUBSET clustering evaluation using 12,803 YAGO categories (containing ≥ 5 documents)

Run ID	*Description*	*Clusters*	*Micro Entropy*	*Macro Entropy*
66	KStar method Stems Bigrams	6961	0.2074	0.6407
65	KStar method Unigram Stems	7019	0.2076	0.6410
12	ground truth for 12,804 YAGO categories	12804	1.0	1.0
67	KStar method Bigrams	35569	0.6964	0.9006
24	random-54575	54575	1.0	1.0

In the experiments carried out, we approached three different representations of data: unigram stems, bigrams and bigram stems which were executed in order to observe the K-Star clustering method performance.

In Tables 1 and 2 we may see the obtained results of the three approaches with respect to two baselines, one random assignment (random-54575) and one ground truth over 73,944 and 12,803 YAGO categories, respectively. It may be observed that the bigram representation shows a high Micro and Macro Purity.

Tables 3 and 4 present the K-Star performance in terms of entropy. In general the three aproaches show a low entropy value.

In general, it can be noticed that the two-step approach presented in this paper performs well, given the above mentioned evaluation measures. However, we consider that the performance may be improved by considering a better way of selecting the cluster representative documents that are used in the second step of the algorithm, for instance, the use of cluster centroids.

6 Conclusions

A recursive method based on the K-Star clustering method has been proposed in this paper. The aim of the presented approach was to allow high scalability of the clustering algorithm. Traditional clustering of huge volumes of data requires to calculate a two dimensional similarity matrix. A process which needs quadratic time complexity with respect to the number of documents. The lower the dimensionality of the similarity matrix, the faster the clustering algorithm will be executed. The high scalability is then the contribution of the clustering method presented in this paper. There still however the fact that we are not taking into account the complete information, since we only considered subsets of the complete dataset. However, we observed that the proposed approach is easy of being implemented and obtained good Purity and Entropy results in the INEX 2009 clustering task.

Acknowledgments

This work has been partially supported by CONACYT and PROMEP projects.

References

1. MacKay, D.J.C.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
2. Mirkin, B.G.: Mathematical Classification and Clustering. Springer, Heidelberg (1996)
3. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press, Berkeley (1967)
4. Meyer zu Eissen, S.: On information need and categorizing search. PhD dissertation, University of Paderborn, Germany (2007)
5. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. Bell Systems Technical Journal 49(2), 291–308 (1970)
6. Shin, K., Han, S.Y.: Fast clustering algorithm for information organization. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, pp. 619–622. Springer, Heidelberg (2003)
7. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. Communications of the ACM 18(11), 613–620 (1975)