UNIVERSITÀ DEGLI STUDI DI PADOVA

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

REPORT DATA MINING PROJECT

TRACK: SUGGESTED PROJECT - GOAL B

Matteo Biasetton - ID: 1141416
Luca Borin - ID: 1134473
Iacopo Mandatelli - ID: 1151791
Luca Piazzon - ID: 1153130

1. Introduction

CHOSEN TRACK

The project aims at implementing and testing clustering strategies on the dataset. Our group chose to explore the GOAL B track, described as follow:

Goal B. Try to assess to what extent a clustering is consistent with the categories attached to the Wikipedia pages. To this purpose you will have to decide how to measure the consistency of a clustering with the categories, and to find out which preprocessing, document representation and clustering type, yields the best results.

DATASET DESCRIPTION

The dataset consists of a set of Wikipedia pages (in English), where each page is characterized by a unique ID, a title, a text (sequence of words separated by space), and a list of at most 279 categories out of a set of size 1102644. There are two available datasets:

- **medium-size dataset (compressed)** (100054 pages, 67 MB)
- **large-size dataset** (500137 pages, 330 MB).

2. Implemented clustering techniques

Taking for granted the definitions of the course's slides, we explored different type of clustering techniques:

- **K-Means||:** we used the standard implementation of K-Means|| by Spark 2.1.0 [1].
- **Latent Dirichlet Allocation (LDA):** it's a topic model which infers topics from a collection of text documents, intended as vectors of word counts (bag of words). Rather than estimating a clustering using a traditional distance, LDA [2] uses a function based on a statistical model of how text documents are generated.
  In order to compare LDA results with the other clustering techniques, we decided to cluster every document into the most representative topic.
  There are no warranties that every inferred topic is assigned to at least one document, so the number of clusters may be lower than the number of inferred topics.

- **K-Center (Farthest First Traversal approximation):** we implemented the algorithm defined in course's slides [3], in each iteration we keep the nearest center instead of computing the partitions at the end.
- **K-Medians:** we implemented the algorithm defined in course's slides [3], and later we implemented the proposed faster version[4].
- **Random Centers:** The algorithm first chooses K random points from the dataset, then associates each document to his cluster according to the nearest center.
- **Random Clustering:** this algorithms is the simplest one, it assigns a random cluster to each document.

3. Implemented evaluation metrics

For the evaluation of the clustering we implemented the metrics described in the course's slide [5].

- **Unsupervised**
  - Silhouette
  - Cohesion
  - Separation
  - Hopkin statistic
- **Supervised**
  - Rand coefficient
  - Jaccard coefficient
  - Cluster entropy
  - Category entropy

4. Project structure

The project has been organized in the following steps:

1. Dataset preprocessing (Dataset acquisition, Words preprocessing)
2. Creation of a proper document representation (Word2Vec, Bag of Words)
3. Execution of the cluster algorithm
4. Category mapping (Most frequent category, Nearest category)
5. Evaluation of the clustering results

5. Initial parameter assessment

Considering the available computational power, we firstly decided to use a reduced version of the medium dataset to execute the preliminary testing. In particular, we used a 10% sample to determine the best document representation (Word2Vec vs BagOfWords), the best size of the vector representation of the document and the minimum number of occurrences of each word in the entire corpus to later train the Word2Vec model. Initially, to execute the tests, we used two standard distance functions, the euclidean distance and the cosine distance, in agreement with the course's slides [3]. After some tests with the two distances we noticed that the euclidean distance produced a poor clustering, in fact with k-means and k-center all the documents were grouped in few clusters with a lot of pages. In sight of this, we decided to execute all the remaining tests with the cosine distance, but this has prevented the use of k-means which requires the usage of the euclidean distance. We also chose the Word2Vec model as document representation, thus preventing subsequent analysis of the LDA clustering algorithm. Finally, we moved to a

40% sample in order to explore the remaining parameters, fixing the previously determined ones to perform a feasible grid search.

6. Preprocessing and document representation

We applied lemmatization and stop-word removal both to the text and categories of the raw Wikipedia pages using the edu.stanford.nlp library [6]. We then trained the Word2Vec model over the lemmatized text and, as suggested, we represented each document as the average of the vectorial representation of the words contained in its text.

7. Category mapping

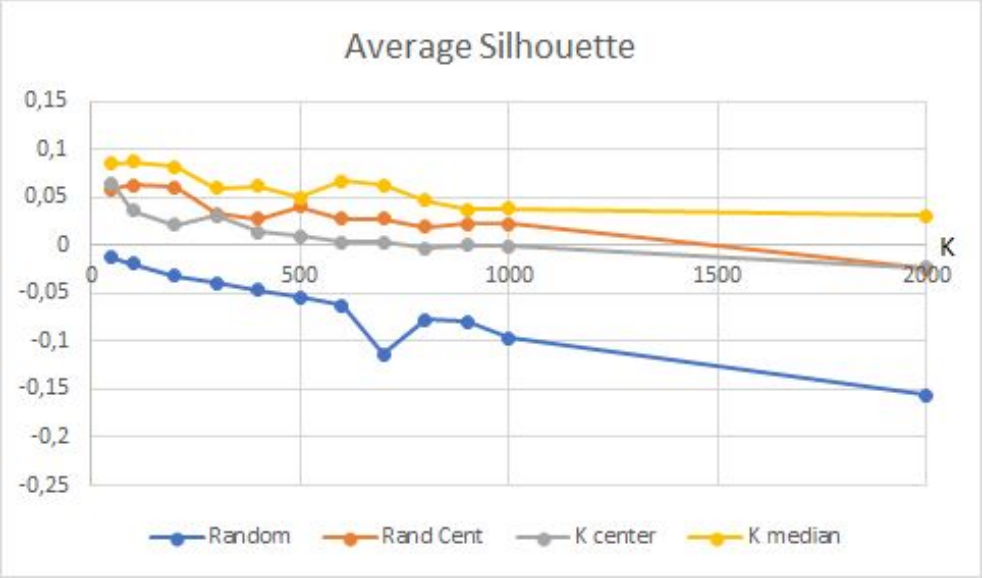
Initially we removed the insignificant categories, then, in order to evaluate the custer quality for documents with multiple categories, we decided to map each document category set into one single category using two methods:

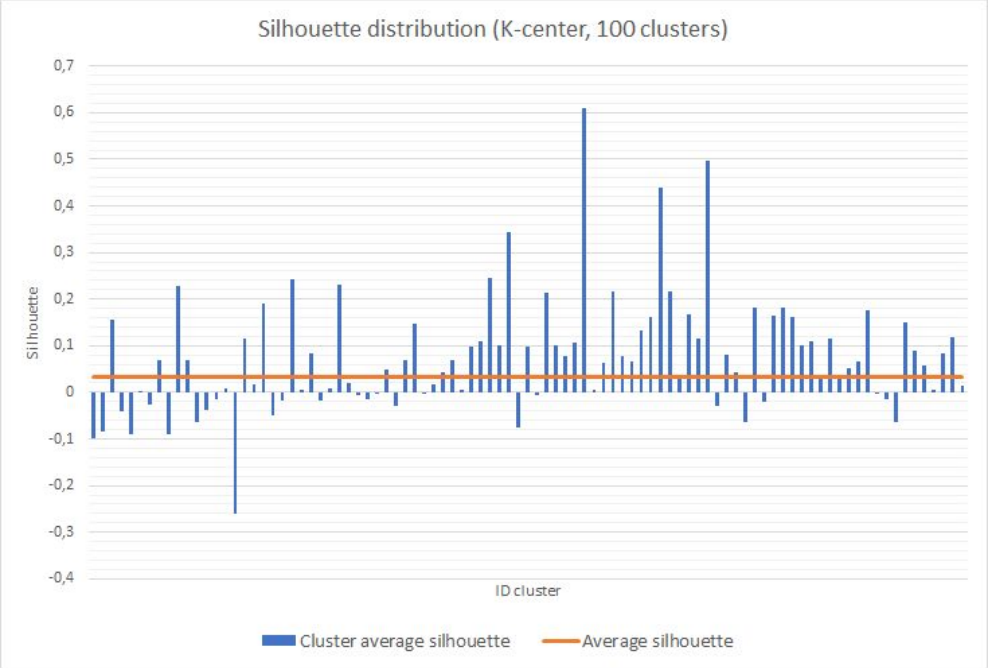
- **Most frequent category**: most frequent category of the document according to the frequency in the entire corpus.
- **Nearest category**: nearest category of the document from a vectorial distance standpoint to the document vector representation. In order to apply this idea, we used the Word2Vec model to transform each category, removing the ones that couldn't be transformed due to the absence of the word in the model.

Both approaches are meaningful, but exploring the dataset we noticed that the last idea behaves better when there are relatively rare categories strongly connected to the document that would be masked by the general frequency, however this could be also a drawback considering the desire of generalization introduced during the clustering.

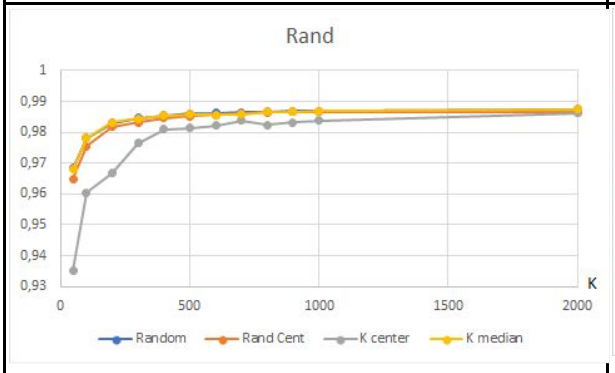
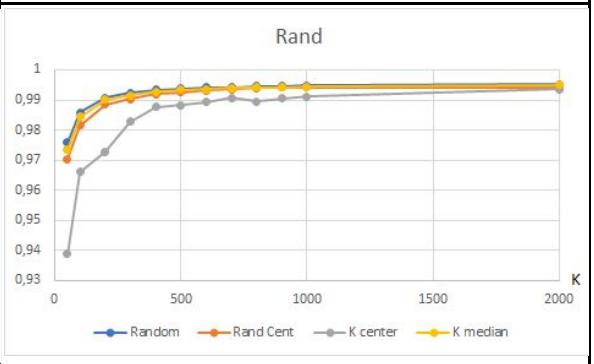
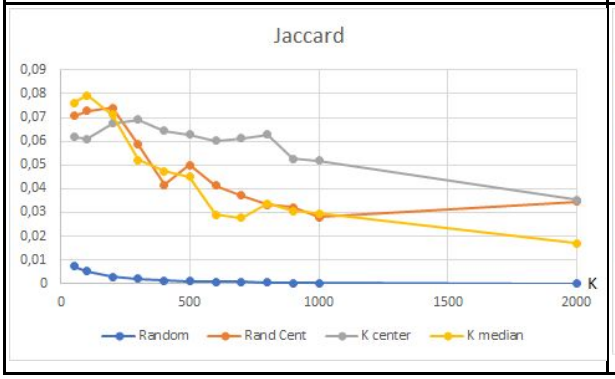
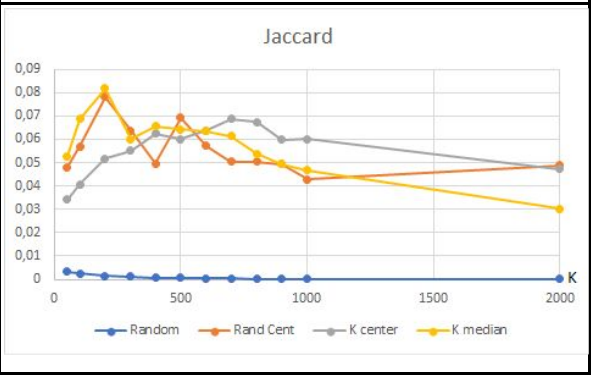
8. Evaluation of the clustering results

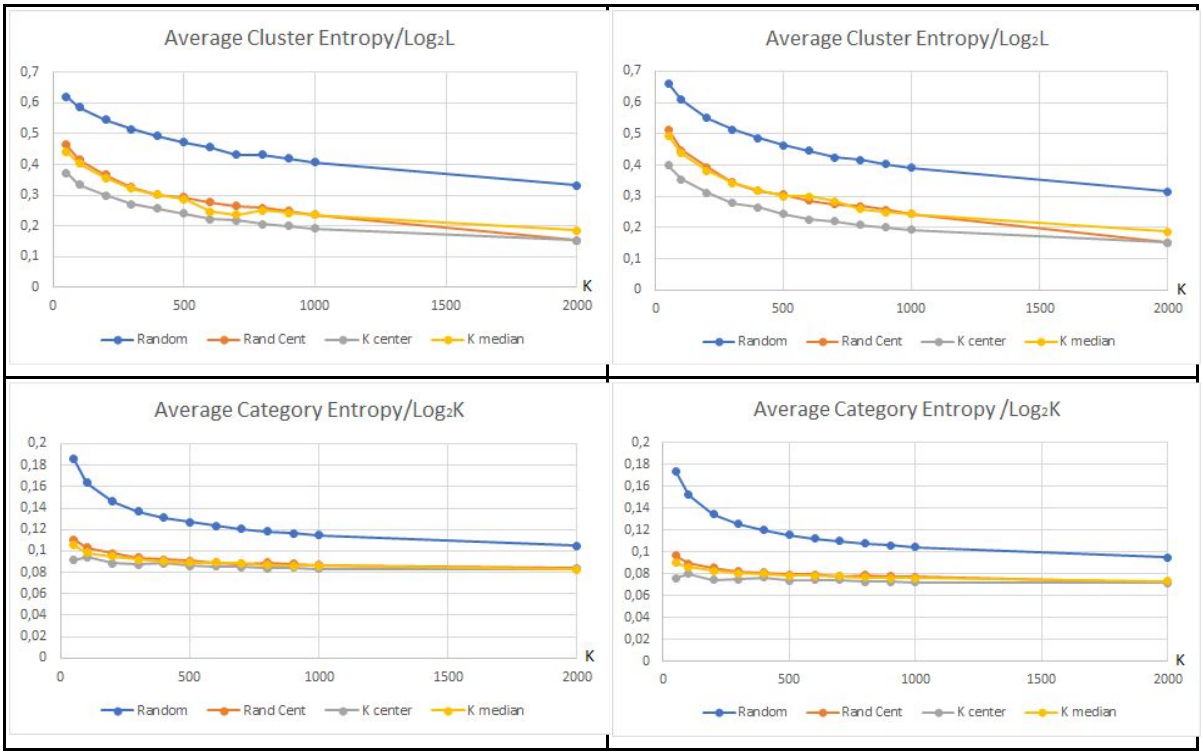
From the graphs we noticed that the evaluation metrics highlight a middling performance of the clustering algorithm. As expected, Silhouette rates K-center, K-medians and Rand-center better than the random clustering. However the average silhouette (over all clusters) doesn't well represent the average cluster silhouette, as we can observe by the following example:





Some clusters have a positive score, but there are many (small) clusters which lead to a general poor result.

Most Frequent	Nearest
	
	



in the graphs L: # categories, K: # clusters

From the Rand plots, the scenario is too optimistic and not significative: in fact this coefficient is largely determined by the component counting the couples of documents of different categories mapped in different clusters. In particular this makes random cluster results appear better than the other ones. Jaccard coefficient represents better the situation, with K-center as best clustering technique, but the coefficients are still low, coherently with Silhouette.

Entropy measures show that increasing the number of clusters, the documents with equal categories tend to be mapped in the same cluster and points in the clusters tend to have equal categories.

Cluster entropy is greater than Class entropy because there are more categories than clusters, consequently the impurity of some clusters cannot be 0 by the fact that multiple categories are mapped in the same cluster.

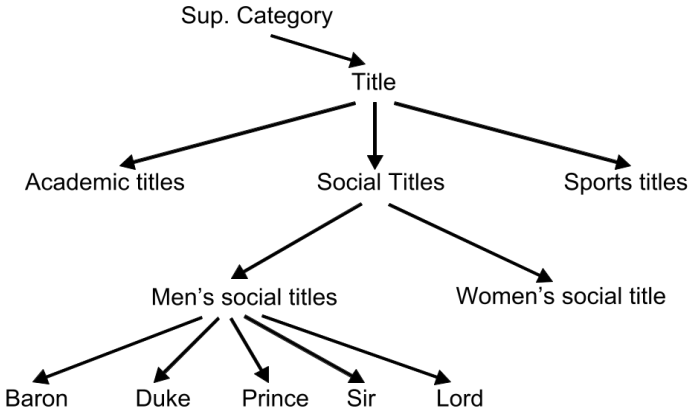
9. Problems in cluster quality evaluation

Manually inspecting the cluster composition, we observed that in general the aggregation provided were not as bad as the computed coefficients would indicate. One main problem is that there's not a clear division of the dataset directly using the original categories of the pages: two documents with similar content are usually mapped in a single cluster, but the categories might not be exactly equals, leading to a poor evaluation score of the clustering, although the grouping is not wrong from a logical point of view. We tried to face this problem using the lemmatization preprocessing of the categories and through the category mapping, which led to a more general division of the corpus. The clustering algorithms however didn't provide a sufficiently fine division of the document space, leading to poor scores. This could indicate that the clustering algorithms that we analyzed were not particularly adapt to the goal, also increasing the number of requested clusters, or that the categories need a more aggressive preprocessing in order to generalize better the document content and be comparable with the obtained clustering.

10. Conclusions and future works

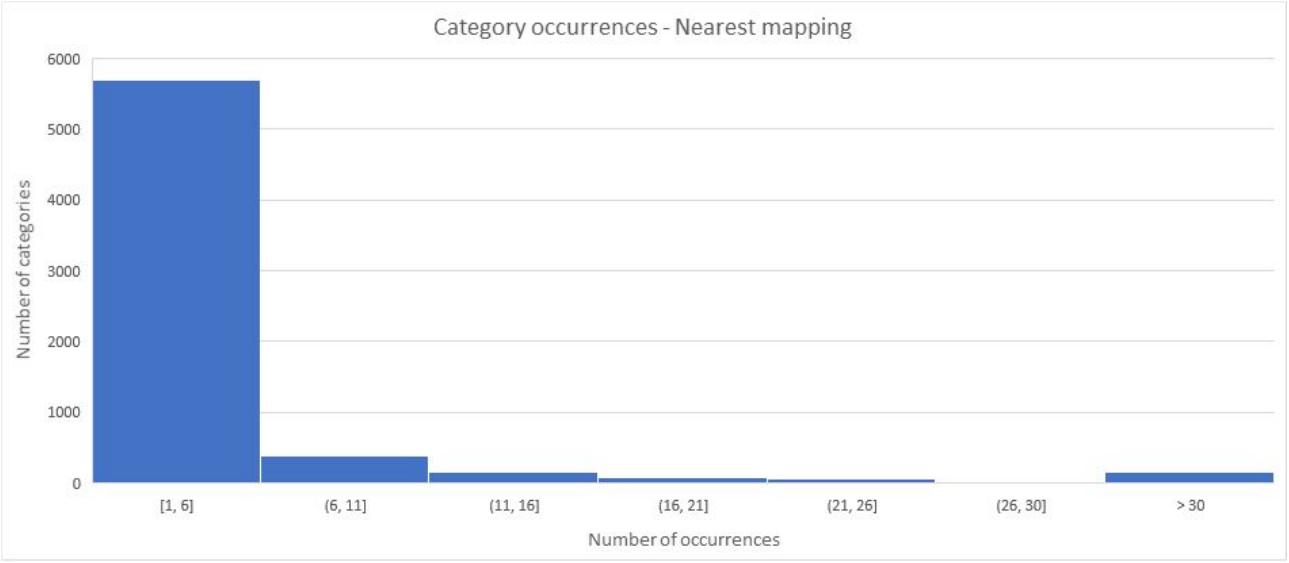
Both nearest and most frequent mapping type show that k-center is slightly better than the other clustering types. Considering the small difference observed among them, we decided to manually inspect the categories of the wikipages and found out that:

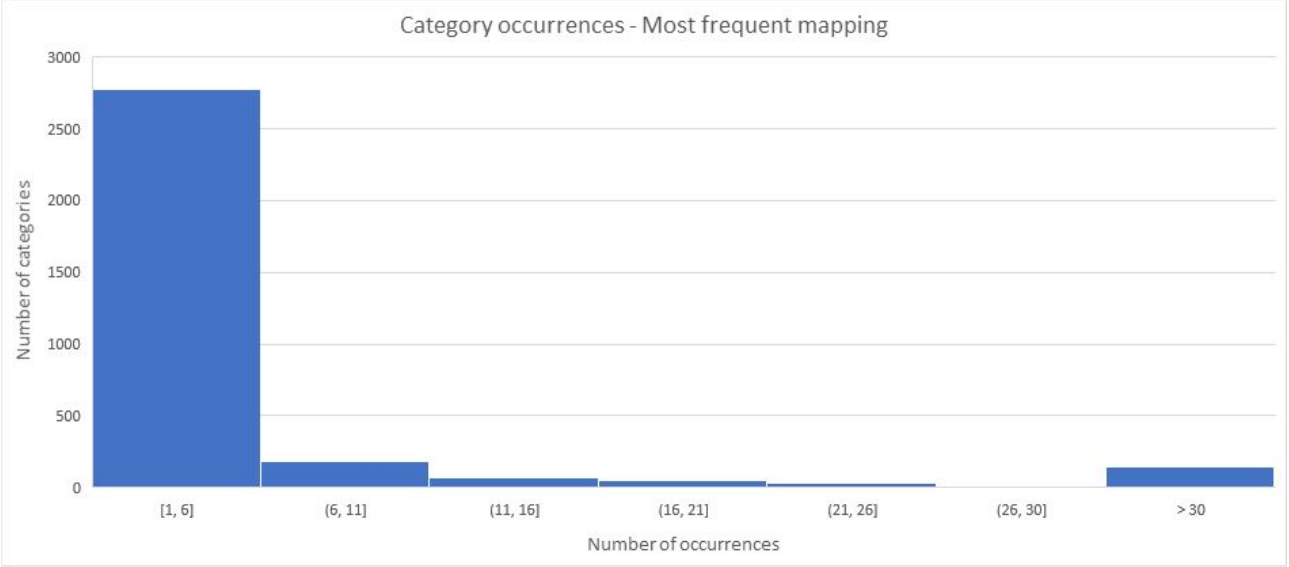
- Every editable wikipedia page is a wikipage (including disambiguation pages and pages about the categories themselves) and could have several associated categories.
- A category could be divided into subcategories, hence forming a complex category tree. This produce a huge amount of very specialized categories and, as a result, there are many rare categories and a small number with an high frequency.
- Sometimes a wikipage has not only a determined category, but also the category "predecessor" in the relative tree.



Consequently, the categories can't be considered equally and independently, while in our work we chose such simple treatment.

In order to sustain our hypothesis, we analyzed the frequency of each category in the dataset:





We can observe from these graphs that with both mapping methods, the majority of the documents are grouped in very small groups, which would require the creation of very little clusters. The creation of defined small clusters is not only really hard for a general cluster algorithm, but it is also not particularly meaningful, considering that logically the goal is a more broad grouping. Notice that with the mapping we already provided a reduction of the category space, which is also more complex in the original dataset dump.

A future work should require a deeper analysis of the wikipage categories and their structure with an adequate preprocessing, that probably would lead to better metrics also for the clustering evaluation.

One idea is to reconstruct the category tree, but this is a very complicated task because would require to automatically follow the live weblinks. Considering this tree as given, a certain reference level could be chosen and all the children could be mapped in the proper ancestor, leading to an adequate reduction of the target document distribution with respect to their categories.

An easier alternative would be to reduce the categories directly using the Word2Vec model, mapping the categories which are synonyms into a single one. This approach would require as well a deeper analysis, but may not be strong enough to create general document groups aligned with the clustering goals.

11. References

[1] “K-Means”, https://spark.apache.org/docs/2.1.0/mllib-clustering.html#k-means

[2] “LDA”, https://spark.apache.org/docs/2.1.0/mllib-clustering.html#latent-dirichlet-allocation-lda

[3] “Slide clustering part 1”, http://www.dei.unipd.it/~capri/DM/MATERIALE/Clustering1617.pdf

[4] “K-Median”, H.S. Park, C.H. Jun. A simple and fast algorithm for K-medoids clustering. Expert Syst. Appl. 36(2):3336-3341, 200

[5] “Slide clustering part 2”, http://www.dei.unipd.it/~capri/DM/MATERIALE/Clustering-2-1617.pdf

[6] “Stanford library”, https://github.com/stanfordnlp/CoreNLP