IPR project II

Group 24, Diogo Moura 86976, Pedro Pereira 90766, Vasco Morganho 81920

1 CLUSTERING

To test clustering we picked 5 topics (R101, R121, R150, R170, R180), and got all documents with judgements for those topics, using qrels.test, to use as document collection. The size of this collection is 2478. Using the entire Dtrain collection would require too much memory. We used the TfidfVectorizer from sklearn, with stop word removal and removal of terms with document frequency below 2 and terms that appear on more than 90% of documents. We used KMeans as the clustering method.

a) What is the (hypothesized) number of topic clusters? And document clusters in the *D*train collection?

.

The hypothesized number of topic clusters would be around 100 because topics are supposed to be different from one another thus each one should belong to a different cluster. The hypothesized number of document clusters would also be around 100, since the topics and the document collection are related.

b) Are the clusters from previous solutions cohesive? And well separated?

For the topics we obtained a value of 0.032 for the average cohesion of all topics and a value of 11.356 for the average separation. The clusters for the topic are very cohesive and well separated, apart for some outliers. This can be explained by the fact that some clusters are only composed by one topic.

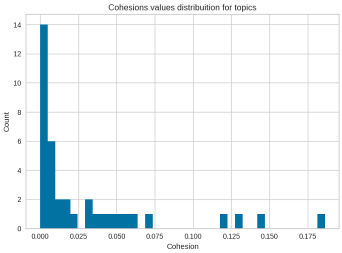

Figure 1: Cohesion Value Distribution For Topics

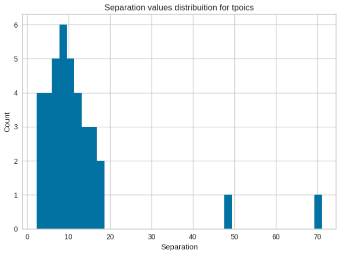

Figure 2: Separation Value Distribution For Topics

For the document collection we obtained a value of 167.041 for the average cohesion, and a value of 20923.967 for the average separation, meaning that the clusters for the document set present high cohesion and are well separated, apart for some out liars.

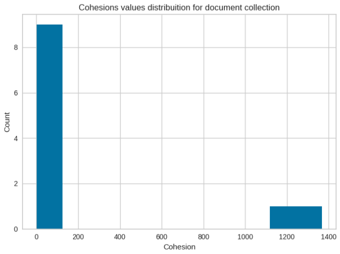

Figure 3: Cohesion Value Distribution For Document Set

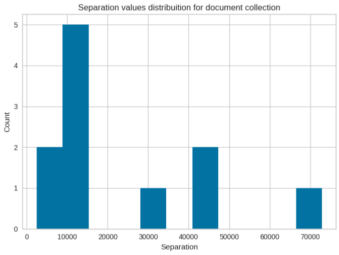

Figure 4: Separation Value Distribution For Document Set

c) What the clustering of topic documents, Q, reveals regarding their conceptual organization and independence? Are there highly similar/overlapping topics?

According to the silhouette method we concluded that the topic documents can be organized into 38 different clusters, meaning that there are some topic documents that have some degree of similarity

2nd delivery of the Information Processing and Retrieval Class (IST 2020), D. Moura, P. Pereira, V. Morganho, November 2020, Lisbon, Portugal

https://https://tecnico.ulisboa.pt

between each other. If all the topics were highly different we would have gotten values for the optimal k close to 100.

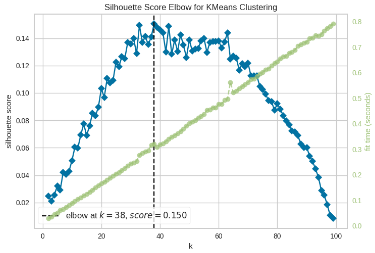

Figure 5: Silhouette Score for the topics at different values of k

d) Given a specific cluster of topics, check whether the medoid (a sort of prototype topic for the cluster) adequately represents the remaining topics in the given cluster.

Picking a cluster whose medoid is topic R112, which contains topics R120, R132 and R137. Topic R112 is about school bus accidents which resulted in death, topics R120 is about deaths in mining accidents, topics R132 is about friendly fire deaths and R137 is about sea turtle deaths. As we can see the common theme in this topic cluster is death, with R120 being more specific, by referring to death in accidents just like the medoid, R112.

Note: in some occasions topic R112 may not be the medoid of a cluster, because sometimes the clustering operation returns different clusters.

e) How are the documents in the target collection organized? Briefly discuss the importance of this information to understand the behavior of the target IR system.

The documents in the target collection are organized in 11 clusters. We came up with this value by by plotting the silhouette score for several k (number of clusters) as shown in graph 6. This means that when we perform information retrieval on this document collection, it is very likely that all the retrieved documents belong to the same cluster, at the least for small values of p(number of documents to be retrieved).

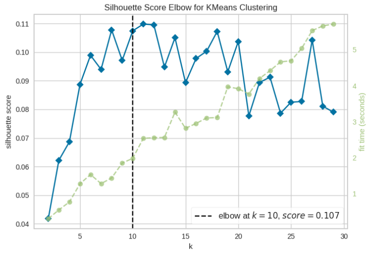

Figure 6: Silhouette Score for the document collection at different values of k

2 CLASSIFICATION

Strategies for ranking purposes

There are two strategies:

(1) Rank the documents according to their probability of being relevant.
 (a) Pros- If the classifier presents good performance (i.e. precision >0.5) this strategy will more likely aid more the IR system, since it can use the classifier to separate the relevant/non-relevant documents.
 (b) Cons- If the training dataset has very few samples, all the documents may be assigned very similar/equal probabilities and the documents won't actually be ranked (for example for KNN).

(2) Rank only the relevant documents according to the RRF
 (a) Pros- does not decrease as much the performance of the IR system with very few samples and overfitting.
 (b) Cons- it doesn't differ much from not using the classifier and if the classifier has a good performance may have worse performance than the previous strategy, since the classifier discrimination capacity over the features is lost.

Overall the results are very similar for each strategy in this collection and the chosen strategy doesn't have much impact in the performance of the aided IR System.

(a) Does the incorporation of relevance feedback significantly impact the performance of the IR system?

No, it does not. The precision of the classifiers are not very good.

Particularly, the logistic classifier could improve the performance of the IR model. On the other hand, the XGBoost classifier could not do so.

This is highly related to the types of features we are using. In fact, the classifiers can not get high precisions on the classification task if relying exclusively on the ranking scores (in our case BM25, Cosine similarity, and Term-Frequency).

In fact, if the features also included the vector representation of the document (vector space model), the classifiers could get much higher MAP score, more or less 0.7, but with the type of features used there aren't enough patterns for the classifier to pick on the data and the MAP obtained is more or less 0.4, much lower.

Despite the fact that the overall performance performance isn't greatly improved, there are some topics for which the classification greatly improves the IR system performance, for example for topic R101. This fact could be used to aid the IR system only for these topics.

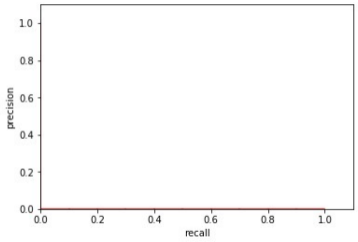

Figure 7: Precision@Recall Curve of the non-Aided IR system with $k = 10$ for topic R101

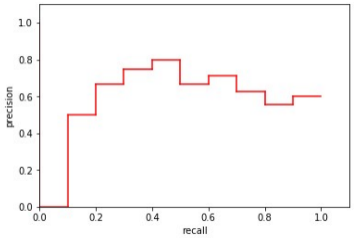

Figure 8: Precision@Recall Curve of the Aided IR system with $k = 10$ for topic R101 with document ranking according to the probability of the document being relevant

(b) Are performance improvements approximately uniform across topics? Are there topics substantially harder to classify? Which ones?

There are topics which are much more difficult to classify. This can be related to the training collection having different sizes for each topic, which can make the classification task more prone to overfitting on certain topics.

The topic R117 presents low number of judgements (12) and present very bad classification performance (MAP of the classifier of 0.44), which indicates the classifier is probably overfitting.

There are also topics for which the classification task behaves better than others, depending on how easy is to decide if the document is relevant or not relevant only, based on the the ranking scores.

The topic R118 classification with Logistic Regression classifier has a MAP of 0.98 with 31 documents in the training collection, whereas the topic R120 with more documents in the training collection (54) presents a smaller precision of 0.8. This is related to the fact that are topics for which it is easier to linearly separate the irrelevant and relevant documents.

The fact that the classifier presents good precisions only for some topics could be used to only use the aided IR system for those topics for which the classifier performance is known to be good.

(c) From the tested classification variants (algorithms and parameterizations), which one yields better performance for simple retrieval? Hypothesize why is that so.

The logistic regression classifier yields the best performance for simple retrieval.

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XGBoost supports non linearity, whereas Logistic Regression Classification supports only linear solutions.

The performance of the classifier depends on the number of samples / features present in the dataset. Since many topics have few samples and there are only three features dependent between each, the XGBoost classifier will overfit on the data, since it is a more complex classifier.

In the context of low-dimensional data (i.e. when the number of covariates is small compared to the sample size), logistic regression is considered a standard approach for binary classification.

We have a very small number of features, three, which are also dependent between each other, since much of the information contained in Cosine similarity is already contained in BM25.

Therefore, the Logistic classifier outperformed the XGBoost classifier, since it is more simple, which is better.

We performed a random Grid Search for each topic for the logistic classifier parameters. We tried different solvers, penalties and regularization terms for the Logistic Regression. The hyper-parameters take a lot of time to search and there is little to no improvement against the standard Logistic Classifier with the 'lbfgs'. This is, introducing 'L2' regularization and penalties 'C' does not improve the performance of the classifier.

(d) Does the extension of the classification setting towards ranking yield significant performance improvements? Hypothesize why is that so considering results from rank-aware measures.

The extension of the classification towards ranking does not yield significant performance improvements.

The binary classification already incorporates the information present in the classifier and the probabilities of the classifier do not outperform RRF in ranking documents.

Overall the performance

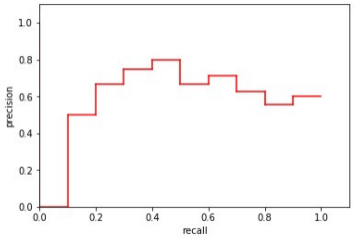

Figure 9: Precision@Recall Curve of the Aided IR system with $k = 10$ for topic R101 with document ranking according to the probability of the document being relevant

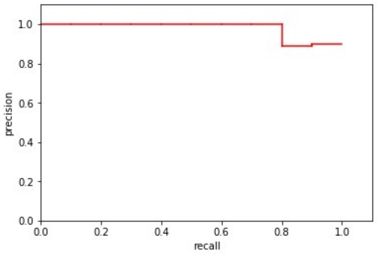

Figure 10: Precision@Recall Curve of the Aided IR system with $k = 10$ for topic R101 without ranking

(e) Does the incorporation of additional features aid the behavior of the target IR system?

The incorporation of additional features which are scores of ranking functions does not aid the behavior of the target IR system that much. Most of the information contained in Cosine Similarity and Term frequency is more or less contained in BM25, which already describes the term distribution.

Using just one feature (BM25) yielded just slightly worse results than using many features (BM25,Cosine-similarity,Term-Frequency) with respective value of MAP of the Logistic Regression classifiers being 0.395 and 0.399.

Other types of features (vector representation of the documents) would drastically improve the MAP of the classifier to approximately 0.7, but would require more memory and computational effort.

3 GRAPH RANKING

Graph Ranking was ranking was performed over the documents in qrels.train for reasons related with time and space.

(a) Does the graph ranking method based on document similarity aid IR?

Page ranking does not aid IR. Looking at table 2 we can see significant downgrades from the normal IR system to the page rank IR system. This is due to the fact that page ranking, in its pure form, it's query independent, meaning that it will always score documents with same value independently of the provided query.

Page Ranking parameters:
Iterations: 20
k (limit for retrieved docs):10
θ: 0.6
Damping Factor: 0.15

Table 1: Average Measure Values for normal retrieval and graph based retrieval

Measure	Normal	Graph Based
Precision	0.147	0.035
Recall	0.034	0.002
Fscore	0.047	0.004
MAP	0.039	0.005
BPREF	0.502	0.931

Note: BPREF is so high because it might happen that no documents classified as non-relevant are being retrieved, which in that case we return 1 to avoid a division by 0.

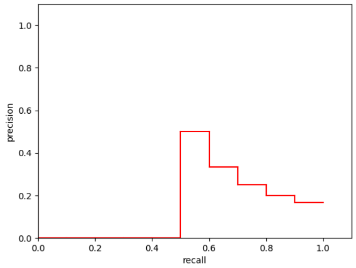

Figure 11: Precision Recall Curve For Normal Retrieval for Topic R124

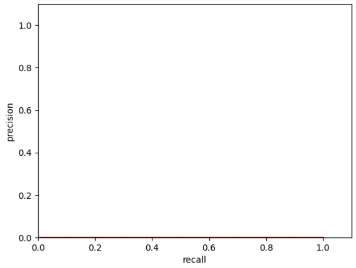

Figure 12: Precision Recall Curve For Graph Based Retrieval for Topic R124

(b) How does ranking performance vary with θ?

There is not much variation in terms of performance for different values of θ

Page Ranking parameters:
Iterations: 20
k (limit for retrieved docs):10
Damping Factor: 0.15

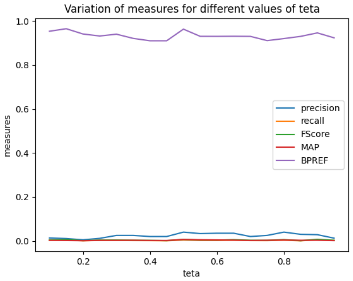

Figure 13: Variation of measures for different values of θ

Note: BPREF is so high because it might happen that no documents classified as non-relevant are being retrieved, which in that case we return 1 to avoid a division by 0.

(c) Upon the analysis of the graph, are there documents with higher graph centrality score? Which ones?

Yes, there are. Documents 20394newsML.xml and 78520newsML.xml are the ones with most graph centrality score.
Page Ranking parameters:
Iterations: 10
k (limit for retrieved docs):10
θ: 0.6
Damping Factor: 0.15

Table 2: Top 10 Page Ranked Documents

Document	Page Rank Score
20394newsML.xml	0.00186
78520newsML.xml	0.00171
20766newsML.xml	0.00168
59321newsML.xml	0.01639
49980newsML.xml	0.00140
60156newsML.xml	0.00139
36154newsML.xml	0.00139
59340newsML.xml	0.00134
48342newsML.xml	0.00132
25699newsML.xml	0.00128

(d) Does the inclusion of non-uniform prior probabilities yield performance improvements? Hypothesize why is that so.

Yes. To achieve this we set the damping factor to 0.6, which according to the page rank formula, would give more weight to the prior probabilities than the edge weights of the graph. We did this because page rank, in its normal form, it's query independent which led to bad scores.
Page Ranking parameters:
Iterations: 20
k (limit for retrieved docs):10
θ: 0.2
Damping Factor: 0.6

Table 3: Average Measure Values for normal retrieval and graph based retrieval with priors

Measure	Normal	Graph Based With Priors
Precision	0.147	0.141
Recall	0.034	0.039
Fscore	0.047	0.051
MAP	0.039	0.043
BPREF	0.501	0.573

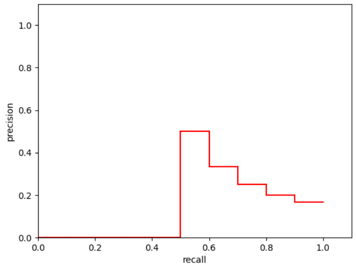

Figure 14: Precision Recall Curve For Normal Retrieval for Topic R113

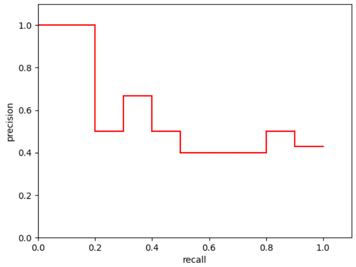

Figure 15: Precision Recall Curve For Graph Based Retrieval for Topic R113