Project II Report - Group 1

Francisco Rola 84717 | Henrique Almeida 84725 | Tomás Oliveira 84773

1 An approach based on graph-ranking

Following the project statement, we implemented an automatic keyphrase extraction method based on graph ranking. In order to do this we first selected a document from which the candidate keyphrases were to be extracted, in this case, a BBC news text. Then, after extracting the sentences from the aforementioned document we, for each of these sentences, extracted its words (filtering stop words) and generated, from these tokens, additional candidates, namely bi-grams and tri-grams. The graph itself was designed to depict associations between sentence co-occurring candidates and, consequently, the nodes of the implemented graph represent the keyphrase candidates, while the edges (undirected) between nodes represent two nodes co-occurring in the same sentence. To be able to rank these candidates (i.e., the graph's nodes) the PageRank algorithm (undirected graph version) was employed with a damping factor of 0.15 and a maximum of 50 iterations. After computing the PageRank algorithm over the graph, its results (i.e., a map between each node and its rank score) were sorted, and the top-5 best-performing (higher rank score) nodes returned as the document's keyphrases.

2 Improving the graph-ranking method

As to extend the graph-ranking method, we implemented different variations of this approach and tested their performance on the 500N-KPCrowd dataset. Similarly to the previous exercise, we first created a graph from the dataset's candidates (word tokens with stop words filtered, bi-grams, and tri-grams) for each of the dataset's documents. Then, for the baseline approach (i.e., PageRank for undirected graphs with uniform priors), the PageRank for each document's graph was computed, and the 10 most relevant keyphrases were returned. Another 4 variations of this approach were then implemented and tested, as well as their combinations.

Firstly, we considered a variation of PageRank with non-uniform prior weights for each node (i.e., candidate). Leveraging the insight that longer keyphrases appearing in the first sentences of a document are more descriptive, the prior weights for each candidate of each document were computed using Equation 1.

$$candidate\ length \times 0.1 + \frac{total\ sentences}{candidate\ position} \quad (1)$$

Secondly, we employed two additional approaches for non-uniform prior weights using TF-IDF and BM25 scores. Both scores were computed over the entire collection of used documents, with each candidate of each document being mapped to a TF-IDF and BM25 score.

Thirdly, the graph's edges, which up until this point were unweighted, were assigned a weight. In order to this, the number of co-occurrences in a sentence per candidate per document was counted, normalized, and added as the weight of the edge connecting the co-occurring nodes.

Lastly, edge weights were assigned using the similarity between candidates' pre-trained word embeddings. Using the *wiki-news-300d-1M* word vectors from fastText, we, for each candidate, obtained the corresponding vector. Due to limited computing power, only the first 200000 words were used and, consequently, some candidates were not present in this set. In order to combat this issue, we, for these candidates, calculated the subword n-grams (4-grams for long words, 3-grams for medium sized words, and 2-grams for short words) of the missing candidate and then summed its subword vectors. For bi-grams and tri-grams, since *wiki-news-300d-1M* only contains embeddings for uni-gram candidates, for each word of the candidate, the corresponding word vector was retrieved, and then the average of these vectors was computed. After every candidate was assigned a representative word vector, in every document for every edge, the similarity (in this case the cosine similarity) between the edge's nodes was computed and assigned as its weight.

To compare the graph-ranking approach variations we used the Mean Average Precision (MAP) metric, having achieved a result of 6.83% for the baseline approach. Of the tested improvements, the approach which used as the edge's weights the number of co-occurrences and non-uniform prior weights based on Equation 1 achieved the best results with a MAP of 7.26%. The results for all tested approaches are present in Table 1.

3 An unsupervised rank aggregation approach

In this exercise we followed the guidelines provided in the project's statement. We started by developing a base solution and then attempted to improve on it by exploring new features and aggregation techniques.

We started by simply aggregating TF-IDF and Page Rank by using a Reciprocal Rank Fusion (RRF). This simple approach requires us to calculate the TF-IDF score for each term in each document and computing the page rank per document based on a graph structure. After having both this results we simply order the vectors in a descending faction and thus obtain an ordered ranking for each term in a document according to each of those features. After having this data, applying RRF is trivial, by following the formula in the project's statement we obtain the top ten candidates for each document for this particular aggregation technique. We obtained a mean average precision of 7.7% by following this approach.

Given our previous experience in the first stage of the project we decided to explore additional features to improve on this result. As a result we added BM25 to the aggregation process. Our results did not improve with this addition and, on the other hand, our mean average precision dropped to 6.4%. This can be explained by various factors such as a dataset characteristics etc.

Since adding a new feature did not improve our results we decided to explore another aggregation model, COMBMNZ. This model is quite simple, it multiplies the number of different rankings associated to a term by the sum of scores for that same term (i.e. if two different features rank a given term as the best key phrase then the sum of the scores is multiplied by one, if the features associate a different rank to the same term then the scalar would be two and so on). This technique combined with the BM25 addition led us to a mean average precision of 5.8%. If we exclude BM25 from this aggregation technique then we obtain 7.8% of mean average precision which was our best result in this exercise.

Lastly, we compared the results obtained in the first stage of the project through Perceptron to the ones obtained using this ranking aggregation technique (RRF) with the same feature set. We obtained a mean average precision of 0.5% with RRF and 5% with Perceptron. Note that Perceptron used an extra feature that considered the document source which cannot be applied in this context due to this nature (i.e. it cannot be placed in a vector and ordered).

We also experimented combinations of the features used in the last stage in this model, for instance running RRF with TFIDF, BM25 and TF as features gave us a mean average precision of 7.5%. Document frequency, term position and length are the features that negatively impact the aggregation model and favour Perceptron results, hence the difference between the two approaches.

Our experiments lead us to conclude that the page rank feature is the one that provides us with the best results, this feature can then be combined with other features in order to improve our model but some features affect the aggregation negatively reducing our mean average precision (e.g. BM25). We can also say that different aggregation models provide better results depending on the data set as COMBMNZ without the BM25 feature outperformed RRF.

4 A practical application

After extracting keyphrases from documents, using the previously mentioned methods, the next step consisted in applying one of this approaches to a real world example, and provide a graphical representation of the extracted key phrases. In this exercise, we opted to create a Wordcloud which was formed by the keyphrases extracted from the World RSS (World.xml) feed from the New York Times, as suggested in the course project statement.

We followed the approach which presented in the highest Mean Average Precision in exercise 2 (i.e. Non-uniform prior weights (Equation 1), edge weights (co-occurrences)), and extracted the top 50 keyphrases from the feed. The results were presented in a Wordcloud visualization in which the the size of each keyphrase is proportional to its page rank score, as seen in Figure 1.

To provider further information about the obtained results, the visualization is accompanied by a table that can be made visible by clicking in each of the keyphrases. This table provides information about the news in which the keyphrase is present, such as the title of the news. Figure 2 shows the support table for the keyphrase "president". The table shows the title, link, and categories for each of the news mentioning the selected keyphrase.

5 Appendix

Table 1: Mean Average Precision (MAP) of the approaches tested in Exercise 2.

Approach	MAP (%)
Baseline	6.83
Non-uniform prior weights (Equation 1)	6.95
Non-uniform prior weights (TF-IDF)	6.83
Non-uniform prior weights (BM25)	6.98
Edge weights (co-occurrences)	7.02
Edge weights (word vectors)	6.45
Non-uniform prior weights (Equation 1), edge weights (co-occurrences)	**7.26**
Non-uniform prior weights (TF-IDF), edge weights (co-occurrences)	7.08
Non-uniform prior weights (BM25), edge weights (co-occurrences)	7.13
Non-uniform prior weights (Equation 1), edge weights (word vectors)	6.44
Non-uniform prior weights (TF-IDF), edge weights (word vectors)	6.44
Non-uniform prior weights (BM25), edge weights (word vectors)	6.43

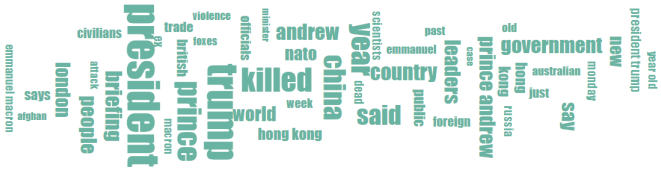
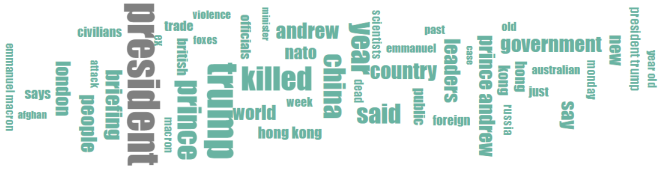


Figure 1: Top50 wordcloud for the World RSS feed.



News	HyperLink	Categories
Live Updates: NATO Chief Vows Unity at Celebration, Despite Tensions	link	United States International Relations,North Atlantic Treaty Organization,Islamic State in Iraq and Syria (ISIS),Macron, Emmanuel (1977-),Pompeo, Mike,London (England)
For Trump and Europe, a Surprising Role Reversal	link	United States International Relations,North Atlantic Treaty Organization,Johnson, Boris,Macron, Emmanuel (1977-),Merkel, Angela,Trump, Donald J,London (England),Terrorism
'I Don't Know Prince Andrew,' Trump Says. Photos Say Otherwise.	link	Politics and Government,Labour Party (Great Britain),Andrew, Duke of York,Epstein, Jeffrey E (1953-),Trump, Donald J,Giuffre, Virginia Roberts,Great Britain
Trump Warns Trade Talks With China May Last Past 2020 Election	link	International Trade and World Market,United States International Relations,Prices (Fares, Fees and Rates),Presidential Election of 2020,Trump, Donald J,China,Customs (Tariff)
In Russia, an Updated Law With New Restrictions on Freedom of Speech	link	Freedom of Speech and Expression,Putin, Vladimir V,Russia
For Boris Johnson, a Perilous Week Negotiating Terrorism and Trump	link	Sentences (Criminal),Terrorism,Conservative Party (Great Britain),Labour Party (Great Britain),Johnson, Boris,London (England),Corbyn, Jeremy (1949-),Khan, Usman (d 2019)
As Troubles Grow, Mexicans Keep the Faith With Their President	link	Lopez Obrador, Andres Manuel,Politics and Government,Economic Conditions and Trends,Labor and Jobs,Unemployment,Pemex,Mexico,Tabasco (Mexico)

Figure 2: Table with information about the "president" keyphrase.