TEAM "*CARMEN MOLA*"

CHALLENGE 1

Introduction

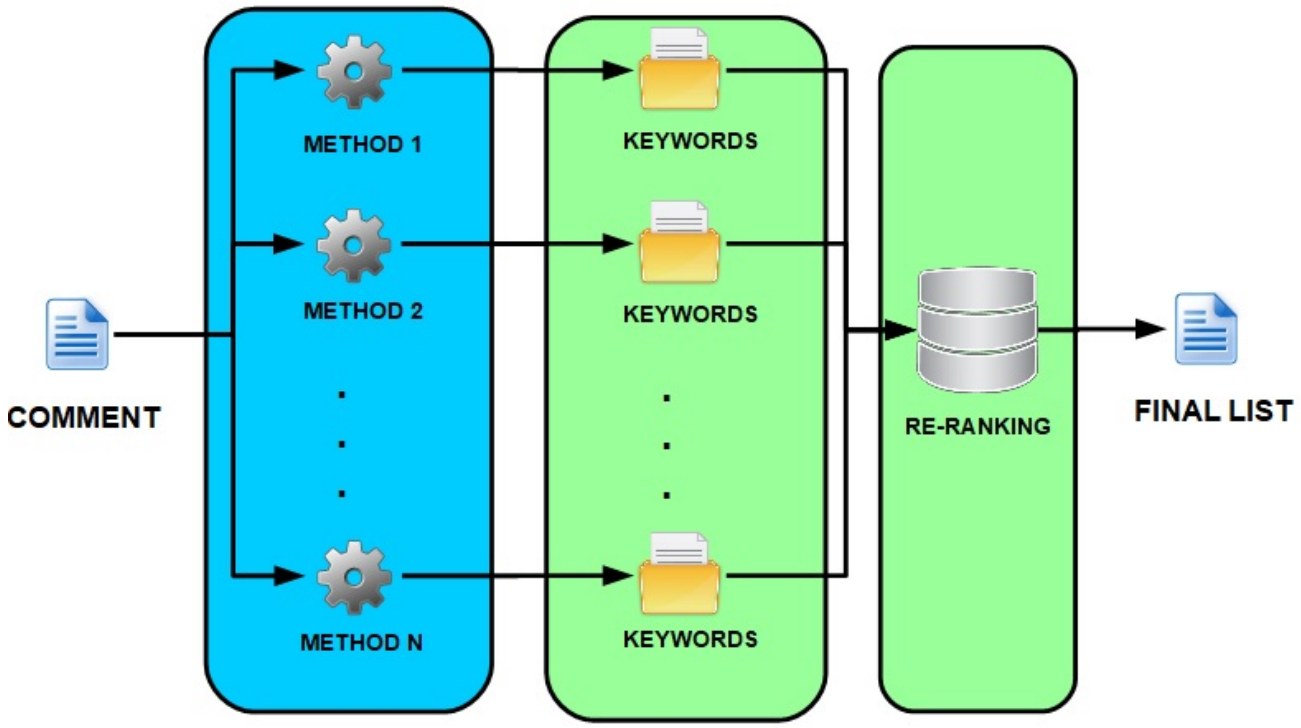
It is important to note that the number of text documents, comments, reviews and opinions available nowadays on the Internet is huge and out of the scale of human supervising. In this context automatic keyword extraction as the process of selection those words that best represent the text content, has raised as a key tool for public and private institutions or businesses.

Several approaches and models have been proposed in these last years from the simplest ones that compute statistics from the text like term frequency (TF) or word co-occurrences. Other approaches employ machine learning that trains a classifier to find keywords. More sophisticated methods are the linguistic approaches. The last methods to be incorporated in this universe are the graph-based.

Our approach is to combine some of the previous methods to create a more robust corpus of relevant keywords than applying only one of them. A similar idea is used in the paper provided by the Datathon organizers [6] which we did not about before. Our inspiration came from the paper [8], where the authors use a combination of graph centrality measures to outperform the individual ones. In order to process the large amount of data and the need of scalability we consider the best to develop a Python script to achieve the goals of the challenge.

Method

Our methodology consists on the combination of already existing methods (both statistical and graph-based) to extract a series of keywords and then choose the (up to) five more relevant for each comment. A final human check was done in order to get rid of the possible gross mistakes (this final step is to be removed in future iterations of the method). An overview of the algorithm used by this team can be seen in the picture.



The choosing of the methods was with the goal of covering a broad spectrum of approaches to extract keywords. Here is the list of the methods that we have employed with an brief explanation in brackets. For further reading, check the links: YAKE! [3] (statistical, unsupervised), TextRank [5] (graph-based, based on PageRank), TF-IDF[7], FirstPhrases[1], PositionRank [4], MultipartiesRank[2], SingleRank[9] and TopicRank [1] (graph-based, based on topics not single words).

Below we describe how out algorithm works: let be "n" the number of methods we use.

- For every comment we extract the 5 most relevant keywords using the methods described above. This step will return $5 \cdot n$ keywords. Notice that the limit of 5 keywords is a parameter, in the paper [6] they use other limits.
- Since every algorithm uses its own ranking scale we need to create a re-ranking method in order to decide which 5 keywords are the most relevant overall. We create a $5 \times n$ matrix where in every column we have the 5 keywords chosen by each method sort by relevance, i.e. with the ordering given by that method. Then, we create a list with the unique words that are in the matrix.
- We associate a vector to each of those unique keywords called $Rel_j[6]$, where $j = j^{th}$ unique keyword. We fill those 6 positions as follows:
  a) In the first position we compute the total occurrences of that keyword. So, $Rel_j[1] = \#$appearances of the keyword in the matrix.
  b) In the second position we compute the number of times that that keyword has been selected as the first option for any of the methods. So, $Rel_j[2] = \#$appearances of the keyword in the first row of the matrix.
  c) In the third position we compute the number of times that that keyword has been selected as the second option for any of the methods. And so on.
- We sort the unique keywords using the values $Rel_j[1], Rel_j[2], \ldots, Rel_j[6]$ in this order and in decreasing value. By doing so, we extract the top 5 keywords.
- Note: in case of a tie in all the parameters between two or more keywords, a random choice among those keywords will be done.

Conclusions and Future Directions

We run this algorithm in a personal laptop with hardware set-up: AMD Ryzen 7 5800H, 3.20 GHz and 16GB RAM. The average execution time for each proposal is 3.67 seconds.

- The use of multiple methods provides a better diversification for the search environment of the solutions, avoiding bias or falling into local maxima.
- For future iterations we consider the idea of creating a function that will give weights to the keywords given by each method and take that parameter into consideration to create the "Rel" vector.

[1] Algorith extracting the first phrases of a document as the most relevant ones.

Bibliography

[1] A. Bougouin, F. Boudin and B. Daille *TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction*, Proceedings of the Sixth International Joint Conference on Natural Language Processing (2013), 543–551.

[2] F. Boudin *Unsupervised keyphrase extraction with multipartite graphs*, Proceedings of the 2018 Conference of the NAACL: Human Language Technologies **2** (2018), 567–672.

[3] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes and A. Jatowt *YAKE! Keyword extraction from single documents using multiple local features*, Information Science **509** (2020), 257–289.

[4] C. Florescu and C. Caragea *PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents*, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (2017), 1105–1115.

[5] R. Mihalcea and P. Tarau *TextRank: Bringing Order into Text*, Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (2004), 404–411.

[6] J. Piskorski, N. Stefanovitch, G. Jacquet and A. Podavini *Exploring Linguistically-Lightweight Keyword Extraction Techniques for Indexing News Articles in a Multilingual Set-up*, Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation **509** (2021).

[7] R. Stephen *Understanding inverse document frequency: on theoretical arguments for IDF*, Journal of Documentation, **60** (2004), 503–520.

[8] D. Vega-Oliveros, P. Spoljaric, E. Milios and L. Berton *A multi-centrality index for graph-based keyword extraction*, Information Processing and Management **56** (2019).

[9] W. Xiaojun, X. Jianguo *Single document keyphrase extraction using neighborhood knowledge*, Proceedings of the 23rd national conference on artificial intelligence (2008), 855–860.