TÉCNICO LISBOA

Processamento e Recuperação de Informação — Grupo: João Santos n. º 67011

Course Project: Part1 — Paulo Anjos n. º 87822

03 November 2017 — Débora Abreu n. º 89242

1 Simple approach based on TF-IDF

We implemented an approach for selecting the most relevant sentences (top-3) to form a summary, based on TF-IDF vectors.

To select sentences we represented them in the vector space model and used the cosine similarity between the vectors. The document is segmented into its constituent sentences and the Natural Language Toolkit (nltk) was used to split the text into substrings using a regular expression. Our solution only considers one or more terms, excluding special characters. The method lower() was used to avoid repetitions (regarding uppercase and lowercase) of terms when we count or save them. We considered only one document. To store the frequency of terms we used a dictionary called invertedList. For each term, the invertedList stores the sentences where the term occurs and the corresponding frequency in the sentence. In the idf formula, N is the number of sentences and ni is the number of sentences where the term occurs in the document.

2 Evaluating the simple approach

We compared the procedure from the previous exercise, against a simple alternative in which the IDF scores are calculated with basis on the entire collection of documents in the TeMario dataset. The documents are segmented using Natural Language Toolkit (nltk) for Portuguese Language. To split a text into sentences we divided it using '\n\n' and then call the nltk sent tokenizer for Portuguese. Idf is now calculated in a different way, where N is the number of sentences in entire collection and ni is the number of sentences in the entire collection where the term occurs. *Precision* is calculated by the number of sentences in common between our resume and the extracted resume in the TeMario dataset, divided by the number of sentences in our resume. Global Precision is calculated by making an average with the individual documents precision. *Recall* is calculated by the number of sentences in common between our resume and the extracted resume in the TeMario dataset, divided by the number of sentences in the TeMario dataset. Global Recall is calculated by making an average with the individual documents recall. We calculated the F1 metric using global recall and global precision. Global MAP was calculated by making an average of the MAP of each document.

```
--- Metrics for 1st Exercise Simple Approach
Precision: 0.4279999999999999
Recall : 0.23974188964630141
F1 : 0.3073329091963022
MAP : 0.1846595313455607
--- Metrics for 2nd Exercise taking in to acount the entire collection
Precision: 0.42999999999999977
Recall : 0.23883250164867806
F1 : 0.30709624743349084
MAP : 0.17497253231517942
```

Figure 1

The only difference between both approaches is the calculation of the idf and as we can observe in *Figure1* precision is slightly increased in the 2st approach meaning that the idf that considers the entire collection is a better heuristic than the simple approach used in the exercise 1.

3 Improving the simple approach

In this exercise we consider not only individual words but also sets of 2 words as seen sequentially in the sentence (e.g., ["eu", "sou", "mortal", "eu sou", "sou mortal"]). Apart from this we also trained a tagger with the **Floresta Sinta(c)tica Corpus** such that it can extract noun phrases matching this grammar: "np: {(<adj>* <n>+ <prp>)? <adj>* <n>+}". Afterwards, any **stopwords**[1] found on the text were excluded. After obtaining the terms we proceeded to implement the BM25 term weighting heuristic towards scoring sentences. This formula takes on

the terms frequencies on each sentence, for a document, and normalizes it considering the average sentence length (average number of terms in the sentences of a document). For this part of the formula we left the values of **k1** and **b** as were shown in the instructions, considering they're usually set to those values. After computing this previous formula we multiply it by the logarithm of the division between: numerator, total number of sentences (or documents) in the collection minus the total number of sentences in which the term is found plus 0.5; denominator, total number of sentences in which the term is found plus 0.5.

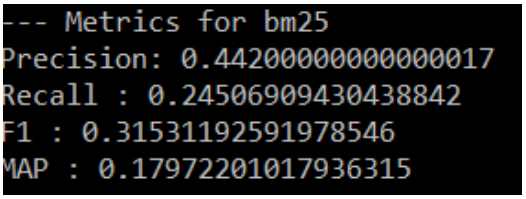

Figure 2

In the end we compared the results of this improved version against the one from exercise 2, and the results we obtained are shown in *Figure2*. Comparing both approaches (TF-IDF and Bm25) we can observer an improvement of precision indicating that Bm25 is a better heuristic than TF-IDF.

4 A more sophisticated approach

For this exercise we decided to use the BM25 as was implemented in the previous one, i.e., we used the previous exercise as a base for this one, considering it had the base components we needed.

To calculate the similarities of the sentence being evaluated by the MMR method we used BM25 as base for the sentence and document scorings, computing the similarity between the sentence being evaluated against the whole document and then subtract with the value obtained from the sum total of the similarities between the sentence being evaluated and all the sentences that were previously picked to be part of the summary. The sentence to be chosen in each iteration, until 5 are found, is the one providing the highest MMR value.

We afterwards compared the results of this approach against selecting just the 5 first sentences of a document and obtained the following results.

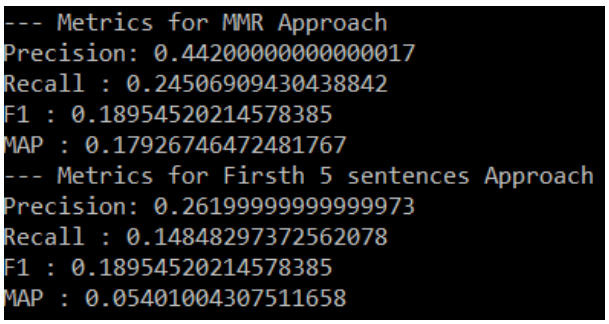

Figure 3

We noticed that when lambda is between to -1 and 1 the metrics obtained are equal to the Bm25 of ex3 and are the best that we measured. For values greater than 1 precision decreases drastically, and for values lower that -1 it has a small decrease.

[1] Note: nltk.corpus.stopwords.words('portuguese')