Group 19 - Automatic Single-Document Extractive Summarization

Pavlo Kovalchuk
MEIC-A
pavlo.kovalchuk@ist.utl.pt

Gonçalo Fialho
MEIC-A
goncalo.f.pires@ist.utl.pt

Pedro Duarte
MEIC-A
pedro.m.duarte@ist.utl.pt

1. INTRODUCTION

For each of the following exercises we use Python as programing language, and then we use the TeMario document set as data samples for each exercise. On the exercises 2, 3 and 4 we used only documents with title for the summarization and the ideal automatic summaries. We worked on a Windows environment so if ran in another OS the paths will need to be changed.

2. EXERCISE 1

2.1 Simple approach based on TF-IDF

In this exercise we implemented a simple approach for document summarization, based on a vector space model where each of the terms weights are calculated using the tf-idf algorithm introduced in the theoretical lectures. First we compute the tf (term frequency) for each sentence present in the document as well as for the entire document.

We then calculate the idf (inverted document frequency) for the entire document, lastly to obtain the final score (weight) for each term we multiply the tf by the idf.

We build the vector space model for each sentence present in the document as well as the model for the entire document based in the tf-idf scores calculated previously. Finally, we calculate the cosine similarity between this two models. In order to obtain the summarization of the document we choose the top 3 best scoring sentences.

2.1.1 Results

To test the correctness of this approach, we created an example document with some similar phrases and verified if the resulting summary was close to what we were expecting. We did not have any annotated corpus to evaluate with precision, however, we think that the results were within our expectation.

3. EXERCISE 2

3.1 Evaluating the simple approach

In the second exercise we apply the previous summarization method to the texts with title from the TeMario dataset, and evaluated the method against the automatic summaries of the TeMario dataset using some common measures, as explained in the next subsection. We also compared this approach with an alternative approach in which the IDF scores were estimated with basis on the entire collection instead of just the sentences. This time, we built summaries with the top 5 sentences ranked by their cosine similarity against the document.

First, we imported all the documents and target results (i.e., the automatic summaries) and pre-processed the documents by removing the punctuation. When separating the sentences, we also made sure to filter the punctuation but still allow it in some cases, like in the case of dates (i.e., 25-12-1995 or phrases inside phrases like "... ele disse "Vamos perder!"). Then we calculate the IDF scores based on the entire collection, and use this score when computing the model representation of both the sentences and the documents. The rest of the method is similar to the first approach.

To see which sentences in the summary are the same as the target results,

3.1.1 Measures

We computed the followed measures when evaluating the approaches:

- Precision
- Recall
- F1 Score
- AP (Average Precision)
- MAP (Mean Average Precision)

To compute this measures we used the same formula introduced in the theoretical lectures, we compare the obtained sumaries from our model against the provided summaries.

3.1.2 Results

Table 1: Exercise 2 Results

Simple approach		Alternative approach	
Precision	0.4580	Precision	0.2660
Recall	0.2541	Recall	0.1504
F1-Score	0.3210	F1-Score	0.1889
MAP	0.1810	MAP	0.0554

With this results [Table 1] we can conclude that the simple approach is better than the alternative one. We think this is because on the alternative approach the idf scores

are calculated over the all collection and the number of the documents is bigger (N) so the weights will be lower making important sentences less relevant than the other approach.

4. EXERCISE 3

4.1 Representing sentences in the vector space model

On this exercise we try to improve our simple approach implementation with 2 different representations of the sentences:

4.1.1 N-grams

We considered now a n-gram word to represent on the vector space model. NLTK[1] was used for the generation of the n-grams along with a self implemented algorithm to count the number of n-grams that the user specifies, this allows the program to be fully modular with different approaches since it can be used with single unigrams, unigrams and bigrams, or ngrams=(min,max) range. This also permits add to the model different long term words like chunks of words (presented in the next sub-section).

4.1.2 Chunks

On this part we created a portuguese sentence tagger to tokenize the words from the dataset. Since nltk does not have a portuguese tagger for this, we trained a model to make this possible with floresta dataset. After tagging the sentences we used a regular expression to get chunks of phrases that match the pattern that we chose:

$$\{(< adj > * < n.* > + < prp >)? < adj > * < n.* > +\}$$

and then added the awarded chunks to the model (if they were not there yet).

4.1.3 Results

Ngrams and Chunks

Precision	0.4460
Recall	0.2455
F1-Score	0.3112
MAP	0.1761

The results for this approach were worse than simple approach, this shouldn't happen but we think the fact there were no specified tagger to tokenize the sentences had a lot of impact on that. Since some chunks of words were not corrected tokenized, the length of the chunk was too small and the difference between having chunks or not almost imperceptible (since the chunks with length equals to the ngram range were not included). Furthermore the pre processing of the sentences may have direct influence on the results, making the simple approach a better one.

4.2 Scoring sentences - BM25

For BM25 term weighting heuristic we reused the code that we implemented for tf-idf approach and replaced the formula of BM25. Since our code was fully modular we were able to do this easily without much effort. The k_1 and b values were set 1.2 and 0.75 respectively since we did not find a better solution.

4.2.1 Results

Using BM25 term weighting

Precision	0.4280
Recall	0.2363
F1-Score	0.2992
MAP	0.1676

One of the properties of the BM25 model is that is prevents the saturation of the term frequency, i.e., the TF on BM25 reaches a point where it stops growing due to the asymptote defined through the variable k. In the normal TF-IDF, the term frequency can grow non-stop causing the TF-IDF scores to be higher than in the BM25. This may cause the similarity between the sentence models and the document models to be smaller, therefore the results are a bit more different from the target data.

5. EXERCISE 4

5.1 MMR

In this exercise it was proposed the usage of MMR [2] (Maximal Marginal Relevance) to re-rank the documents. This means that after ranking the documents using the cosine similarity like in the previous exercises, now we greedily choose the summaries that are both as similar as possible with the document and as different as possible with the summaries, so as to provide a balance between freshness and correct summarization. This balance is directed by the value of λ in the MMR function.

The optimal value of λ was found using a search. For a set of trial values, we evaluated how the algorithm would perform with each value and we chose the value that maximized the MAP function, as seen in figures 1 and 2. Note that the search could also be optimized with any kind of local search such as gradient ascent or simmulated annealing. We chose not to do so as we wanted to see how the evaluation evolved with the variation of the λ.

5.2 Results

Using 1-grams, TF-IDF and MMR

Precision	0.4530
Recall	0.2433
F1-Score	0.3104
MAP	0.1867

Approach that selects the first 5 sentences

Precision	0.2640
Recall	0.1495
F1-Score	0.1876
MAP	0.0545

We compared this approach with another simple approach that simply selected the first five sentences in the document as a summary. The results of the MMR approach were significantly better, although usually the first sentences of a news article are a good summarization of the whole article.

6. REFERENCES

[1] http://www.nltk.org/
[2] https://dl.acm.org/citation.cfm?id=291025

7. APPENDIX

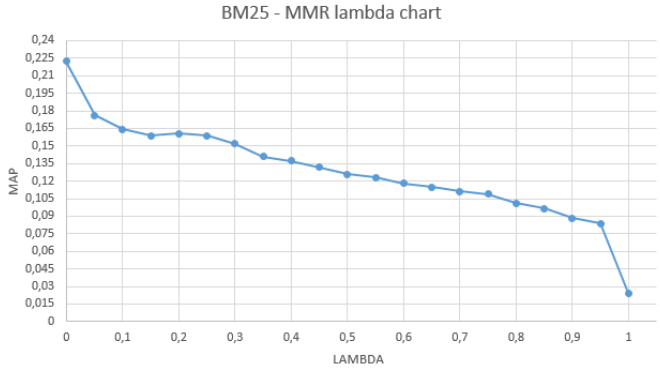

Figure 1: Evolution of the MAP with λ using BM25

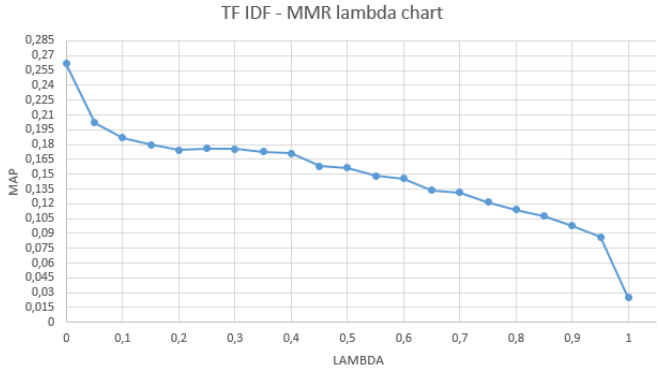

Figure 2: Evolution of the MAP with λ using TF-IDF