Project Report **PRI - PART 1**

Teammates: Patrcia Santos - 77091, Pedro Faleiro - 65007, Ricardo Sequeira - 79750
Group: 32
Date: 02/11/2017

Introduction

The presented document describes our work on the first course project for Information Processing and Retrieval, which aims to explore different alternatives for addressing the task of automatic single-document extractive summarization. In the following sections, our resolution for each of the proposed exercises is detailed.

Implementation

To guarantee the required vector computation methods were correct and to support all feature extraction approaches requested, a custom vectorizer was designed (iteratively and improved as needed) throughout the project. It is able to learn model parameters (fit method) and then return vector representations in that model (transform methods). Although it allows for easy feature addition and consolidates needed calculations, computational performance might not match that of external tools, specially on retrieving noun-phrases. To run each script, make sure your working directory is the project folder and use a python 3 interpreter.

Exercise 1

For exercise one, a simple approach for selecting the most relevant sentences for forming a summary, based on TF-IDF vectors should be implemented. The proposed process of summarization mimics an 'inverse' query approach, i.e., the feature space is first built from the parsed sentences of the document (using an Inverse Sentence Frequency in place of IDF) and the document is then modeled in the same space and matched against all sentences. The vectors are constructed using as score function a TF-IDF computation with the required normalized TF term and logarithmically scaled IDF/ISF. Similarity is, here and for the following exercises, given by the inner product between two normalized vectors (as it is a valid metric for vector distance in any dimension). To improve performance, a set of stop-words is used to remove less semantically valued words. Test is done on the 'catalunha.txt' file, being the summary the top 5 ranked sentences, ordered as they appear in the document.

Exercise 2

In this exercise, the previous approach is compared against one where training considers all documents (and so the traditional IDF). The summary consists again of the 5 most relevant sentences, ordered. Performance is then measured, using the provided ideal extracts to compute the precision, recall and F1 measures (as defined in the literature) on each document. The Mean Average Precision over the entire collection is also evaluated. Performance is, in general, better for the alternative approach.

Exercise 3

To try to improve the performance of the model from exercise 1, two additional types of features are extracted from the texts: bi-grams and noun-phrases. Bi-gram generation simply returns pairs of consecutive words in a sentence, without stop-words removal. To find all noun-phrases in a sentence, words first need to be tagged for they syntactic role. A 3-gram tagger with a 3 taggers' back-off chain is then trained on a corpus with similar characteristics of the evaluation corpora - MacMorpho - and saved for use in on-line sentence tagging. The tags used differ from those specified in the exercise description, so the expression {(<JJ >* <NN.*>+ <IN>)? <JJ>* <NN.*>+)} is translated to the equivalent (or at least less specific) formula {((<ADJ>* <N—NPROP>+ <PREP—KS>)? <ADJ>* <N—NPROP>+)}. (annotation manual found in the nilc.icmc.usp.br/macmorpho) Additionally, the score function used to compute vector weights is changed to the BM25 heuristic. Evaluation was only done with all the described changes applied and, although it beat the alternative model from exercise 2, the performance dropped compared to the simpler, initial model.

note: The extraction of noun-phrases is computationally heavy and so this process takes some time to complete.

Exercise 4

Here, the summarization method changes to try to conciliate relevancy and expressiveness when describing a document, i.e, tries to retrieve the most relevant sentences while maximizing dissimilarity between them. A sentence is selected at every step, which maximizes the MMR function provided, and taken from the evaluating set. The first 5 sentences retrieved form the summary. Several values were tested for the lambda parameter in MRR; lambda = 0,005 gave the best results. The model was built over the entire collection, following the alternative approach form exercise 2, and performance was measured against the first five sentences in each document (title excluded).

Conclusion

Automatic summarization is a hard challenge that relies heavily on Natural Language processing methods (together with available tools and datasets) to solve efficiently. Additionally, when using a vector space model, it has a strong dependency on both the score function and similarity computation methods, and so, their choice should be done carefully. Regarding this project and in spite of the obtained performance, the solution developed provided us an insight of how much work is needed to build robust tools that efficiently address this problem and are able to handle large amounts of data.