Information Processing and Retrieval Project Report – Part 1

Instituto Superior Técnico – Universidade de Lisboa

Margarida Costa
83425

Marta Aparício
83525

Paulo Alves
83538

ABSTRACT

This report focuses on automatic keyphrase extraction, applied to two datasets: the 20-newsgroup collection and the Inspec dataset. To accomplish such task the TF-IDF and the BM25 information retrieval model are used, followed by an evaluation of the models. Lastly classifiers SVC, logistic regression and SGD were applied.

Introduction

To address the problem – automatic keyphrase extraction – two main alternatives are explored: a simple approach based on TF-IDF; and a supervised approach.

The simple approach based on TF-IDF has three phases: First is implemented a simple baseline approach, that applies the keyphrase extraction method to an English textual document; Secondly its implementation is evaluated, based on metrics such as precision, recall, F1 measure, mean value for the precision@5 and mean average precision; Lastly, is improved by candidate selection, candidate scoring and other ideas where tested.

The supervised approach resorts to classification algorithms and a set of features position of the candidate in the document, number of words in the candidate, number of characters and TF-IDF. The results are then evaluated by the mean average precision and a confusion matrix.

The 20-newsgroup collection is only used for the baseline implementation of the simple approach based on TF-IDF. For the rest of approaches, the Inspec dataset is used.

1 Simple approach based on TF-IDF

1.1 Implementation

The libraries "sklearn" and "scipy" were used, together with the documents from the 20newsgroups collection. Only one document is being considered for test and the train was made over the whole 18000 documents and for just 30 documents, with the purpose of analyzing the differences.

When ***TfidfVectorizer*** is being applied to the training set, the conversion of raw text to a TFIDF features matrix is being made. For better fit the purpose of the case study the following parameters were changed: ***ngram_range(1, 3)***; the ***stop_words='english'***; the ***token_pattern=r"(?u)\b[a-zA-Z][a-zA-Z-]*[a-zA-Z]\b"*** that selects only single words or words followed by a hyphen and other words followed by hyphens. Lastly the ***max_df***, ***min_df*** and ***norm*** will be used on the exercise 2 for fine tuning. After applying the ***fit_transform*** method, to all the documents, the vocabulary and the IDF scores are computed based on all the documents allowing for a full coverage of all words, making sure that the vocabulary also contains the words of the test document. Once this is done the ***transform*** method is applied to the test document alone, making a TF-IDF-weighted matrix, where the line is the test document and the columns are the terms of the model. Then, the scores of each term of the matrix are calculated by multiplying the TF-IDF value of the term by the number of words that it has (e.g. if the term is a tri-gram it would be *TF-IDF value* 3*) or by the number of characters that the words have. Finally, the results, from this operation, are sorted and only the 5 most important results, thus candidates, are returned.

1.2 Results

The success or unsuccess of this exercise is undetermined since it is not possible to apply metrics. For the calculation of the final scores the multiplication by the number of words of the n-gram was the chosen option, since it gives more importance to the grams that have more words, thus the grams that will probably describe better the document. The usage of the whole set of train documents played an important role on the results, since they would be more general when all the 18000 documents were included ([‘cms udel edu’, ‘phys psu edu’, ‘dev-null phys psu’ ‘espn’, ‘bed angry’]). However, when only one category of that set was used, the results included more words that defined better what the actual test document was about (['espn', 'psu edu', 'organization penn state', 'hockey', 'run']).

2 Evaluating the simple approach

2.1 Implementation

For evaluating the previous retrieval model, these documents are split into two sets, one train (75% of the documents) and other test (25% of the documents). After this sets are cleaned of xml tags; the train dataset is fitted using the same approach then before and the candidates keyphrases are calculated from the test dataset. Then the target keyphrases are obtained, so the predicted candidates can be compared and evaluated. The evaluation is done by the ***metrics(y_true, y_pred)*** function, that calculates the f1-measure and the precision and recall measures, followed by the calculation of the mean average precision (MAP) value and the mean precision@5 (MP5) for the entire test collection.

2.2 Results

Different approaches were tested. First, an approach where ***TfidfVectorizer*** used the original word from (e.g. studying) the text in order to calculate the candidates, was implemented. Then this was changed so the ***TfidfVectorizer*** used the word lemmas (e.g. study) instead, and the obtained results improved in comparison to the previous approach. This, because the words were reduced to the form of the word that is chosen by convention. Lastly, instead of the utilization of lemmas, the word stems (e.g. stud) were calculated and used by the ***TfidfVectorizer***.

After each approach, fine tuning was done with the ***max_df***, ***min_df*** = 1 and ***norm*** = 'l2' parameter of the ***fit_transform*** method, improving some of the results.

Table 1 - MAP and M@5 result for word, lemma and stem

	Word (max_df = 3)	Lemma (max_df = 2)	Stem (max_df = 9)
MAP	0.122	0.112	0.167
MP5	0.054	0.036	0.060

The results prove that is better to use stems to lemmas or words. Nonetheless, is interesting to see that presents better results for words that for lemmas. This may be due to the fact that Inspec is constituted only by abstracts. This translates in short documents where in fact the keyphrase appear however, they are almost never repeated.

3 Evaluating the simple approach

3.1 Implementation

The extraction of candidates from the xml is similar to the implemented in exercise 2. The differences relies on the generation of the grams, that in this case is made by the ***ngrams*** function provided by *nltk* and on the selection of the candidates by forcing that all candidates obey to the ***grammar*** = {(<JJ>* <NN.*>+ <IN>)? <JJ>* <NN.*>+}. The tags of the terms are obtained with *nltk*. After the documents are preprocessed, a BM25 object is created where a matrix with lines representing a document and columns representing terms is filled with the corresponding IDF part of the BM25 formula. In this computation a creative improvement was made, for all the terms that were contained in more than half of the documents its IDF score is reassigned to ***EPSILON*** = 0.25. The ***get_scores*** will compute the final computations of the BM25 formula (TFIDF). The free parameters considered were: ***b*** = 0.75 and ***k1***= 1.2. Based on this final matrix the predictions of keyphrases are made and used to compute the same metrics as in exercise 2.

3.2 Results

Table 2 - MAP and M@5 results considering or not candidate length for word, lemma and stem

	Word	Lemma	Stem
Candidate Length	map: 0.28 m@5: 0.13	map: 0.22 m@5: 0.09	map: 0.05 m@5: 0.01
No Candidate Length	map: 0.27 m@5: 0.13	map: 0.22 m@5: 0.09	map: 0.05 m@5: 0.01

The results extracted show that using the words returns the best results, even though the lemmas present similar results. However, this goes against the expected. Normally stem and lemmas are preferred, since they reduce different grammatical forms and reduce various suffixes from a word to get its common origin.

This may be because the dataset has small documents with small coverage of terms in each, meaning a small search space. When comparing this, for instance with the SemEval-2010, that was originally used, where the full papers are used, the keyphrases are repeated more often along the text, offering better results. Due to the nature of the Inspec dataset that contains only abstract, the BM25F algorithm doesn't seem to offer a great improvement once the abstracts already are a summary of the papers and have more keyphrases.

The grammar presented doesn't cover all the referenced keyphrases, for instance "changing practices" would never be a candidate since contains a verb.

4 A supervised approach

4.1 Implementation

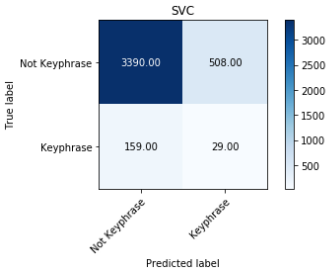
The dataset was preprocessed to lemmas, and words. Candidates were generated with attributes with the attributes mentioned in the Introduction. The model was trained with 3933 candidates removed from 75 documents. It was tested with 3223 candidates removed from 25 documents. There was few candidates keyphrases along the document. This would provoke the model not to learn what a keyphrase would look like because he trained with few examples of keyphrases and a lot of not keyphrases. To overcome this, the train set was balanced: it was chosen randomly candidates who were not keyphrases and removed from the train set.

4.2 Results

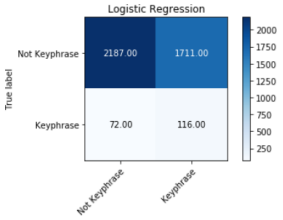
There was not a significant difference when tfidf was removed from the set of attributes of the candidate. The following metrics were tested with no tfidf as attribute and the dataset preprocess with lemmas. Logistic regression and SVC show best results when the class weight is balanced.

Table 3 - Mean average precision results for classifiers using lemmas

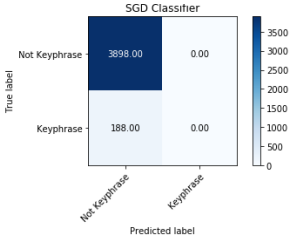
	SVC	Logistic Regression	SGD
MAP	0.144	0.141	0.141



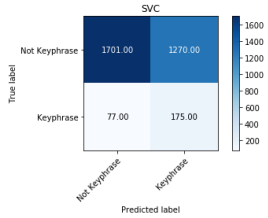
Matrix 1 – Confusion Matrix for classifier SVC using lemmas



Matrix 2 – Confusion Matrix for classifier Logistic Regression using lemmas



Matrix 3 – Confusion Matrix for classifier SGD using lemmas



Matrix 4 - Confusion Matrix for classifier SVC using words

The results do not vary very much in MAP. The logistic regression has the most true positives. SVC and SGD have more true negatives, but are unable to classify keyphrases correctly.

When using the dataset preprocessed with words, the results were better only for the SVC classifier (MAP = 0.27).