TD1 - Knowledge Extraction from text

Nicolas AUDOUX - Thomas PAUL - Yana RAGOZINA

SI5 IA-ID

Introduction

In this lab we will be focusing on performing a keyword/keyphrase extraction analysis (KPE) on a datasest of documents. Therefore, we will try to build and evaluate different document summaries generated by different keyphrase extraction algorithms.

We will try to implement 3 different keyphrase extraction algorithms in order to analyse and compare their functionalities and performances.

We will use one of the datasets designed for automatic keyphrase extraction, *Inspec*, collecting 2000 abstract documents in English language in the domain of Computer Science.

As for the tested KPE algorithms, we chose to analyse *PositionRank*, *SingleRank* and *TextRank*.

The performances will be evaluated with ROUGE framework (Recall-Oriented Understudy for Gisting Evaluation), designed to compare an automatically produced summary by the algorithms against a reference.

1. **Algorithms used :**
   We chose to implement 3 KPE algorithms: PositionRank, SingleRank and TextRank. A description of each algorithm is presented below:

   - PositionRank :

   Position Rank extracts keyphrases by determining the importance of a word based on its position in the document.

It's an unsupervised algorithm that is decomposed like this :

1. Calculates the Term Frequency of a word(TF)
2. Adjusts the term frequency based on the length of the document (Document Length Normalization)
3. Assigns scores to words based on their positions within sentences. **Words in the beggining and end of sentences have higher scores.** (Sentence Position Score)
4. Combines the term frequency and sentence position scores to determine the overall importance of each word (Sentence Salience Score)
5. Extracts words that have the highest salience scores (Keyphrase Extraction)

- SingleRank :

SingleRank is a graph-based summarization algorithm. Therefore, it performs in the first place a graph representation of a given document where each sentence is represented as a node. Plus, this algorithm measures the degree of similarity between each pair of sentences of the document. Then, an importance degree (centrality score) is assigned to each sentence, allowing a more precise analysis for building a summary. The further process of the summary generation can be seen as follows :

- Calculating the similarity between each pair of sentences in the document, often based on the content overlap
- Mapping the similar sentences as nodes to the graph where the measured similarities are used to build edges between the nodes.
- Measuring the degree of importance of each node (sentence) within the graph
- Ranking each sentence in descending order based on the previously calculated degree of importance (centrality score). Sentences that are frequently visited or connected to other important sentences receive higher centrality scores
- Generating the summary of the document based on the highest-ranking sentences to form the summary

-TextRank :

TextRank is an algorithm that identifies keywords by assessing their significance within a connected graph. It functions by analyzing the relationships between words or phrases to determine their importance in the context of the overall text. Here is he algorithm:

- Tokenization and part of speech tagging
- Reducing the number of words based on a syntactic filter (in our case we keep only noons propositions and adjectives)
- With all the remainig words are added to the graph and an edge is craeted for every words that co-occur in a window of N words (in our case, N=10)

At this point we have an undirected unweigth graph.

- Then a initial value of 1 is set for every vertice
- Finally a modify version of the PageRank algorithm is run to upgrade the vertice score. The main idea behind this algorithm is to give more importance to a word which is linked by many others. Moreover a link to word which is linked by many other is more important than a link to word which is linked to only one word. This is the same algorithm used to rankes web pages. The only difference is that we also use a weight to each wich corresponds to the co-occurence score
- After that, we keep only a a third of our vertices which corresponds to the vertices which have the highest score.
- A post processing is done on the remainng vertices and if two words appears next to each other in the document a multi-word keyword is created.

2. **Best ROUGE score :**

From the 3 keyphrase extraction algorithms (Position Rank, Single Rank and Text Rank), Single Rank has the best ROUGE score.

3. **Extracted keyphrases evaluation :**

Overall, the extracted keyphrases seem to be semantically correct as the algorithms managed to capture the most pertinent words from the documents. However, these

words don’t often seem to be the best solution for retrieving the main context of the document as they sometimes fail to generalize the essence and the subject of each document, capturing only some frequent words from the document. For example, the second best keyphrase candidate by SingleRank for the following document was *“such services”,* which is not a suitable solution for a summary :

“Waiting for the wave to crest [wavelength services]
Wavelength services have been hyped ad nauseam for years. But despite their quick turn-up time and impressive margins, such services have yet to live up to the industry's expectations. The reasons for this lukewarm reception are many, not the least of which is the confusion that still surrounds the technology, but most industry observers are still convinced that wavelength services with ultimately flourish”

For notice, the first best keyphrase candidate for this document is *“wavelength services”*, which seems to be an optimal summary.

4. **Runtime comparison :**

We note the following runtimes for KP extraction for each algorithm :

- PositionRank:
- SingleRank:
- TextRank:

5. **Best algorithm analysis**

Our tests have shown the best ROUGE score for SingleRank algorithm.

All the 3 tested algorithms are unsupervised, graph-based solutions for KPE. However, we may suggest that SingleRank was the most suited solution for our dataset due to several factors. First of all, PositionRank is mostly based on calculating a Term Frequency score. As far as the documents in the dataset are mostly very short, this algorithm might fail to calculate the most frequent words as there may be often no frequent words.

On the other hand, TextRank performs tokenization and speech tagging. As far as Inspec dataset contains abstract documents in the domain of Computer Science,

tokenization may be inaccurate for this domain. Plus, this algorithm performs reduces the number of words in each document. Having mostly very brief documents, we can suppose that reducing the quantity of words may not be the optimal solution.

Nevertheless, SingleRank, having the best ROUGE score, still has some insufficient results. We suppose that this may be due to the inaccurate measurements of the similarity degrees between each pair of sentences. This issue may come from our dataset and from the way the sentences are formed in each document. Thus, the centrality score may have been the best-suited metrics for our dataset.

6. **Keyphrase Knowledge Graph**

For representing each document and its respective extracted keyphrases in the form of a knowledge graph, we suggest the following structure :

- Low-level : each document would have a graph with nodes representing the extracted keyphrases.
- High-level : every documents would be represented as 1 node and would be linked by their predominant keyphrase extracted.

We can use the vocabulary of the extracted keyphrases of all the documents.

To represent this as a graph, we would have each document and each keyphrases as node. If a key phrase represents a document, we connect each other with a weighted link to measure the level of confidence. The vocabulary used can be all the extracted keyphrases. To have a more interesting graph, we could also compute the similarity between differents keyphrases to link them or link every keyphrases to a general topic.