ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Natural Language Processing

Graph-Based Keyword Extraction from Scientific Paper Abstracts using Word Embeddings

Author:
Dinno Koluh

Supervisor:
Prof. Paolo Torroni

Co-Supervisor:
Dr. Federico Ruggeri

Bologna,
October 2023.

Abstract

In the era of information overload it became essential to efficiently extract concise, precise and quality information from large texts. One aspect of information extraction is keyword extraction where large texts are represented as sets of tokens i.e. keywords. This prospect of keyword extraction is paramount to researchers as they deal with huge numbers of scientific papers, and having a good and concise representation of those papers is essential for them. This thesis paper addresses that problem in the realm of natural language processing (NLP).

Using core concepts of NLP and modeling texts as graphs, we are going to build a model for the automatic extraction of keywords. This is done in an unsupervised manner as the importance of a word is calculated through the position and weights associated with respective words in the graph. The first metric used to calculate the graph weights are co-occurrence matrices and the other metric are word embeddings. Word embeddings became a crucial way of representing the semantic information of words as dense vectors.

The results of this paper were compared with keywords that were provided by authors of scientific papers in the area of computer science which act as the ground truth, but crucially are not a component in the model construction, but just serve as a verifier of the model's accuracy.

Keywords: NLP, keyword extraction, scientific papers, graphs, word-embeddings

Contents

List of Figures

1 Introduction and Motivation

Natural Language Processing (NLP) is a subfield or Artificial Intelligence (AI) and linguistics that has a focus on the interaction between computers and human languages. This is mostly restricted to written language as other fields (like Speech Processing) deal with spoken language (using audio instead of textual features). In the past decade NLP has seen huge attention with the introduction of some key concepts like *word-embeddings* [1] (which we are going to use) and *transformers* [2]. At the moment of writing of this paper, NLP is the subfield of AI that has the most resources invested into it, mostly in research and development, with new large language models (LLMs) coming out on a daily basis. Our focus will be a bit shifted from Deep Neural Network (DNN) models and more to traditional Machine Learning (ML) models. NLP has many subfields and application areas such as:

- Text classification
- Information retrieval
- Automatic translation
- Speech analysis
- Question answering
- Conversational agents
- Sentiment analysis

The field we are going to be working on will be **information retrieval**, more precisely *keyword extraction*. Keyword (keyphrase) extraction is the automatic selection of important and topical phrases from the body of a document [3]. Scientific papers are usually the area where keywords are frequently used as researchers use them for a quick overview of papers and also sorting papers into different categories. This is the topic we are going to be working on as well. The keywords are going to be extracted from the abstracts of the scientific papers. One detail we should address that will be important later is the distinction between *keywords* and *keyphrases*.

Keywords would be single words or at most MWEs (Multi-Word expressions, i.e. deep learning) while keyphrases are more complicated entities comprised of several words and they act as a single unit (e.g. the phrase "scientific paper" would be a keyphrase). When doing keyword extraction we might have as an output a combination of both, keywords and keyphrases. Usually keyphrases carry more information than keywords but a combination of both as an output is most representative. From now on, when referring to *keywords* it will also include *keyphrases*, if not explicitly stated otherwise.

The model to be used for keyword extraction is **graph-based**. The idea is to model words in a paper abstract as nodes of a graph. Not all the words in the abstract should be included, as keywords are usually composed of a combination of nouns and adjectives (e.g. scientific[ADJ] paper[N]). This means that some preprocessing of the raw text will be required, especially when dealing with MWEs. We will speak about this in the next chapter.

As stated, the nodes of the graph will be modeled as words, but the question comes on how to model the edges of the graph?

It starts from the assumption that words that occur in the same context tend to carry a similar meaning. The idea is to use a **sliding window** of some predefined size and words that are in that window are connected with and edge. In that way the final graph carries semantic (the meaning of words) information of the input text. The number of occurrences in when sliding the window over the entire text will give us the initial graph weights. To solidify this relationship another metric that we will use are word embeddings. Word embeddings are essentially a way of representing words as vectors of numbers. We will dive more deeply into how word embeddings are computed ans used but for now we can assume that word embeddings are vectors of numbers and that these vectors carry semantic information of the corresponding word. This means that words that have a similar meaning (whatever that might mean in some general picture) also have similar vector representations and that we can use the usual mathematical tools on these vectors like the dot product which means that we can actually measure the similarity between words.

We now have the complete representation of the input text as a graph. Now we need to somehow rank the nodes according to the graph edges and weights. We can refer to the ranking procedure as to finding the importance of each node. There are several algorithms which can

find the importance of nodes, and we will speak in more detail about them later, but for now can assume we have a black box which takes in the graph representation of the abstract as described before and gives us all the nodes (word) ranked by their importance.

We now have a list of the most important words, but there is one more step that we can do to get a more robust output. The given output would be a list of words, but we might want to have phrases instead of keywords (e.g. the phrase "scientific paper" is more representative instead of just "paper"). We can use the words we got out from the graph and traverse the initial text for phrases in which they appear. Then based on those phrases in the initial text and the importance of the words which they are made of, we can get alongside the ranked words also the phrases in which they appear. In this way we can also generalize the problem by letting a phrase be only made up of one keyword. If it is made up of more than one keyword we can average it out, and in the end get the ranking of the specific phrases which appear in the input text.

This was a rough explanation of the procedures which should give us an overview of the steps involved in developing a pipeline for the extraction of keywords from paper abstracts. In the next few chapters we are going to look in more detail in the inner workings of the steps involved. We will start with the NLP preprocessing pipeline.

2 NLP Pipeline

3 Word Embeddings

4 Graph Construction

5 Implementation of Keyword Extraction

6 Testing, Results and Discussion

7 Conclusion

8 Literature

[1] Almeida, Felipe, and Geraldo Xexéo. "*Word embeddings: A survey*.", arXiv preprint arXiv:1901.09069 (2019).
URL: https://arxiv.org/pdf/1901.09069.pdf

[2] Lin, Tianyang, et al. "*A survey of transformers*.", AI Open (2022).
URL: https://www.sciencedirect.com/science/article/pii/S2666651022000146

[3] Zu, Xian, Fei Xie, and Xiaojian Liu. "*Graph-based keyphrase extraction using word and document embeddings*.", 2020 IEEE International Conference on Knowledge Graph (ICKG). IEEE, 2020.
URL: https://ieeexplore.ieee.org/abstract/document/9194571

[4] Jurić Ž. "*Diskretna matematika za studente tehničkih nauka*". ETF Sarajevo, UNSA, 2017.

[5] Michael Yoshpe (2020). Distance from points to polyline or polygon.
MATLAB Central File Exchange. Retrieved December 11, 2020.
URL: https://www.mathworks.com/matlabcentral/fileexchange/12744-distance-from-points-to-polyline-or-polygon

[6] `lsqlin` function description. Solve constrained linear least-squares problems.
URL: https://www.mathworks.com/help/optim/ug/lsqlin.html