UNSUPERVISED KEYWORD EXTRACTION FROM GREEK BIOMEDICAL TEXT USING BERT

Konstantinos Giantsios
M.Sc. Data and Web Science
Aristotle University
54124 Thessaloniki, Greece
giantsik@csd.auth.gr

Stratos Grigoroudis
M.Sc. Data and Web Science
Aristotle University
54124 Thessaloniki, Greece
egrigorou@csd.auth.gr

ABSTRACT

BERT has recently emerged as a very effective language representation model. BERT is conceptually simple and empirically powerful. In this paper, we gather a novel dataset from biomedical greek websites and produce word embeddings using the fine tuned BERT model for the keyword extraction from the specific domain of greek biomedical texts. Experiments and evaluation conducted on already existing unsupervised keyword extraction methods compared to our approach shows that BERT can learn from greek biomedical texts. Code is publicly available at: https://github.com/CoGian/keyword-extraction-with-greekBERT and our fine tuned model is available at: Google Drive.

KEYWORDS

keyword extraction, unsupervised, embeddings, BERT transformer

1 Introduction

With the growing amount of biomedical information available in textual form, there have been significant advances in the development of pretraining language representations that can be applied to a range of different tasks in the biomedical domain, such as pre-trained word embeddings, sentence embeddings, and contextual representations. Keyword extraction is a text analysis technique that automatically extracts the most used and most important words and expressions from a text. It helps summarize the content of texts and recognize the main topics discussed and for easier information retrieval of a document and as a pre step in other information extraction tasks. The unsupervised techniques for keyword extraction are divided into 4 main categories : Statistics based ,Graph-Based, Embeddings-based and Language model-based. Yake! (Yet Another Keyword Extractor!) is a novel feature-based system for multilingual keyword extraction from single documents, which supports texts of different sizes, domains or languages. Follows an unsupervised approach which builds upon features extracted from the text, making it thus applicable to documents written in many different languages without the need for external knowledge [1]. Rapid Automatic Keyword Extraction (RAKE) is a well-known method of extracting keywords that uses a list of stopwords and keyword phrases to detect the most relevant words or phrases in a piece of text [2]. TextRank, is a graph-based algorithm that is based on graphs, which means that the primary data model used is a graph. It is an innovative unsupervised method for keyword extraction and works in a similar way with PageRank, an algorithm used by Google to determine the relevance of a web page to other web pages [3]. Our approach is using embeddings produced by BERT

(Bidirectional Encoder Representations from Transformers) model to implement unsupervised keyword/keyphrase extraction.

2 Dataset

First of all, we would like to describe how we managed to create a novel dataset from Greek biomedical texts. We followed the technique of scraping on Greek websites that host biomedical texts and magazines such as mednet.gr, sebe.gr, or iatrolexi.gr [4]. It is worth noting that it was quite a difficult process due to the limitations of some websites, such as the digital library of Aristotle University or in general the structure of the Greek websites did not help in extracting the information we were looking for. Our purpose was to collect titles, abstracts and keywords. We used rule-based methods taking into account the html tags used by websites to retrieve useful information. We collected from mednet 1164 abstracts with their keywords and titles. From sebe we collected 361 abstracts with their keywords and titles.

- Example: IF (html_tag= span.AbsText::text) THEN (keywords = the_included_text)

The iatrolexi.gr website also had unstructured text, but we managed to edit it, using simple rules such as: If a word belongs to a list of possible abstract words then this text is abstract until we come across some possible stopword.

- Example:IF word in set_of_possible_abstract_names THEN abstract=following_text_until_a_possible_keyword_name

In this way we managed to collect 382 abstracts with their keywords from and enrich the iatorlexi corpus. In total, we managed to collect 1907 abstracts along with their keywords and titles.

As a next step, we had to pre-process the data we collected. Essentially, our goal was to convert them into a format that is predictable, normalized and can be analyzed for the task we have chosen. Initially, we combined the corpus that we collected from the Greek medical websites. Pre-processing included lower word casing, tab removal, newline characters, html tags or latex tags using regular expressions. Next, we divided the dataset into train and test datasets. For the training we used 1715 examples which was about 90% of the total dataset which included abstracts, keywords and titles. The remaining 10% was used for testing. Finally, we created another dataset which contains all the sentences from the train dataset summaries using the spacy module. The total number of sentences collected in this dataset was about 15000 sentences.

3 Method

3.1 BERT & Model Fine Tuning

The model that we use for embeddings production is the BERT model, whose initials are (Bidirectional Encoder Representations From Transformers). BERT is a bi-directional transformer model that allows us to convert phrases and documents into vectors that capture their meaning. BERT's key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training [5].

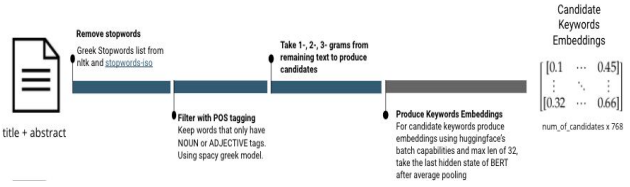
Before using the embeddings produced by BERT we try to do the fine tuning model on the sentences dataset [6]. The purpose of this fine tuning is different from fine tuning in downstream tasks because we aim to teach the model the semantics of a specific domain. We used the basic version of Greek BERT [7]. The task we use is the Masked Language Modeling and the training lasted 4 epochs and took place in the Google Colab environment. Masked Language Modeling is a fill-in-the-blank task, where a model uses the content words that surround a mask token to try to predict what the masked word should be. Our strategy was to randomly cover 15% of the total token for each sentence.

3.2 Extraction process

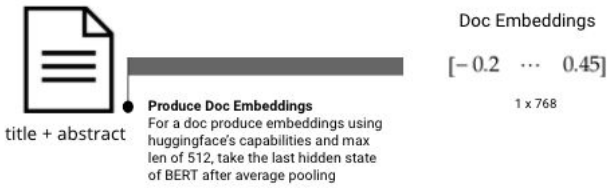
Our approach to extracting keywords or keyphrases using embeddings [8] in addition to BERT fine tuning includes 3 more stages:

1. Candidate keywords productions: the candidates are produced from a text, an abstract and a title in our case, taking n-grams.
2. Embeddings productions: produce embeddings for the text and the candidates using the fine tuned BERT or simple BERT.
3. Candidate keywords ranking: rank candidates taking into account to be diverse enough [9][10].

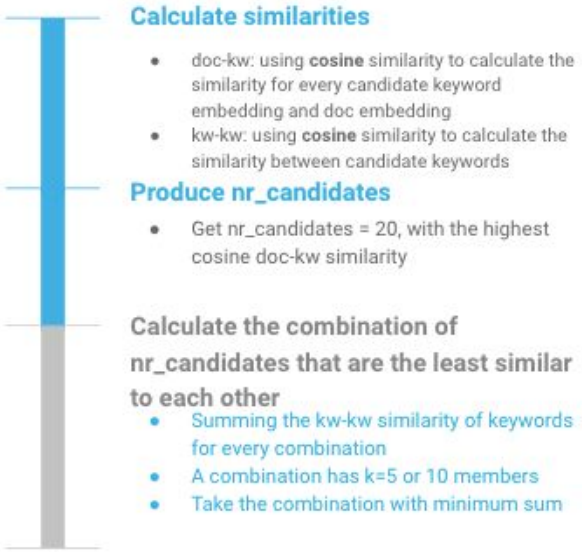
To produce candidate keywords, given a document, we first combine the title and summary of the document. Then we remove the stop words using the Greek lists for stopwords of nltk and stop words-iso. The next step is to keep only the words that are nouns or adjectives, using the Greek model of spacy for part of speech tagging as recommended by some other approaches [11]. Finally we get unigrams, bigrams and trigrams from the remaining text. For the production of keyword embeddings we use the capabilities of Huggingface [12] for batch forward pass of the keywords in the BERT model. In addition, a max length of a sequence equal to 32 is used. And at the end we get the last hidden state of BERT after applying average pooling, and we produce a two-dimensional table where each line is also a keyword.



Huggingface's features are again used to generate embeddings for the document. This time the text is not further preprocessed and is just the title attached to the abstract. Using max length of a sequence equal to 512 and again we get the last hidden state of BERT after applying average pooling, and we produce a two-dimensional table where in essence in the first dimension there is only one element.



In the last stage where the keywords are in order, the goal is to maximize the similarity between the keywords and the document but at the same time to minimize the similarity between the keywords for differentiation. The cosine similarities are first calculated using candidate keywords embeddings and doc embeddings to find the similarities between the keywords and the document. And then only the keywords embeddings to find similarities between the keywords. Then we get the 20 nr_candidate keywords with the highest similarity to the text. Finally we must calculate the combinations of the 20 keys with the least similarity between them. Each combination has k members, we add the similarities between the k members, and we get the combination with the smallest sum where the exported keywords are.



4. Evaluation

For the evaluation process we use the test set we talked about earlier, that has 192 samples with abstracts, titles and keywords. The system we use is the google colab environment with a TESLA T4 GPU and an Intel Xeon CPU. We followed a partial matching approach, because it is difficult for an unsupervised approach to extract the keywords that have the same syntax as the gold keywords [13] [14].

So we consider it as a match if a gold keyword contains one or more tokens of a predicted keyword. A gold keyword can only match one predicted keyword and vice versa. In addition we used stemming as a preprocessing step. We use precision@k, recall@k and f1@k.

model	precission@5	recall@5	f1@5	inference time
bert-base-greek	35.4	42.2	38.5	~1 second
bio-bert (our bert)	34.8	41.5	37.8	~1 second
YAKE	24.6	29.3	26.7	~0.02 second
RAKE	23.9	23.2	23.5	~speed of light
TEXTRANK	29.4	34.0	31.5	~0.08 second

In the upper table we calculate metrics for k equal to 5. It seems that our approach using BERT without fine tuning has the best results, while 2nd comes our approach again but using the fine tuned model.

model	precission@10	recall@10	f1@10	inference time
bert-base-greek	23.0	54.9	32.4	~9 second
bio-bert (our bert)	23.3	55.5	32.8	~9 second
YAKE	18.0	42.9	25.3	~0.02 second
RAKE	18.3	27.8	22,1	~speed of light
TEXTRANK	24.6	54.3	33.8	~0.08 second

In this table we calculate the metrics for k equal to 10, and first comes the textrank method, while 2nd comes again our approach using the fine tuned model and having the best recall@10.

Using only metrics it is not enough to evaluate a keywords extraction process. For this reason we need a quality evaluation.
Title: “Ενδοκρινικοί διαταράκτες και θυρεοειδική λειτουργία“
Gold Keywords: ["ενδοκρινικοί διαταράκτες", "θυρεοειδής", "καρκίνος θυρεοειδούς", "μεταβολισμός θυρεοειδικών ορμονών"]

model	predicted keywords

bert-base-greek	['θειοκυανιούχα νιτρικά ιόντα', 'θυρεοειδική λειτουργία ενδοκρινικοί', 'πολυχλωριωμένα διφαινύλια διφαινύλια', 'ενδοκρινικοί διαταράκτες θυρεοειδική', 'θυρεοειδικής υπεροξειδάσης ισοφλαβόνες']
bio-bert (our bert)	['θυρεοειδική λειτουργία ενδοκρινικοί', 'ορμονών πολυχλωριωμένα διφαινύλια', 'υπερχλωρικά θειοκυανιούχα νιτρικά', 'υπεροξειδάσης ισοφλαβόνες σύνδεση', 'θυρεοειδικής υπεροξειδάσης ισοφλαβόνες']
YAKE	['ορμονικού συστήματος τόσο', 'και', 'τόσο του ανθρώπου', 'διφαινύλια', 'παρεμβαίνουν στην ομαλή']
RAKE	['φυσικές ή συνθετικές', 'την παιδική ηλικία', 'σε ορισμένες περιπτώσεις', 'στο κεφάλαιο αυτό', 'το νερό']
TEXTRANK	['ενδοκρινικούς διαταράκτες', 'ενδοκρινικοί διαταράκτες', 'διαταράκτες', 'και θυρεοειδική λειτουργία', 'ανάγκη υλοποίησης']

As you can see our approach in the first 2 lines produces relatively good keywords / key phrases with a differentiation. In contradiction Textrank method produces similar keywords. RAKE and YAKE don't produce any useful keywords.

5. Conclusion

In conclusion, BERT is able to learn from biomedical Greek texts, as we saw when increasing the k from 5 to 10. Our approach produces quite good and differentiated keywords in contradiction with Textrank. More model training would help but that means more time and data. In the future we would like to include other methods such as KPMINER. It would also be good to have a strict match evaluation process. Of course we need to find out if there are even better values for nr_candidates and max length sequence. Finally we would like to see if we can use BERT in a supervised approach.

REFERENCES

[1] Ricardo Campos , Vítor Mangaravite , Arian Pasquali , Alípio Mário Jorge , Célia Nunes , and Adam Jatowt , "YAKE! Collection-Independent Automatic Keyword Extractor, 2018 OriginalPaper, Springer International Publishing

[2] Stuart Rose, Dave Engel, Nick Cramer, " Automatic Keyword Extraction from Individual Documents",March 2010,DOI: 10.1002/9780470689646.ch1, In book: Text Mining: Applications and Theory (pp.1 - 20)

[3] Rada Mihalcea and Paul Tarau, "TextRank: Bringing Order into Texts",Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing July 2004, Barcelona, Spain, Association for Computational Linguistics Pages: 404–411

[4] Christos, T., Giorgos, O., Elena, M., Mavina, P., Christos, D. and Aristides, V., Developing a Greek biomedical corpus towards text mining.
[5] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[6] Peng, Y., Yan, S. and Lu, Z., 2019. Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets. arXiv preprint arXiv:1906.05474.
[7] Koutsikakis, J., Chalkidis, I., Malakasiotis, P. and Androutsopoulos, I., 2020, September. Greek-bert: The greeks visiting sesame street. In 11th Hellenic Conference on Artificial Intelligence (pp. 110-117) (https://github.com/nlpaueb/greek-bert)
[8] Sharma, Prafull, and Yingbo Li. "Self-Supervised Contextual Keyword and Keyphrase Retrieval with Self-Labelling." (2019).
[9] Bennani-Smires, Kamil, et al. "Simple unsupervised keyphrase extraction using sentence embeddings." arXiv preprint arXiv:1801.04470 (2018).
[10] https://github.com/MaartenGr/KeyBERT
[11] Papagiannopoulou, Eirini, and Grigorios Tsoumakas. "Local word vectors guiding keyphrase extraction." Information Processing & Management 54.6 (2018): 888-902.
[12] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Davison, J., 2019. HuggingFace's Transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.
[13] Papagiannopoulou, Eirini, and Grigorios Tsoumakas. "A review of keyphrase extraction." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10.2 (2020): e1339.
[14] Rousseau, François, and Michalis Vazirgiannis. "Main core retention on graph-of-words for single-document keyword extraction." European Conference on Information Retrieval. Springer, Cham, 2015.