Adapting Multilingual Sentence Transformers for Unsupervised Key-Phrase Extraction from Long Documents

Artur Guimarães
INESC-ID
IST, University of Lisbon
Lisbon
Portugal
artur.guimas@gmail.com

Abstract

Key-phrase extraction concerns retrieving a small set of phrases that encapsulate the core concepts of an input textual document. As in other NLP and IR tasks, current methods often rely on pre-trained neural language models. State-of-the-art supervised systems for key-phrase extraction require large amounts of labelled data and generalize poorly outside the training domain, while unsupervised approaches generally present a lower accuracy. This paper presents a multilingual unsupervised approach to key-phrase extraction, improving upon previous methods in several ways (e.g., using representations from pre-trained Transformer models, while supporting the processing of long documents). Experimental results on datasets covering multiple languages and domains attest to the quality of the results. The source code supporting the experiments is available from a public Github repository[1].

Keywords: Key-phrase extraction, Multilingual text processing, Transformers

1 Introduction

Key-Phrase Extraction (KPE) can be defined as the task of retrieving a small set of phrases from a given text document, to best describe its main concepts. Similarly to other Natural Language Processing (NLP) and Information Retrieval (IR) tasks, recent methods involve the use of text representations produced through neural models.

Supervised approaches for KPE require quantity and quality of in-domain annotated data, motivating work on transfer learning (Xiong et al., 2019; Joshi et al., 2022), weakly-supervised (Wang et al., 2020), or unsupervised (Shen et al., 2021; Bennani-Smires et al., 2018; Sun et al., 2020; Ding and Luo, 2021; Zhang et al., 2021) methods, that do not require the usage of expensive annotations nor extensive training procedures to obtain strong results. For instance, EmbedRank (Bennani-Smires et al., 2018) was created as a simple, yet very effective, unsupervised KPE method that can be broken down into three main steps: (1) candidate phrase extraction using patterns over parts-of-speech (POS) tags, selecting phrases with zero or more adjectives followed by one or more nouns; (2) using Sent2Vec[2] or Doc2Vec[3] sentence embeddings to represent both the candidate phrases and the analyzed document; (3) ranking candidate phrases using the cosine similarity measure between representations for each candidate phrase and the document. Experimental results showed that despite its simplicity, EmbedRank could outperform previous unsupervised methods for KPE, mostly based on graph-ranking approaches (Mihalcea and Tarau, 2004; Florescu and Caragea, 2017).

SIFRank (Sun et al., 2020) shares the same methodology of EmbedRank for extracting and ranking candidate phrases, but changes how the candidate phrases and the document are embedded. The ELMo (Peters et al., 2018) pre-trained language model, based on a deep recurrent neural network, is used to create the embeddings, and instead of directly comparing embeddings SIFRank uses a word weight balancing operation based on contextual information, which compares the domain corpus of the input document and a baseline common corpus, seeking to adapt the model to the specific domain at hand.

Both EmbedRank and SIFRank rely on the assumption that the similarity between a candidate phrase and a document is a good measure of how relevant that candidate phrase is. MDERank (Zhang et al., 2021) subverts this assumption, testing the hypothesis that a relevant candidate phrase maximizes the difference in a document when it is absent. To do so, MDERank ranks can-

[1] https://github.com/araag2/KP_Extraction

[2] https://github.com/epfml/sent2vec

[3] https://github.com/jhlau/doc2vec

didate phrases by replacing their occurrences in the document with a special [MASK] token, afterwards representing the document by embedding it using a Bidirectional Encoder Representation from Transformers (BERT) pre-trained language model (Devlin et al., 2018). The cosine distance towards the original document is measured, and candidate phrases having a higher distance are finally ranked as better.

Despite achieving strong empirical results, there are also some limitations in previous unsupervised methods. For instance most previous studies focused only on the English language, while it would be interesting to see if similar approaches can also generalize across languages. Problems also arise when considering large documents, as pre-trained language models often struggle to process long input sequences (e.g., the base BERT model has a 512 token limit).

This paper explores KPE in an unsupervised multilingual scenario, specifically by adapting and reconfiguring pre-existing methods (i.e., EmbedRank and MDERank) to work with representations produced with a Sentence-Transformers (Reimers and Gurevych, 2019) model, at the same time also supporting the processing of long text documents by converting the Sentence-Transformers model into a Long Document Transformer (Longformer) (Beltagy et al., 2020). The proposed KPE methods were evaluated on different domains (i.e., involving texts of different sizes, types, and languages), and the results show that they offer good generalization and improvements over previous approaches.

The rest of the paper is organized as follows: Section 2 details the proposed approaches, while Section 3 presents the experimental evaluation methodology and the obtained results. Section 4 summarizes our contributions and discusses future work.

2 The Proposed Approaches

Given a text document $\boldsymbol{d}$ belonging to a dataset $\mathfrak{D}$, we seek to extract a set $\mathfrak{C}$ of candidate phrases $\boldsymbol{c}$ that contains as many relevant key-phrases as possible, for describing the contents of $\boldsymbol{d}$. After extracting $\mathfrak{C}$, our second goal is to rank the top-k candidates within that set.

The first task is addressed using models from the spaCy[4] library, according to the language of the documents within dataset $\mathfrak{D}$, to tokenize and perform parts-of-speech tagging of each document $\boldsymbol{d}$. The regular expression <PROPN|NOUN|ADJ>*<PROPN|NOUN>+<ADJ>*, over sequences of universal parts-of-speech tags, is used as a heuristic method to extract candidate phrases, relying only a simple tagset that is common to different languages (Petrov et al., 2011). We also perform lemmatization to join candidates with slight differences into a single representation, through the simplemma[5] library which offers complete multilingual options. We keep a mapping between each possible form and the corresponding lemmatized candidates, so that matches in the text can be aggregated into the lemmatized versions.

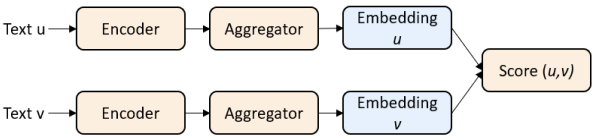

Figure 1: General bi-encoder architecture.

On what regards the ranking task, we first need to find suitable representations for the documents and the candidate phrases, adhering to some constraints: computational efficiency to perform multiple comparisons between documents and candidate phrases, support for multilingual text, and adequate handling of large documents.

2.1 Text Representations from a Longformer Built from a Sentence-Transformer

Transformer encoder models like BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) can produce effective text representations, but they are also computationally demanding. They can be used as cross-encoders to assess the similarity between a pair of input texts (i.e., processing the concatenation of both texts, and directly outputting a similarity score), but a more efficient approach is to instead consider a bi-encoder setting, in which the texts to be compared are modeled separately, and then a similarity score is computed over aggregates (e.g., token averages) from the resulting representations – see Figure 1. Moreover, these models also struggle when processing long documents, due to the quadratic complexity associated to the self-attention mechanism computed over all pairs of positions from input sequences. Recent approaches such as the Longformer (Beltagy et al., 2020) or BigBird (Zaheer et al., 2020) address this limitation, slightly changing the self-attention operations

[4] https://spacy.io/models/

[5] https://github.com/adbar/simplemma/

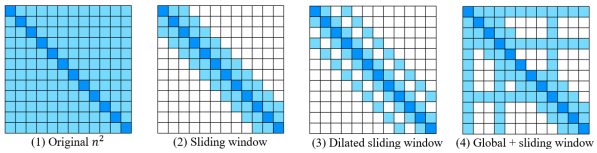

Figure 2: Self-attention in a standard Transformer, versus the attention patterns in a Longformer model.

in order to limit how the different positions interact – see the illustration on Figure 2.

In our Key-Phrase Extraction (KPE) methods, we use text representations obtained with a multilingual model based on RoBERTa, adapted from a model available from the Sentence-Transformers library[6] and pre-trained as a bi-encoder for assessing sentence similarity (Reimers and Gurevych, 2019). The RoBERTa-based model was adapted into a Longformer without any additional training, extending the input sequence limit to 4096 tokens (i.e., initializing the additional position embeddings by copying the embeddings of the first positions) and changing the implementation of the self-attention operations within the different layers, while keeping the model parameters.

In brief, Sentence-Transformers bi-encoders process strings independently through the same Transformer encoder, followed by mean pooling aggregation to create fixed-sized sentence embeddings. These models are trained either to directly predict sentence similarity scores as given in training data corresponding to annotated sentence pairs, or to predict similarity relations between sentences (e.g., given an anchor sentence a, a positive sentence p with high similarity towards a, and a negative sentence n, we can consider a loss function that tunes the network such that the distance between a and p is smaller than the distance between a and n). We specifically started from the Sentence-Transformers model named `paraphrase-multilingual-mpnet-base-v2`, i.e. a pre-existing model built from a multi-lingual RoBERTa and trained to mimic the results of another mono-ligual Sentence-Transformers bi-encoder, through a knowledge distillation objective (Reimers and Gurevych, 2020). This model was then adapted through the procedure described in the Longformer paper to build a Long Document Transformer starting from a RoBERTa checkpoint.

As can be seen in Figure 2, three attention patterns are combined within the Longformer architecture: sliding window, focusing on the local context and examining a fixed-size window w around each token; dilated sliding window, which adds a gap of size d between each token considered in the sliding window, with d varying across layers and attention heads; and global attention, in which some specific input locations (e.g., the initial `[CLS]` token) will attend to (and be attended by) all other tokens.

Our Longformer model employs a sliding window attention with window size of 512 tokens, thus involving approximately the same amount of computation as a standard RoBERTa, and also behaving like RoBERTa when the input has less than 512 tokens. One additional attention pattern was also considered, in which the specific positions corresponding to the occurrences of the key-phrase candidates were also considered for global attention, when representing candidates or documents.

In the remaining parts of this paper, we refer to the proposed text representation model as the Multilingual Sentence-Longformer (MSL). Using the word representations from MSL, we built different approaches to address the candidate ranking problem for KPE.

2.2 LMEmbedRank

Longformer Multilingual EmbedRank (LMEmbedRank) corresponds to an adaptation of EmbedRank (Bennani-Smires et al., 2018) that represents documents through MSL embeddings, and candidate phrases as the average of all MSL token embeddings that form the multiple occurrences of the candidate (first averaging the token representations from each occurrence, and then averaging across occurrences). As can be seen in Figure 3, where colored words represent the tokens that are considered, LMEmbendRank averages all token representations in order to create the embedding representation of a document, whilst to embed a candidate phrase (e.g., *core concepts*) it performs an average pooling operation over all tokens that form each occurrence over the document, and then averages over all occurrences. For candidates that only occur after the first 4096 tokens (i.e., the Longformer limit for input sequences), we still manage to generate a representation with a back-off procedure that processes the candidate string alone, without any contextual information.

Following the standard EmbedRank procedure, we use the cosine measure to rank candidate phrases according to the similarity of their repre-

[6] https://www.sbert.net/

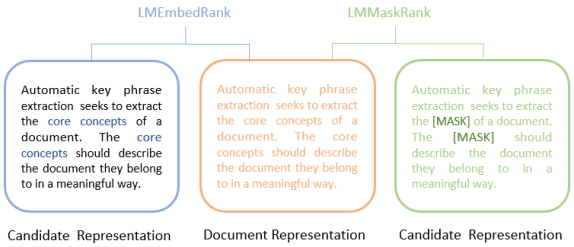

Figure 3: Overview on how LMEmbedRank and LMMaskRank represent the candidate *core concepts*.

Dataset	Lang.	Avg. #KPs	Cand. Recall	Absent KPs	Avg. #Words	#Docs.
DUC	EN	8	87.2%	6.8%	740	308
NUS	EN	11	88.2%	4.3%	5201	209
Inspec	EN	10	58.7%	35.6%	128	2000
SemEval	EN	16	95.0%	3.2%	8332	243
PubMed	EN	15	80.2%	15.8%	3992	1320
PT-KP	PT	24	53.6%	5.2%	304	110
CACIC	ES	5	72.3%	7.3%	3985	888
WICC	ES	5	74.3%	5.9%	1955	1640
FR-WIKI	FR	12	79.1%	4.4%	293	100
TeKET	DE	5	93.5%	0.0%	11524	10

Table 1: Statistics for the considered datasets.

sentations towards the document representation, in descending order of similarity.

2.3 LMMaskRank

Longformer Multilingual MaskRank (LMMaskRank) corresponds to an adaption of MDERank (Zhang et al., 2021) that also represents documents and candidate phrases through MSL embeddings. As can be seen again in Figure 3, where colored words represent the considered tokens, in order to create the embedding representation of a document LMMaskRank uses the same mechanism as LMEmbedRank, whilst to embed a candidate phrase (e.g., *core concepts*) the method starts by replacing all of the candidate occurrences by the `[MASK]` token, and then embeds the entirety of the document. As described in the original paper, candidate phrases are then ranked using the cosine distance measure, in descending order of distance (i.e., candidate representations that are further away from that of the original document are preferred).

It is interesting to note that the most computationally expensive operation, in both LMEmbedRank and LMMaskRank, corresponds to obtaining the MSL embeddings (i.e., one forward pass over the Longformer model). LMMaskRank is thus much more demanding, given that LMEmbedRank only needs to compute MSL embeddings once, while LMMaskRank needs a separate computation for each candidate (i.e., replacing the candidate occurrences with `[MASK]` tokens, before computing the corresponding representations).

2.4 Combining Both Ranking Approaches

Longformer Multilingual Rank (LMRank) corresponds to a hybrid approach that uses weighted averages of scores obtained by both previous methods, based on the hypothesis that each method would be better suited to handle different types of documents, and thus together they could probably perform better. This general approach can be implemented through different combination schemes, and we tested the arithmetic and harmonic averages of LMEmbedRank and LMMaskRank scores.

3 Experimental Evaluation

This section starts by introducing the datasets that were used in the experiments, together with the considered evaluation metrics. It then follows with an overview on the experimental results across all datasets. We also provide a comparison with previous methods, as well as against ablated versions of the proposed approaches.

3.1 Metrics and Datasets

To evaluate the performance of our models in different languages and domains, we relied on a wide variety of datasets used in previous studies: five English datasets, namely NUS (Nguyen and Kan, 2007), DUC-2001 (Wan and Xiao, 2008), Inspec (Hulth, 2003), SemEval (Kim et al., 2010) and PubMed (Aronson et al., 2000); an European Portuguese dataset named 110-PT-BN-KP (PT-KP) (Marujo et al., 2013); two Spanish datasets, namely CACIC (Aquino and Lanzarini, 2015a) and WICC (Aquino and Lanzarini, 2015b); a French dataset named WikiNews (FR-WIKI) (Bougouin et al., 2013); and a German dataset (TeKET) (Rabby et al., 2020). Additional information is presented in Table 1.

The candidate-phrase extraction component was initially evaluated in terms of recall, by comparing our extraction with the ground truth key-phrases of each document. Notice that without a high recall it will be impossible to accurately rank the key-phrase candidates, as the correct key-phrases will not be available to be ranked.

It is interesting to note that there exists an upper bound on the possible recall value, as the candidate

Model	DUC				NUS				Inspec				SemEval				PubMed			
	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$
LMEmbedRank	**45.01**	**30.28**	**33.72**	**34.10**	28.55	20.61	22.87	23.12	**45.95**	**31.25**	**36.97**	**38.05**	22.16	14.49	18.74	**20.93**	13.14	7.94	10.22	10.28
LMMaskRank	37.00	24.41	28.31	28.15	30.53	18.58	20.01	17.38	41.19	27.75	33.31	34.78	21.10	15.38	17.48	18.10	27.72	17.36	18.93	18.14
LMRank$_{avg_a}$	39.57	25.58	29.96	31.66	28.16	22.76	24.49	24.86	43.54	29.66	35.34	36.45	19.78	14.59	16.22	16.27	23.65	15.1	16.46	**20.35**
LMRank$_{avg_h}$	40.98	29.68	31.29	30.05	**38.11**	**28.17**	**32.69**	**31.79**	43.43	30.10	34.67	35.14	**22.97**	**16.69**	**19.17**	18.25	**29.48**	**17.94**	**20.15**	18.74
Model	PT-KP				CACIC				WICC				FR-WIKI				TeKET			
	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$
LMEmbedRank	36.78	23.91	32.91	37.29	**40.15**	**26.49**	**16.93**	13.20	24.88	15.31	14.85	13.97	43.64	24.15	32.92	26.50	6.48	0.00	6.95	6.15
LMMaskRank	39.90	25.96	34.38	37.63	22.52	16.75	14.72	14.07	27.50	17.76	16.48	14.31	49.62	37.06	37.65	34.83	**17.65**	10.44	**10.95**	**10.75**
LMRank$_{avg_a}$	**41.26**	28.04	**36.62**	**39.59**	17.68	11.09	13.13	**15.37**	17.99	13.20	12.37	11.20	50.88	37.53	**39.97**	**36.19**	15.87	**10.66**	10.28	10.15
LMRank$_{avg_h}$	41.09	**28.06**	35.79	38.91	25.64	19.11	15.98	14.77	**31.43**	**20.57**	**17.29**	**15.01**	**49.95**	**37.93**	38.43	35.86	**17.65**	10.44	**10.95**	**10.75**

Table 2: Key-phrase extraction results on each dataset and for each of the proposed methods.

extraction method is unable to find correct key-phrases that do not appear within the input text documents (although the lematization operation does help in this regard). The candidate extraction recall, for each dataset, is also shown in Table 1, together with the percentage of ground-truth key-phrases that do not occur in the text.

Overall, we can see that the proposed candidate extraction method is able to correctly find a large percentage of the ground-truth key-phrases in the majority of the datasets, with exceptions for Inspec (where we also have a very large number of absent key-phrases from the contents of the documents) and PT-KP. In this latter case, our regular expression pattern was unable to find many of the ground-truth key-phrases, which do not always correspond to a noun phrase.

The candidate rankings are handled as an ordered list, and a specific cut-off point k can then be defined, comparing the top k ranked candidates with the ground-truth key-phrases. The performance on the ranking task is measured with the F_1-score ($F_{1,k}$) metric, at the cut-off points $k = \{5, 10, 15\}$. Additionally, we also use the Normalized Discounted Cumulative Gain (nDCG) metric over the complete ranked list.

Following most previous studies in the area, both the extracted and the ground-truth key-phrases are processed through a stemming algorithm, prior to performing comparisons for ranking evaluation.

3.2 Experimental Results

Table 2 presents experimental results over the multiple datasets, comparing the alternatives discussed in the previous section. The lines named MRank$_{avg_a}$ and LMRank$_{avg_h}$ correspond to using an arithmetic or harmonic average of LMEmbedRank and LMMaskRank scores, as described previously in Subsection 2.4.

Although the different evaluation metrics mostly agree on how the methods should be ranked according to result quality, different methods can perform slightly better on some of the datasets:

- LMEmbedRank, which is also the computationally more effective method, performs clearly better than LMMaskRank on the DUC, Inspec, and CACIC datasets.
- In turn, LMMaskRank clearly performs better than LMEmbedRank on Pubmed, PT-KP, WICC, FR-WIKI, and TeKET.
- Datasets like NUS, SemEval, PubMed, and particularly TeKET, feature very long documents, beyond the 4096 token limit in Longformer. In these cases, LMMaskRank tends to perform better, although the relation between result quality and the characteristics of the documents (e.g., size, language, or candidate recall) is not entirely clear. Note that LMMaskRank is biased towards preferring candidates occurring in the first 4096 tokens, since occurrences beyond this limit will not impact the representations (i.e., the representations for these candidates are exactly equal to those from the documents, and hence they will be ranked below other candidates).
- The combination of both approaches, particularly when considering the harmonic mean, is beneficial in most cases. In the NUS, SemEval, PubMed, WICC, and FR-WIKI datasets, the best results are achieved with

Model	DUC				NUS				Inspec				SemEval				PT-KP			
	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$	nDCG	$F1_5$	$F1_{10}$	$F1_{15}$
LMEmbedRank	45.01	30.28	33.72	34.10	28.55	20.61	22.87	23.12	45.95	31.25	36.97	38.05	22.16	14.49	18.74	20.93	36.78	23.91	32.91	37.29
- Longformer	40.92	27.56	31.26	31.23	18.12	11.14	13.52	12.49	44.51	29.70	35.40	36.46	12.35	1.82	8.69	12.05	34.82	21.41	30.95	35.37
- Sentence-BERT	26.86	15.75	22.10	22.50	17.54	12.23	12.57	11.90	23.68	17.54	19.80	18.27	10.27	2.03	7.07	11.32	–	–	–	–
- Lematization	42.10	29.09	31.85	31.93	25.79	19.31	20.72	21.14	43.42	29.88	35.01	36.31	19.21	12.78	16.48	18.88	33.84	22.43	30.89	35.28
- Global Attention	44.15	29.59	33.16	33.37	27.56	19.93	22.22	22.41	45.23	30.88	36.57	37.6	21.18	13.77	18.05	20.34	35.87	23.03	32.32	36.64
LMMaskRank	37.00	24.41	28.31	28.15	30.53	18.58	20.01	17.38	41.19	27.75	33.31	34.78	21.10	15.38	17.48	18.10	39.90	25.96	34.38	37.63
- Longformer	33.89	22.46	25.85	24.82	20.79	9.75	10.72	7.78	39.83	26.34	31.75	33.11	12.21	2.99	7.91	8.69	36.42	22.75	30.95	35.03
- Sentence-BERT	32.92	22.39	27.71	26.50	23.74	12.36	14.09	11.83	27.09	20.45	22.82	23.03	15.64	10.93	12.99	14.21	–	–	–	–
- Lemmatization	34.23	23.1	26.19	26.25	27.55	17.49	18.09	15.28	38.29	26.32	31.47	32.87	18.13	13.90	15.22	16.5	33.65	20.56	28.82	30.97
- Global Attention	36.04	23.5	27.37	27.01	29.6	18.00	19.25	16.61	40.58	27.46	32.94	34.27	20.26	14.36	16.96	17.44	39.05	23.71	31.55	35.07
LMRank$_{avg_h}$	40.98	29.68	31.29	30.05	38.11	28.17	32.69	31.79	43.43	30.10	34.67	35.14	22.97	16.69	19.17	18.25	41.09	28.06	35.79	38.91
- Longformer	37.62	27.22	29.14	27.14	29.37	21.40	27.32	25.38	41.25	28.86	32.87	33.21	15.27	8.92	11.28	11.91	37.87	25.15	33.33	36.13
- Sentence-BERT	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
- Lemmatization	38.32	28.14	29.49	28.13	36.09	27.09	30.88	29.95	41.31	28.91	32.78	33.31	20.67	15.04	17.44	16.51	33.39	20.71	29.63	31.58
- Global Attention	39.68	28.63	30.14	28.78	37.12	26.92	31.65	30.47	42.60	28.72	33.90	34.30	22.10	15.66	18.48	17.07	40.13	26.42	34.79	37.64
+ Weighting	40.57	29.06	30.82	29.33	38.81	28.65	33.23	32.38	42.64	29.62	34.14	34.52	22.36	16.32	18.85	17.65	41.83	28.45	36.25	39.53

Table 3: Results with ablated versions of the proposed key-phrase extraction methods.

a combined method. On the other datasets, the combination performs similarly to the best method, improving over the LMEmbedRank or LMMaskRank strategies.

In turn, Table 3 presents results for ablated versions of the LMEmbedRank, LMMaskRank, and LMRank$_{avg_h}$ methods, specifically assessing the impact of different ideas introduced in our proposal. The following alternatives were tested on 5 of the datasets also seen in Table 2:

- Using the regular Sentence-Transformers model based on a multi-lingual RoBERTa, instead of converting the model into a Longformer. In the case of LMEmbedRank, the candidates that occur after the maximum token limit of the model were also represented with the back-off procedure that processes the candidate string alone;
- Using a standard English Longformer[7], instead of the Sentence-Transformers model pre-trained only for Masked Language Modeling (MLM). This way, we can assess the impact of model pre-training with sentence similarity tasks, noting also that previous studies such as MDERank (Zhang et al., 2021) have only explored the use of regular Transformer encoders pre-trained for MLM.
- Removing the lemmatization procedure that aggregates similar candidates appearing with a slightly different surface form;
- Removing the Longformer attention pattern that considers a global attention for the tokens that correspond to candidate occurrences, instead leaving only the `[CLS]` token with the global attention over all other tokens.

The results show that all the four previous aspects, and particularly the pre-training over sentence similarity tasks (i.e., using an adapted Sentence-Transformers model) and the conversion of the Sentence-Transformers model into a Longformer, contribute to improved results. Higher differences in the result quality are also seen in the datasets involving longer documents.

Besides the aforementioned ablations, we also considered extensions over the methods in Table 2, leveraging ideas advanced in previous studies. These included (a) post-processing the document/candidate embeddings prior to computing similarities (Sajjad et al., 2021; Huang et al., 2021; Jégou and Chum, 2012; Su et al., 2021), or (b) weighting the individual tokens when computing the representations within LMEmbedRank, e.g. proportionally to attention scores (Ding and Luo, 2021). Still, results were consistently worse, and we decided not to report these scores.

One particular extension that we tested involves

[7] https://huggingface.co/allenai/longformer-large-4096

Model	DUC			NUS			Inspec			SemEval			PT-KP			TeKET		
	$F1_5$	$F1_{10}$	$F1_{15}$	$F1_5$	$F1_{10}$	$F1_{15}$	$F1_5$	$F1_{10}$	$F1_{15}$	$F1_5$	$F1_{10}$	$F1_{15}$	$F1_5$	$F1_{10}$	$F1_{15}$	$F1_5$	$F1_{10}$	$F1_{15}$
TF-IDF	9.21	10.63	11.60	11.60	14.20	12.50	24.20	28.00	24.80	16.10	16.70	15.30	-	17.9	-	7.50	8.60	9.60
TopicRank	19.97	21.73	20.97	4.54	7.93	9.37	12.20	17.24	19.33	9.93	12.52	12.26	-	14.80	-	6.30	7.60	8.10
EmbedRank	21.75	25.09	24.68	2.13	2.94	3.56	14.51	21.02	23.79	9.63	13.90	14.79	-	-	-	-	-	-
SIFRank	24.30	27.60	27.96	3.01	5.34	5.86	29.38	**39.12**	**39.82**	11.16	16.03	18.42	-	-	-	-	-	-
MDERank	13.05	17.31	19.13	15.24	18.33	17.95	26.17	33.81	36.17	10.16	15.32	17.76	-	-	-	-	-	-
AttentionRank	-	-	-	-	-	-	24.45	32.15	34.49	11.39	15.12	16.66	-	-	-	-	-	-
YAKE!	11.99	14.18	14.18	7.85	11.05	13.09	8.02	11.47	13.65	6.82	11.01	12.55	-	10.70	-	8.83	**12.30**	**13.80**
Multipartite	21.70	24.10	23.62	6.17	8.57	10.82	13.41	18.18	20.52	10.13	12.91	13.24	12.05	15.60	-	7.10	9.10	9.70
CDKGen	-	-	-	41.20	38.10	-	33.10	34.70	-	34.20	35.50	-	-	-	-	-	-	-
SEG-Net	-	-	-	39.60	-	-	21.60	-	-	28.30	-	-	-	-	-	-	-	-
SKE-Base-Rank	-	-	-	38.90	36.50	-	28.90	32.10	-	35.40	33.70	-	-	-	-	-	-	-
LMEmbedRank	**30.28**	**33.72**	**34.10**	20.61	22.87	23.12	**31.25**	36.97	38.05	14.49	18.74	**20.93**	23.91	32.91	37.29	0.0	6.95	6.15
LMRank$_{avg_h}$	29.68	31.29	30.05	**28.17**	**32.69**	**31.79**	30.10	34.67	35.14	**16.69**	**19.17**	18.25	**28.06**	**35.79**	**38.91**	**10.44**	10.95	10.75

Table 4: Comparison between our best key-phrase extraction methods and previously published results.

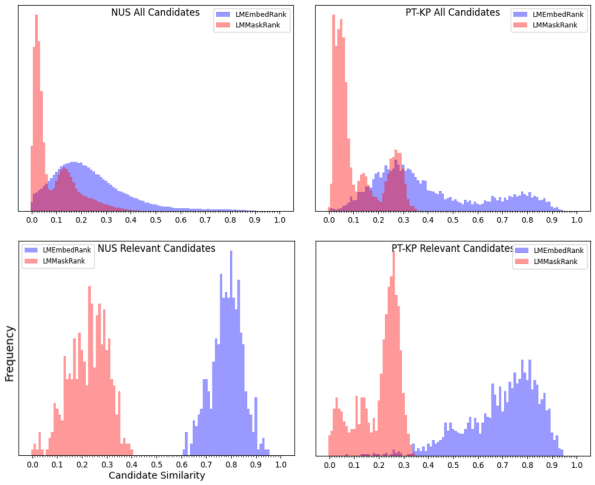

Figure 4: Distribution of similarity scores between candidates and documents, considering all candidates and documents for two different datasets (on top), or only the subsets of relevant candidates (at the bottom).

weighting the scores of the LMEmbedRank and LMMaskRank methods prior to their combination, in an attempt to further improve results. In a first step towards doing this, we started by analyzing the distribution of the similarity scores between candidate and document representations, for the two different methods (i.e., LMEmbedRank and LMMaskRank) and over the different datsets. Figure 4 illustrates the results of this analysis, specifically for the NUS and PT-KP datasets (similar patterns could be observed also for the other datasets).

The results showed that LMEmbedRank produces similarity scores that are more evenly spread, whereas LMMaskRank mostly produces results in the interval $[0, 0.5]$. Both methods also produce two peaks in terms of the distribution for the similarity values, corresponding to a good distinction between the relevant and the irrelevant candidates (i.e., the top charts in Figure 4 correspond to the entire sets of candidates, whereas the bottom charts show only the similarity scores for the subset of relevant candidates).

With basis on the analysis, we then tested a combination method in which a constant of 0.5 is added to the scores from LMMaskRank procedure, prior to the combination with LMEmbedRank. The results are shown in the bottom row from Table 2, although no noticable improvements were seen.

Finally, Table 4 compares the best proposed methods, specifically LMEmbedRank (i.e., the simplest and fastest method) and LMRank$_{avg_h}$, against the results reported in publications presenting and using previous methods (including results for the original EmbedRank (Bennani-Smires et al., 2018) and MDERank (Zhang et al., 2021) methods). We present results for the datasets over which more previous methods have been tested (i.e., mostly by re-using results presented on previous comparisons (Zhang et al., 2021)), also including results for some recent supervised approaches (i.e., the second set of rows in Table 4).

The results in Table 4 show that the proposed approaches are very competitive within the realm of unsupervised KPE, outperforming most unsupervised methods in the majority of the datasets and often by very large margin, while simultaneously being simple, multilingual, and thus easy to generalize to different types of applications.

Notable exceptions correspond to the Inspec and the TeKET datasets. In the specific case of Inspec, SIFRank (Sun et al., 2020) outperforms the proposed methods in $F1_{10}$ and $F1_{15}$, likely due to the small size of the documents (128 words on average) which offset the positive Longformer effect. On TeKET, YAKE! (Campos et al., 2020) outper-

forms the proposed approaches also in in $F1_{10}$ and $F1_{15}$, but in this case it is difficult to draw many conclusions because the dataset only features 10 very long documents (11524 words on average), and hence the results can be very noisy.

It is also important to notice that the differences towards recent supervised methods are still very significant. Previous methods such as CDKGen (Diao et al., 2020), SEG-NET (Ahmad et al., 2021), or SKE-Base-Rank (Mu et al., 2020) are, usually, still significantly better than the best unsupervised approaches, although this also varies depending on characteristics of the datasets (e.g., on Inspec, the best results in terms of $F1_{10}$ and $F1_{15}$ are obtained with unsupervised methods).

4 Conclusions and Future Work

We proposed new unsupervised methods for keyphrase extraction, extending the previous EmbedRank (Bennani-Smires et al., 2018) and MDERank (Zhang et al., 2021) approaches in different directions. We tested the proposed approaches over multiple datasets, with results showing a very competitive performance against state-of-the-art unsupervised methods, while also generalizing across different languages and domains.

For future work, we can consider other methods for handling long inputs besides the Longformer (e.g., memory efficient attention implementations (Rabe and Staats, 2021), or other sparse attention patterns such as those in the Hypercube Transformer (Wang et al., 2022)). We would also like to experiment on scenarios that involve multi-document key-phrase extraction (Shapira et al., 2021), this way further stressing the length of the textual inputs that need to be analyzed.

Acknowledgements

This research was supported by the European Union's H2020 research and innovation programme, under grant agreement No. 874850 (MOOD), as well as by Fundação para a Ciência e Tecnologia (FCT), namely through the INESC-ID multi-annual funding from the PIDDAC programme with reference UIDB/50021/2020, and also through the project grants with references PTDC/CCI-CIF/32607/2017 (MIMU), DSAIPA/DS/0102/2019 (DEBAQI), and POCI/01/0145/FEDER/031460 (DARGMINTS).

References

Wasi Ahmad, Xiao Bai, Soomin Lee, and Kai-Wei Chang. 2021. Select, extract and generate: Neural keyphrase generation with layer-wise coverage attention. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Germán Osvaldo Aquino and Laura Cristina Lanzarini. 2015a. Keyword identification in Spanish documents using neural networks. *Journal of Computer Science & Technology*, 15.

Germán Osvaldo Aquino and Laura Cristina Lanzarini. 2015b. Keyword identification in Spanish documents using neural networks. *Journal of Computer Science & Technology*, 15.

Alan R Aronson, Olivier Bodenreider, H Florence Chang, Susanne M Humphrey, James G Mork, Stuart J Nelson, Thomas C Rindflesch, and W John Wilbur. 2000. The NLM Indexing Initiative. In *Proceedings of the American Medical Informatics Association Symposium*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint 2004.05150*.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the Conference on Computational Natural Language Learning*.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the International Joint Conference on Natural Language Processing*.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. YAKE! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint 1810.04805*.

Shizhe Diao, Yan Song, and Tong Zhang. 2020. Keyphrase generation with cross-document attention. *arXiv preprint arXiv:2004.09800*.

Haoran Ding and Xiao Luo. 2021. AttentionRank: Unsupervised keyphrase extraction using self and cross attentions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Corina Florescu and Cornelia Caragea. 2017. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An easy unsupervised sentence embedding approach. *arXiv preprint 1801.04470*.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Hervé Jégou and Ondřej Chum. 2012. Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *Proceedings of the European Conference on Computer Vision*.

Rishabh Joshi, Vidhisha Balachandran, Emily Saldanha, Maria Glenski, Svitlana Volkova, and Yulia Tsvetkov. 2022. Unsupervised keyphrase extraction via interpretable neural networks. *arXiv preprint 2203.07640*.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the International Workshop on Semantic Evaluation*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint 1907.11692*.

Luis Marujo, Márcio Viveiros, and João Paulo da Silva Neto. 2013. Keyphrase cloud generation of broadcast news. *arXiv preprint 1306.4606*.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Funan Mu, Zhenting Yu, LiFeng Wang, Yequan Wang, Qingyu Yin, Yibo Sun, Liqun Liu, Teng Ma, Jing Tang, and Xing Zhou. 2020. Keyphrase extraction with span-based feature representations. *arXiv preprint arXiv:2002.05407*.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *Proceedings of the International Conference on Asian Digital Libraries*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint 1802.05365*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint 1104.2086*.

Gollam Rabby, Saiful Azad, Mufti Mahmud, Kamal Z Zamli, and Mohammed Mostafizur Rahman. 2020. TeKET: a tree-based unsupervised keyphrase extraction technique. *Cognitive Computation*, 12(4).

Markus N. Rabe and Charles Staats. 2021. Self-attention does not need $o(n^2)$ memory. *arXiv preprint 2112.05682*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv preprint 1908.10084*.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Hassan Sajjad, Firoj Alam, Fahim Dalvi, and Nadir Durrani. 2021. Effect of post-processing on contextualized word representations. *arXiv preprint 2104.07456*.

Ori Shapira, Ramakanth Pasunuru, Ido Dagan, and Yael Amsterdamer. 2021. Multi-document keyphrase extraction: A literature review and the first dataset. *arXiv preprint arXiv:2110.01073*.

Xianjie Shen, Yinghan Wang, Rui Meng, and Jingbo Shang. 2021. Unsupervised deep keyphrase generation. *arXiv preprint arXiv:2104.08729*.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Y. Sun, H. Qiu, Y. Zheng, Z. Wang, and C. Zhang. 2020. SIFRank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model. *IEEE Access*, 8:10896–10906.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence*.

Xuan Wang, Xiangchen Song, Bangzheng Li, Yingjun Guan, and Jiawei Han. 2020. Comprehensive named entity recognition on CORD-19 with distant or weak supervision. *arXiv preprint 2003.12218*.

Yuxing Wang, Chu-Tak Lee, Zhangyue Yin Qipeng Guo, Yunhua Zhou, Xuanjing Huang, and Xipeng Qiu. 2022. What dense graph do you need for self-attention? *arXiv preprint arXiv:2205.14014*.

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Proceedings of the Annual Meeting on Neural Information Processing Systems*.

Linhan Zhang, Qian Chen, Wen Wang, Chong Deng, Shiliang Zhang, Bing Li, Wei Wang, and Xin Cao. 2021. MDERank: A masked document embedding rank approach for unsupervised keyphrase extraction. *arXiv preprint arXiv:2110.06651*.