Extending Neural Keyword Extraction with TF-IDF tagset matching

Boshko Koloski
Jožef Stefan Institute
Jožef Stefan IPS
Jamova 39, Ljubljana
`boshko.koloski@ijs.si`

Senja Pollak
Jožef Stefan Institute
Jamova 39, Ljubljana
`senja.pollak@ijs.si`

Blaž Škrlj
Jožef Stefan Institute
Jožef Stefan IPS
Jamova 39, Ljubljana
`blaz.skrlj@ijs.si`

Matej Martinc
Jožef Stefan Institute
Jamova 39, Ljubljana
`matej.martinc@ijs.si`

Abstract

Keyword extraction is the task of identifying words (or multi-word expressions) that best describe a given document and serve in news portals to link articles of similar topics. In this work, we develop and evaluate our methods on four novel data sets covering less-represented, morphologically-rich languages in European news media industry (Croatian, Estonian, Latvian, and Russian). First, we perform evaluation of two supervised neural transformer-based methods, Transformer-based Neural Tagger for Keyword Identification (TNT-KID) and Bidirectional Encoder Representations from Transformers (BERT) with an additional Bidirectional Long Short-Term Memory Conditional Random Fields (BiLSTM CRF) classification head, and compare them to a baseline Term Frequency - Inverse Document Frequency (TF-IDF) based unsupervised approach. Next, we show that by combining the keywords retrieved by both neural transformer-based methods and extending the final set of keywords with an unsupervised TF-IDF based technique, we can drastically improve the recall of the system, making it appropriate for usage as a recommendation system in the media house environment.

1 Introduction

Keywords are words (or multi-word expressions) that best describe the subject of a document, effectively summarise it and can also be used in several document categorization tasks. In online news portals, keywords help with efficient retrieval of articles when needed. Similar keywords characterise articles of similar topics, which can help editors to link related articles, journalists to find similar articles and readers to retrieve articles of interest

when browsing the portals. For journalists manually assigning tags (keywords) to articles represents a demanding task, and high-quality automated keyword extraction shows to be one of components in news digitalization process that many media houses seek for.

The task of keyword extraction can generally be tackled in an unsupervised way, i.e., by relying on frequency based statistical measures (Campos et al., 2020) or graph statistics (Škrlj et al., 2019), or with a supervised keyword extraction tool, which requires a training set of sufficient size and from appropriate domain. While supervised methods tend to work better due to their ability to adapt to a specifics of the syntax, semantics, content, genre and keyword assignment regime of a specific text (Martinc et al., 2020a), their training for some less resource languages is problematic due to scarcity of large manually annotated resources. For this reason, studies about supervised keyword extraction conducted on less resourced languages are still very rare. To overcome this research gap, in this paper we focus on supervised keyword extraction on three less resourced languages, Croatian, Latvian, and Estonian, and one fairly well resourced language (Russian) and conduct experiments on data sets of media partners in the EMBEDDIA project[1]. The code for the experiments is made available on GitHub under the MIT license[2].

In media house environments, automatic keyword extraction systems are expected to return a diverse list of keyword candidates (of constant length), which is then inspected by a journalist who

---

[1] http://embeddia.eu/
[2] https://github.com/bkolosk1/Extending-Neural-Keyword-Extraction-with-TF-IDF-tagset-matching/

manually selects appropriate candidates. While the state-of-the-art supervised approaches in most cases offer good enough precision for this type of usage as a recommendation system, the recall of these systems is nevertheless problematic. Supervised systems learn how many keywords should be returned for each news article on the gold standard train set, which generally contains only a small amount of manually approved candidates for each news article. For example, among the datasets used in our experiments (see Section 3), the Russian train set contains the most (on average 4.44) present keywords (i.e., keywords which appear in the text of the article and can be used for training of the supervised models) per article, while the Croatian test set contains only 1.19 keywords per article. This means that for Croatian, the model will learn to return around 1.19 keywords for each article, which is not enough.

To solve this problem we show that we can improve the recall of the existing supervised keyword extraction system by:

- Proposing an additional TF-IDF tagset matching technique, which finds additional keyword candidates by ranking the words in the news article that have appeared in the predefined keyword set containing words from the gold standard train set. The new hybrid system first checks how many keywords were returned by the supervised approach and if the number is smaller than needed, the list is expanded by the best ranked keywords returned by the TF-IDF based extraction system.

- Combining the outputs of several state-of-the-art supervised keyword extraction approaches.

The rest of this work is structured as follows: Section 2 presents the related work, while Section 3 describes the datasets on which we evaluate our method. Section 4 describes our proposed method with all corresponding steps. The experiment settings are described in Section 5 and the evaluation of the proposed methods is shown in Section 6. The conclusions and the proposed further work are presented in Section 7.

2 Related Work

Many different approaches have been developed to tackle the problem of extracting keywords. The early approaches, such as KP-MINER (El-Beltagy and Rafea, 2009) and RAKE (Rose et al., 2010) rely on unsupervised techniques which employ frequency based metrics for extraction of keywords from text. Formally, aforementioned approaches search for the words w from vocabulary $\mathcal{V}$ that maximize a given metric h for a given text t:

$$\text{kw} = \underset{w \in \mathcal{V}}{\text{argmax}}\, h(w, t).$$

In these approaches, frequency is of high relevance and it is assumed that the more frequent a given word, the more important the meaning this word carries for a given document. Most popular such metrics are the naïve frequency (word count) and the term frequency-inverse document frequency (TF-IDF) (Salton and McGill, 1986).

Most recent state-of-the-art statistical approaches, such as YAKE (Campos et al., 2020), also employ frequency based features, but combine them with other features such as casing, position, relatedness to context and dispersion of a specific term in order to derive a final score for each keyword candidate.

Another line of research models this problem by exploiting concepts from graph theory. Approaches, such as TextRank (Mihalcea and Tarau, 2004), Single Rank (Wan and Xiao, 2008), TopicRank (Bougouin et al., 2013) and Topical PageRank (Sterckx et al., 2015) build a graph G, i.e., a mathematical construct described by a set of vertexes V and a set of edges E connecting two vertices. In one of the most recent approaches called RaKUn (Škrlj et al., 2019), a directed graph is constructed from text, where vertexes V and two words w_i, w_{i+1} are linked if they appear following one another. Keywords are ranked by a shortest path-based metric from graph theory - the load centrality.

The task of keyword extraction can also be tackled in a supervised way. One of the first supervised approaches was an algorithm named KEA (Witten et al., 2005), which uses only TF-IDF and the term's position in the text as features for term identification. More recent neural approaches to keyword detection consider the problem as a sequence-to-sequence generation task (Meng et al., 2017) and employ a generative model for keyword prediction with a recurrent encoder-decoder framework and an attention mechanism capable of detecting keywords in the input text sequence whilst also potentially finding keywords that do not appear in the text.

Finally, the newest branch of models consider keyword extraction as a sequence labelling task and tackle keyword detection with transformers. Sahrawat et al. (2020) fed contextual embeddings generated by several transformer models (BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), GPT-2 (Radford et al., 2019), etc.) into two types of neural architectures, a bidirectional Long short-term memory network (BiLSTM) and a BiLSTM network with an additional Conditional random fields layer (BiLSTM-CRF). Sun et al. (2020) on the other hand proposed BERT-JointKPE that employs a chunking network to identify phrases and a ranking network to learn their salience in the document. By training BERT jointly on the chunking and ranking tasks the model manages to establish balance between the estimation of keyphrase quality and salience.

Another state-of-the-art transformer based approach is TNT-KID (Transformer-based Neural Tagger for Keyword Identification) (Martinc et al., 2020a), which does not rely on pretrained language models such as BERT, but rather allows the user to train their own language model on the appropriate domain. The study shows that smaller unlabelled domain specific corpora can be successfully used for unsupervised pretraining, which makes the proposed approach easily transferable to low-resource languages. It also proposes several modifications to the transformer architecture in order to adapt it for a keyword extraction task and improve performance of the model.

3   Data Description

We conducted experiments on datasets containing news in four languages; Latvian, Estonian, Russian, and Croatian. Latvian, Estonian and Russian datasets contain news from the Ekspress Group, specifically from Estonian Ekspress Meedia (news in Estonian and Russian) and from Latvian Delfi (news in Latvian and Russian). The dataset statistics are presented in Table 2, and the datasets (Pollak et al., 2021) and their train/test splits[3] are publicly available. The media-houses provided news articles from 2015 up to the 2019. We divided them into training and test sets. For the Latvian, Estonian, and Russian training sets, we used the articles from 2018, while for the test set the articles from 2019 were used. For Croatian, the articles

from 2019 are arranged by date and split into training and test (i.e., about 10% of the 2019 articles with the most recent date) set. In our study, we also use tagsets of keywords. Tagset corresponds either to a collection of keywords maintained by editors of a media house (see e.g. Estonian tagset), or to a tagset constructed from assigned keywords from articles available in the training set. The type of tagset and the number of unique tags for each language are listed in Table 1.

Dataset	Unique tags	Type of tags
Croatian	21,165	Constructed
Estonian	52,068	Provided
Russian	5,899	Provided
Latvian	4,015	Constructed

Table 1: Distribution of tags provided per language. The media houses provided tagsets for Estonian and Russian, while the tags for Latvian and Croatian were extracted from the train set.

4   Methodology

The recent supervised neural methods are very precise, but, as was already mentioned in Section 1, in same cases they do not return a sufficient number of keywords. This is due to the fact that the methods are trained on the training data with a low number of gold standard keywords (as it can be seen from Table 2). To meet the media partners' needs, we designed a method that complements state-of-the-art neural methods (the TNT-KID method (Martinc et al., 2020b) and the transformer-based method proposed by Sahrawat et al. (2020), which are both described in Section 2) by a tagset matching approach, returning constant number of keywords (k=10).

4.1   Transformer-based Keyword Extraction

Both supervised neural approaches employed in this study are based on the Transformer architecture (Vaswani et al., 2017), which was somewhat adapted for the specific task at hand. Both models are fed lowercased text consisting of the title and the body of the article. Tokenization is conducted by either using the default BERT tokenizer (when BERT is used) or by employing Sentencepiece tokenizer (Kudo and Richardson, 2018) (when TNT-KID is used). While the multilingual BERT model is already pretrained on a large corpus consisting of Wikipedias of about 100 languages (Devlin et al.,

Dataset	Total docs	Total kw.	Avg. Train					Avg. Test				
			Total docs	Doc len	Kw.	% present kw.	present kw.	Total docs	Doc len	Kw.	% present kw.	Present kw.
Croatian	35,805	126,684	32,223	438.50	3.54	0.32	1.19	3582	464.39	3.53	0.34	1.26
Estonian	18,497	59,242	10,750	395.24	3.81	0.65	2.77	7,747	411.59	4.09	0.69	3.12
Russian	25,306	5,953	13,831	392.82	5.66	0.76	4.44	11,475	335.93	5.43	0.79	4.33
Latvian	24,774	4,036	13,133	378.03	3.23	0.53	1.69	11,641	460.15	3.19	0.55	1.71

Table 2: Media partners' datasets used for empirical evaluation of keyword extraction algorithms.

2018), TNT-KID requires an additional language model pretraining on the domain specific corpus.

4.2 TF-IDF(tm) Tagset Matching

In our approach, we first take the keywords returned by a neural keyword extraction method and next complement the returned keyword list by adding the missing keywords to achieve the set goal of k keywords. The added keywords are selected by taking the top-ranked candidates from the TF-IDF tagset matching extraction conducted on the preprocessed news articles and keywords.

4.2.1 Preprocessing

First, we concatenate the body and the title of the article. After that we lowercase the text and remove stopwords. Finally, the text is tokenized and lemmatized with the Lemmagen3 lemmatizer (Juršič et al., 2010), which supports lemmatization for all the languages except Latvian. For Latvian we use the LatvianStemmer [4]. For the stopword removal we used the *Stopwords-ISO* [5] Python library which contained stopwords for all four languages. The final cleaned textual input consists of the concatenation of all of the preprocessed words from the document. We apply the same preprocessing procedure on the predetermined tagsets for each language. The preprocessing procedure is visualized in Figure 1.
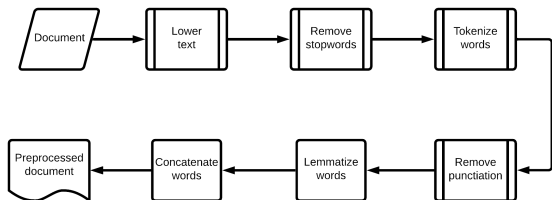

Figure 1: Preprocessing pipeline used for the document normalization and cleaning.

4.2.2 TF-IDF Weighting Scheme

The TF-IDF weighting scheme (Salton and McGill, 1986) assigns each word its weight w based on the frequency of the word in the document (term frequency) and the number of documents the word appears in (inverse document frequency). More specifically, TF-IDF is calculated with the following equation:

$$TF - IDF_i = tf_{i,j} \cdot \log_e(\frac{|D|}{df_i})$$

The formula has two main components:

- *Term-frequency* (tf) that counts the number of appearances of a word in the document (in the equation above, $tf_{i,j}$ denotes the number of occurrences of the word i in the document j)

- *Inverse-document-frequency* (idf) ensures that words appearing in more documents are assigned lower weights (in the formula above df_i is the number of documents containing word i and $|D|$ denotes the number of documents).

The assumption is that words with a higher TF-IDF value are more likely to be keywords.

4.3 Tagset Matching Keyword Expansion

For a given neural keyword extraction method N, and for each document d, we select l best ranked keywords according to the TF-IDF(tm), which appear in the keyword tagset for each specific dataset. Here, l corresponds to k - m, where $k = 10$ and m corresponds to the number of keywords returned by a neural method.

Since some of the keywords in the tagsets provided by the media partners were variations of the same root word (i.e., keywords are not lemmatized), we created a mapping from a root word (i.e., a word lemma or a stem) to a list of possible variations in the keyword dataset. For example, a word *'riigieksam'* (*'exam'*) appearing in the article, could be mapped to three tags in the tagset by the Estonian media house with the same root form *'riigieksam'*: *'riigieksamid'*, *'riigieksamide'* and *'riigieksam'*.

We tested several strategies for mapping the occurrence of a word in the news article to a specific tag in the tagset. For each lemma that mapped to multiple tags, we tested returning a random tag, a tag with minimal length and a tag of maximal length. In the final version, we opted to return the tag with the minimal length, since this tag corresponded to the lemma of the word most often.

5   Experimental Settings

We conducted experiments on the datasets described in Section 3. We evaluate the following methods and combinations of methods:

- **TF-IDF(tm):** Here, we employ the preprocessing and TF-IDF-based weighting of keywords described in Section 4 and select the top-ranked keywords that are present in the tagset.

- **TNT-KID** (Martinc et al., 2020b): For each dataset, we first pretrain the model with an autoregressive language model objective. After that, the model is fine-tuned on the same train set for the keyword extraction task. Sequence length was set to 256, embedding size to 512 and batch size to 8, and we employ the same preprocessing as in the original study (Martinc et al., 2020b).

- **BERT + BiLSTM-CRF** (Sahrawat et al., 2020): We employ an uncased multilingual BERT[6] model with an embedding size of 768 and 12 attention heads, with an additional BiLSTM-CRF token classification head, same as in Sahrawat et al. (2020).

- **TNT-KID & BERT + BiLSTM-CRF:** We extracted keywords with both of the methods and complemented the TNT-KID extracted keywords with the BERT + BiLSTM-CRF extracted keywords in order to retrieve more keywords. Duplicates (i.e., keywords extracted by both methods) are removed.

- **TNT-KID & TF-IDF:** If the keyword set extracted by TNT-KID contains less than 10 keywords, it is expanded with keywords retrieved with the proposed TF-IDF(tm) approach, i.e.,

best ranked keywords according to TF-IDF, which do not appear in the keyword set extracted by TNT-KID.

- **BERT + BiLSTM-CRF & TF-IDF:** If the keyword set extracted by BERT + BiLSTM-CRF contains less than 10 keywords, it is expanded with keywords retrieved with the proposed TF-IDF(tm) approach, i.e., best ranked keywords according to TF-IDF, which do not appear in the keyword set extracted by BERT + BiLSTM-CRF.

- **TNT-KID & BERT + BiLSTM-CRF & TF-IDF:** the keyword set extracted with the TNT-KID is complemented by keywords extracted with BERT + BiLSTM-CRF (duplicates are removed). If after the expansion the keyword set still contains less than 10 keywords, it is expanded again, this time with keywords retrieved by the TF-IDF(tm) approach.

For TNT-KID, which is the only model that requires language model pretraining, language models were trained on train sets in Table 2 for up to ten epochs. Next, TNT-KID and BERT + BiLSTM-CRF were fine-tuned on the training datasets, which were randomly split into 80 percent of documents used for training and 20 percent of documents used for validation. The documents containing more than 256 tokens are truncated, while the documents containing less than 256 tokens are padded with a special $< pad >$ token at the end. We fine-tuned each model for a maximum of 10 epochs and after each epoch the trained model was tested on the documents chosen for validation. The model that showed the best performance on this set of validation documents (in terms of F@10 score) was used for keyword detection on the test set.

6   Evaluation

For evaluation, we employ precision, recall and F1 score. While F1@10 and recall@10 are the most relevant metrics for the media partners, we also report precision@10, precision@5, recall@5 and F1@5. Only keywords which appear in a text (present keywords) were used as a gold standard, since we only evaluate approaches for keyword tagging that are not capable of finding keywords which do not appear in the text. Lowercasing and lemmatization (stemming in the case of Latvian) are performed on both the gold standard and the

---

Model	P@5	R@5	F1@5	P@10	R@10	F1@10
Croatian						
TF-IDF	0.2226	0.4543	0.2988	0.1466	0.5888	0.2347
TNT-KID	0.3296	0.5135	0.4015	0.3167	0.5359	0.3981
BERT + BiLSTM-CRF	**0.4607**	0.4672	**0.4640**	**0.4599**	0.4708	**0.4654**
TNT-KID & TF-IDF(tm)	0.2659	0.5670	0.3621	0.1688	0.6944	0.2716
BERT + BiLSTM-CRF & TF-IDF(tm)	0.2644	0.5656	0.3604	0.1549	0.6410	0.2495
TNT-KID & BERT + BiLSTM-CRF	0.2940	0.5447	0.3820	0.2659	0.5968	0.3679
TNT-KID & BERT + BiLSTM-CRF & TF-IDF(tm)	0.2648	**0.5681**	0.3612	0.1699	**0.7040**	0.2738
Estonian						
TF-IDF	0.0716	0.1488	0.0966	0.0496	0.1950	0.0790
TNT-KID	**0.5194**	0.5676	**0.5424**	**0.5098**	0.5942	**0.5942**
BERT + BiLSTM-CRF	0.5118	0.4617	0.4855	0.5078	0.4775	0.4922
TNT-KID & TF-IDF(tm)	0.3463	0.5997	0.4391	0.1978	0.6541	0.3037
BERT + BiLSTM-CRF & TF-IDF(tm)	0.3175	0.4978	0.3877	0.1789	0.5381	0.2686
TNT-KID & BERT + BiLSTM-CRF	0.4421	0.6014	0.5096	0.4028	0.6438	0.4956
TNT-KID & BERT + BiLSTM-CRF & TF-IDF(tm)	0.3588	**0.6206**	0.4547	0.2107	**0.6912**	0.3230
Russian						
TF-IDF	0.1764	0.2314	0.2002	0.1663	0.3350	0.2223
TNT-KID	**0.7108**	0.6007	**0.6512**	**0.7038**	0.6250	**0.6621**
BERT + BiLSTM-CRF	0.6901	0.5467	0.5467	0.6849	0.5643	0.6187
TNT-KID & TF-IDF(tm)	0.4519	0.6293	0.5261	0.2981	0.6946	0.4172
BERT + BiLSTM-CRF & TF-IDF(tm)	0.4157	0.5728	0.4818	0.2753	0.6378	0.3846
TNT-KID & BERT + BiLSTM-CRF	0.6226	0.6375	0.6300	0.5877	0.6707	0.6265
TNT-KID & BERT + BiLSTM-CRF & TF-IDF(tm)	0.4622	**0.6527**	0.5412	0.2965	**0.7213**	0.4203
Latvian						
TF-IDF	0.2258	0.5035	0.3118	0.1708	0.5965	0.2655
TNT-KID	0.6089	0.6887	**0.6464**	0.6054	0.6960	**0.6476**
BERT + BiLSTM-CRF	**0.6215**	0.6214	0.6214	**0.6204**	0.6243	0.6223
TNT-KID & TF-IDF(tm)	0.3402	**0.7934**	0.4762	0.2253	0.8653	0.3575
BERT + BiLSTM-CRF & TF-IDF(tm)	0.2985	0.6957	0.4178	0.1889	0.7427	0.3012
TNT-KID & BERT + BiLSTM-CRF	0.4545	0.7189	0.5569	0.4341	0.7297	0.5443
TNT-KID & BERT + BiLSTM-CRF & TF-IDF(tm)	0.3318	0.7852	0.4666	0.2124	**0.8672**	0.3414

Table 3: Results on the EMBEDDIA media partner datasets.

extracted keywords (keyphrases) during the evaluation. The results of the evaluation on all four languages are listed in Table 3.

Results suggest, that neural approaches, TNT-KID and BERT+BiLSTM-CRF offer comparable performance on all datasets but nevertheless achieve different results for different languages. TNT-KID outperforms BERT-BiLSTM-CRF model according to all the evaluation metrics on the Estonian and Russian news dataset. It also outperforms all other methods in terms of precision and F1 score. On the other hand, BERT+BiLSTM-CRF performs better on the Croatian dataset in terms of precision and F1-score. On Latvian TNT-KID achieves top results in terms of F1, while BERT+BiLSTM-CRF offers better precision.

Even though the TF-IDF tagset matching method performs poorly on its own, we can nevertheless

drastically improve the recall@5 and the recall@10 of both neural systems, if we expand the keyword tag sets returned by the neural methods with the TF-IDF ranked keywords. The improvement is substantial and consistent for all datasets, but it nevertheless comes at the expanse of the lower precision and F1 score. This is not surprising, since the final expanded keyword set always returns 10 keywords, i.e., much more than the average number of present gold standard keywords in the media partner datasets (see Table 2), which badly affects the precision of the approach. Nevertheless, since for a journalist a manual inspection of 10 keyword candidates per article and manual selection of good candidates (e.g., by clicking on them) still requires less time than the manual selection of keywords from an article, we argue that the improvement of recall at the expanse of the precision is a good trade

off, if the system is intended to be used as a recommendation system in the media house environment.

Combining keywords returned by TNT-KID and BERT + BiLSTM-CRF also consistently improves recall, but again at the expanse of lower precision and F1 score. Overall, for all four languages, the best performing method in terms of recall is the TNT-KID & BERT + BiLSTM-CRF & TF-IDF(tm).

7 Conclusion and Future Work

In this work, we tested two state-of-the-art neural approaches for keyword extraction, TNT-KID (Martinc et al., 2020a) and BERT BiLSTM-CRF (Sahrawat et al., 2020), on three less resourced European languages, Estonian, Latvian, Croatian, as well as on Russian. We also proposed a tagset based keyword expansion approach, which drastically improves the recall of the method, making it more suitable for the application in the media house environment.

Our study is one of the very few studies where supervised keyword extraction models were employed on several less resourced languages. The results suggest that these models perform well on languages other than English and could also be successfully leveraged for keyword extraction on morphologically rich languages.

The focus of the study was whether we can improve the recall of the supervised models, in order to make them more useful as recommendation systems in the media house environment. Our method manages to increase the number of retrieved keywords, which drastically improves the recall for all languages. For example, by combing all neural methods and the TF-IDF based approach, we improve on the recall@10 achieved by the best performing neural model, TNT-KID, by 16.81 percentage points for Croatian, 9.70 percentage points for Estonian, 9.63 percentage points for Russian and 17.12 percentage points for Latvian. The resulting method nevertheless offers lower precision, which we will try to improve in the future work.

In the future we also plan to perform a qualitative evaluation of our methods by journalists from the media houses. Next, we plan to explore how adding background knowledge from knowledge databases - lexical (e.g. Wordnet(Fellbaum, 1998)) or factual (e.g. WikiData(Vrandečić and Krötzsch, 2014)) would benefit the aforementioned methods. The assumption is that with the linkage of the text representation and the background knowledge we would achieve a more representative understanding of the articles and the concepts appearing in them, which would result in a more successful keyword extraction.

In traditional machine-learning setting a common practice of combining different classifier outputs to a single output is referred to as stacking. We propose further research on this topic by testing combinations of various keyword extraction models. Finally, we also plan to further improve our unsupervised TF-IDF based keyword extraction method. One way to to do this would be to add the notion of positional encoding, since some of the keywords in the news-media domain often can be found at the beginning of the article and the TF-IDF(tm) does not take this into account while applying the weighting on the matched terms.

8 Acknowledgements

References

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International joint conference on natural language processing (IJCNLP)*, pages 543–551.

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 arXiv:1810.04805*.

Samhaa R. El-Beltagy and Ahmed Rafea. 2009. Kpminer: A keyphrase extraction system for english and arabic documents. *Inf. Syst.*, 34(1):132–144.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Matjaž Juršič, Igor Mozetič, Tomaž Erjavec, and Nada Lavrač. 2010. Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16(9):1190–1214.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

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 arXiv:1907.11692*.

Matej Martinc, Blaž Škrlj, and Senja Pollak. 2020a. Tnt-kid: Transformer-based neural tagger for keyword identification. *arXiv preprint arXiv:2003.09166*.

Matej Martinc, Blaž Škrlj, and Senja Pollak. 2020b. Tnt-kid: Transformer-based neural tagger for keyword identification.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. *arXiv preprint arXiv:1704.06879*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Senja Pollak, Marko Robnik Šikonja, Matthew Purver, Michele Boggia, Ravi Shekhar, Marko Pranjić, Salla Salmela, Ivar Krustok, Tarmo Paju, Carl-Gustav Linden, Leo Leppänen, Elaine Zosa, Matej Ulčar, Linda Freienthal, Silver Traat, Luis Adrián Cabrera-Diego, Matej Martinc, Nada Lavrač, Blaž Škrlj, Martin Žnidaršič, Andraž Pelicon, Boshko Koloski, Vid Podpečan, Janez Kranjc, Shane Sheehan, Hannu Toivonen, Emanuela Boros, Jose Moreno, and Antoine Doucet. 2021. EMBEDDIA tools, datasets and challenges: Resources and hackathon contributions. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20.

Dhruva Sahrawat, Debanjan Mahata, Mayank Kulkarni, Haimin Zhang, Rakesh Gosangi, Amanda Stent,

Agniv Sharma, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings. In *Proceedings of European Conference on Information Retrieval (ECIR 2020)*, pages 328–335.

Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval.

Blaž Škrlj, Andraž Repar, and Senja Pollak. 2019. Rakun: Rank-based keyword extraction via unsupervised learning and meta vertex aggregation. In *International Conference on Statistical Language and Speech Processing*, pages 311–323. Springer.

Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2015. Topical word importance for fast keyphrase extraction. In *Proceedings of the 24th International Conference on World Wide Web*, pages 121–122.

Si Sun, Chenyan Xiong, Zhenghao Liu, Zhiyuan Liu, and Jie Bao. 2020. Joint keyphrase chunking and salience ranking with bert. *arXiv preprint arXiv:2004.13639*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, volume 8, pages 855–860.

Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. 2005. Kea: Practical automated keyphrase extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, pages 129–152. IGI global.