# What works for Slovenian? A comparative study of different keyword extraction systems

**Boshko Koloski, Senja Pollak, Matej Martinc**

Jožef Stefan Institute, Jožef Stefan International Postgraduate School
Jamova cesta 39, Ljubljana, Slovenia
{boshko.koloski,senja.pollak,matej.martinc}@ijs.si

**Abstract**

Identifying and retrieving keywords from a given document is one of the fundamental problems of natural language processing. In this paper, we conduct a thorough comparative analysis of several distinct approaches for keyword identification on a new benchmark Slovenian keyword extraction corpus, SentiNews. The first group of methods is based on a supervised methodology, where previously annotated data is required for the models to learn. We evaluate two such approaches, *TNT-KID* and *BERT* . The other paradigm relies on unsupervised approaches, where no previously annotated data for training is needed. We evaluate five different unsupervised approaches, covering three main types of unsupervised systems: statistical, graph-based and embedding-based. The results show that supervised models perform significantly better than unsupervised approaches. By applying the *TNT-KID* method on the Slovenian corpus for the first time, we also advance the state-of-the-art on the SentiNews corpus.

## 1. Introduction

Identifying and retrieving keywords from a given document represents one of the crucial tasks for organization of textual resources. It is employed extensively in media organizations with large daily article production that needs to be categorized in a fast and efficient manner. While some media houses use keywords to link articles and produce networks based on keywords, journalists use keywords to search for news stories related to newly produced articles and also to summarize new articles with a handful of words. Manual categorization and tagging of these articles is a burdensome and time demanding task, therefore development of algorithms capable of tackling keyword extraction automatically, and therefore allowing the journalists to spend more time on more important investigative assignments, has become a necessity.

The approaches for automatic detection of keywords can be divided based on their need for annotated data prior to learning. One paradigm of keyword extraction focuses on extracting keywords without prior training (i.e. unsupervised approaches), while the other focuses on learning to identify keyphrases from an annotated data-set (i.e. supervised approaches). While unsupervised approaches can be easily applied for domains and languages that have low to no amount of labeled data, they nevertheless tend to offer non-competitive performance when compared to supervised approaches (Martinc et al., 2020), since they can not be adapted to the specific language and domain through training. On the other hand, supervised state-of-the-art approaches based on the transformer architecture (Vaswani et al., 2017) have become very effective in solving the task, but they do usually require substantial amounts of labeled data which is hard to obtain for some low-resource domains and languages.

In this research, we focus on one of the low-resource languages, Slovenian, for which not a lot of manually labeled data that could be leveraged for training of keyword extractors, is available. We systematically evaluate several distinct strategies for keyword extraction on Slovenian, among them also some, which have not been tested before on Slovenian. We show that the employment of the TNT-KID model (Martinc et al., 2020), a model specifically adapted for the monolingual low-resource scenario, leads to advance in state-of-the-art on the Slovenian SentiNews keyword extraction benchmark dataset (Bučar, 2017). To summarize, the main contributions of this work include:

- A systematical analysis of a keyword extraction dataset of Slovenian news.
- Thorough comparison of several supervised and unsupervised keyword extraction strategies on the Slovenian data-set. Supervised methods include the monolingual TNT-KID method, which has not been employed for Slovenian before, and an application of the multilingual BERT model (Devlin et al., 2019), same as in Koloski et al. (2022b). We also cover several unsupervised methods in this study, including statistical, graph-based and embedding based models.
- The advancement in state-of-the-art on the Slovenian keyword extraction dataset from SentiNews
- Release of a dockerized pretrained model of the best performing system TNT-KID-Slovene in terms of F1-score.

The paper is organized in the following manner: Section 2. describes the related work in the field, followed by the description of data and the exploratory data analysis in Section 3. Section 4. describes the experimental setting considered in this study and in Section 5., we discuss the results. Finally, Section 6. presents the conclusions of the study and proposes further work.

## 2. Related work

Keyword extraction approaches are either supervised or unsupervised.

### 2.1. Unsupervised methods

Modern supervised learning approaches are very successful in keyword extraction, but they are data intensive and time consuming. Unsupervised keyword detectors can address both problems and typically require much less computational resources and no training data, but this comes with the price of lower overall performance. Unsupervised methods can be divided into four main categories:

- *statistical* - methods that belong to this family are based on calculating various text statistics to capture keywords, such as frequency of appearance, position in the text, *etc*. KPMiner (El-Beltagy and Rafea, 2009) is one of the oldest methods and focuses on the frequency and position of a given keyphrase. After calculating several frequency based statistics, the method uses post-processing filtering to remove some keyphrases that are too rare or that are not positioned within the first $k$ characters of the document. YAKE (Campos et al., 2018) represents one of the latest upgrades of the statistical approaches, and includes the simpler features proposed by the KPMiner. The main novelty is that it also considers the relatedness of term candidates to general document context, dispersion, and casing of a specific term candidate.

- *graph-based* - methods focus on creating graphs from a given document and then exploit graph properties in order to rank words and phrases. In the first, graph creation step, authors usually consider two adjacent words as two adjacent nodes in a graph *G*. Usually before the graph-creation step some form of word normalization is performed - either stemming or lemmatisation. Since keyword phrases can consist of multiple words, the methods consider the use of a sliding windows to obtain n-grams up to specific value of n, and using obtained n-grams as nodes. Text Rank (Mihalcea and Tarau, 2004) is one of the first such methods. In the second, keyword ranking step, it leverages Google's PageRank (Page et al., 1999) algorithm to rank the nodes according to their importance within the graph *G*. While TextRank is a robust method, it does not account for the position of a given term in the document. This was improved in the PositionRank (Florescu and Caragea, 2017) method that leverages PageRank on one side, and the position of a given term on the other side. An upgrade to the graph-creation step was introduced in Boudin (2018), where they consider encoding the potential keywords into a multipartite[1] graph structure. The method in addition also considers topic information. Similarly to TextRank it leverages PageRank (Page et al., 1999) to rank the nodes. RaKUn (Škrlj et al., 2019) is one of the most recent additions to the family of graph based keyword extractors. The main contribution of this method is that it introduces an intermediate step, that constructs meta-nodes from the initial nodes of the graph via aggregation of the existing nodes. After the construction of the meta-graph, it applies the *load centrality* metric for the term ranking, and also relies on multiple graph redundancy measures.

- *embedding-based* methods are gaining traction with the recent introduction of various off-the shelf pretrained embeddings such as FastText (Bojanowski et al., 2016) or transformer - BERT (Devlin et al., 2019) based embeddings. Key2Vec (Mahata et al., 2018) represents the pioneer of this type of methods, followed by the EmbedRank (Bennani-Smires et al., 2018) method. The aforementioned methods consider the semantic information captured by the distributed word and sentence embedding representations. KeyBERT (Grootendorst, 2020) is currently the state-of-the-art method of the type. The foundation of this method are pre-trained sentence-BERT (Reimers and Gurevych, 2019) based representations. The method considers embedding n-grams of a given size and compares them to the embedding of the entire document. The n-grams closely matching the representation of an entire document (i.e. keywords most representative of an entire document) are retrieved as keywords that best describe the overall document content. In order to diversify the results, the method also introduces the *Max Sum Similarity* metric with which the model selects the candidate phrases with the highest rank that are least similar to each other.

- *language model-based* - methods use language model derived statistics to extract keywords from text. Tomokiyo and Hurst (2003) considered multiple language models and measured the Kullback-Leibler Divergence (Joyce, 2011) for ranking both phrasesness and the informativeness of candidate terms.

### 2.2. Supervised methods

Supervised methods require manually annotated data for training. The methods can be divided into neural and non-neural.

#### 2.2.1. Non-neural

The first methods that proposed a solution in a supervised manner, considered keyword extraction as a classification task. The KEA method (Witten et al., 1999) treats each word or phrase as a potential keyword, and uses TF-IDF (Sammut and Webb, 2010) metric and word position for representation, and Naive Bayes for classification of a given term as a keyword or not.

#### 2.2.2. Neural

With the recent-gain in computing power and introduction of more modern deep architectures, the field of keyword extraction was taken by storm of neural architectures. The neural approaches can be divided are two groups: one that treat the task as a sequence-to-sequence generation and the one that model the task as sequence-labelling.

Meng et al. (2017) first proposed the idea of keyword extraction as a sequence-to-sequence generation task. In their work they proposed a recurrent generative model with an attention and a copying mechanism (Gu et al., 2016)

[1]Family of graphs where the nodes can be split into multiple disjoint sets.

based on the positional information. An additional strongpoint of this model is that is able to find keywords that do not appear in the text due to it's generative nature.

The first representative of the sequence-labelling method is the approach by Luan et al. (2017), where the authors consider bidirectional Long Short-Term Memory (BiLSTM) layer and a conditional random field (CRF) layer for classification. The more recent approaches of this type utilize the transformer architecture (Vaswani et al., 2017) in their models. An upgrade of the approach by Luan et al. (2017) was proposed by Sahrawat et al. (2020), where contextual embeddings generated by BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and GPT-2 (Radford et al., 2019) were fed into the BiLSTM network. Currently, the state-of-the-art model based on the transformer architecture is the one proposed by Martinc et al. (2020). They employ the tactic of not relying on the massive language model pretraining but rather on the language model pretraining on the much smaller domain specific corpora. This makes the approach more easily transferable to less resourced domains and languages.

Most keyword recognition studies still focus on English. Nevertheless, several multilingual and cross-lingual studies have been conducted recently, also including low-resource languages. One of them is the study by Koloski et al. (2021), which compared the performance of two supervised transformer-based models, a multilingual BERT with a BiLSTM-CRF classification head (Sahrawat et al., 2020) and TNT-KID, in a multilingual setting with Estonian, Latvian, Croatian and Russian news corpora. The authors also investigated whether combining the results of the supervised models with the results of the unsupervised models can improve the recall of the system. In Koloski et al. (2022b), an extensive study was conducted to compare the performance of supervised zero-shot cross-lingual approaches with unsupervised approaches. The study was conducted for six languages - Slovenian, English, Estonian, Latvian, Croatian, and Russian. The authors show that models fine-tuned to extract keywords on a combination of languages *outperform* the unsupervised models, when evaluated on a new previously unseen language not included in the training dataset.

## 3. Data

We conduct our experiments on the Slovenian SentiNews dataset (Bučar, 2017), which was originally used for news sentiment analysis, but nevertheless does contain manually labeled keywords and was therefore identified as suitable for keyword extraction (Koloski et al., 2022a). Before feeding the datasets to the models, they are lowercased. We split the dataset into three different splits: *train*, *validation* and *test*.

### 3.1. Exploratory data analysis

Next, we preform exploratory data analysis (*EDA*) on the given dataset. There are total of *7514* documents, *4796 (64%)* for training, *1199 (16%)* for validation and *1519 (20%)* for testing, which makes the dataset relatively small in comparison to some English keyword extraction datasets, such as for example KPTimes (Gallina et al., 2019), containing more than 200,000 documents. We benchmark all of our models on the same test split that was already used in the study by (Koloski et al., 2022b), in order to make our results directly comparable to the ones in the related work.

The documents have a similar structure in all of the three splits, having on average *370* words (*370.10* words in the train split, *366.89* words in the validation split and *377.46* words in the test split) and on average around *15* sentences (*15.419* sentences in the train split, *15.203* sentences in the validation split and *15.662* sentences in the test split).

| | Split | | |
|---|---|---|---|
| Property | Train | Valid | Test |
| *Document statistics* | | | |
| # of documents | 4796 | 1199 | 1519 |
| avg. # of sentences | 15.419 | 15.2026 | 15.6622 |
| avg. # of words | 370.10 | 366.89 | 377.46 |
| *Keywords statistics* | | | |
| # of keywords | 19429 | 4773 | 5903 |
| # of unique keywords | 4414 | 1854 | 2049 |
| # of unique keywords per document | 0.9203 | 1.5462 | 1.3489 |
| # of keywords per document | 4.0052 | 4.1643 | 3.8861 |
| keywords present in the document | 59.91 % | 60.54 % | 59.95 % |
| *Keyword composition statistics* | | | |
| Proportion of 1-word terms | 92.77% | 93.17 % | 92.68 % |
| Proportion of 2-word terms | 5.88 % | 5.61 % | 5.98 % |
| Proportion of 3-word terms | 0.62 % | 0.57 % | 0.58 % |
| Proportion of more than 3-word terms | 0.74 % | 0.65 % | 0.76 % |

Table 1: Dataset statistics. We conducted three different statistical analyses. The first one was on the document level and it considered counting the word and sentence tokens. The second focused on the keyword level statistics, such as total number of keywords, number of unique keywords, and the proportion of all versus unique keywords per document. Finally, we explored the composition of keywords, i.e. how many of them were composed of single words, two words, three words or more words.

There are in total *30,105* keywords in the dataset, with *8,317* of them being unique. On average there are *4* keywords per document in the training split, *4.16* keywords per document in the validation split and *3.8861* keywords per document in the test split. In regards to the unique keywords per split, there are *0.92* unique keywords per document in the training split, *1.55* in the validation split and *1.35* keywords per document in the test split. Since the keyword extractors used in this study are only able to extract keywords that are present in the data, we also calculated the share of keywords that are present in the document. In the training set, there were $59.91\%$ of the keywords present, in the validation set $60.54\%$ and in the testing set $59.95\%$.

Finally, we conducted a study on the composition of keywords in which we explored how many words constitute a specific keyphrase. In all of the splits, more than $92\%$ of the keywords contained only a single words, 2-word terms represented about $5\%$ of the keywords, while 3 or more word terms represented around $3\%$ of all keywords. The most common keyword was *gospodarstvo* with *2,350* occurrences (representing roughly 12% of all keyword occurences), followed by *ekonomija* with *1315 (6.76%)* oc-

currences, followed by *banka* with *147 (0.08%)* occurrences.

These keywords suggest that most of the articles come from the economic and financial domain. In order to explore the structure and content of the dataset in more detail, we do additional network science analysis on the graph of 100 most-frequent terms. We construct a graph $G_{100}$ in the following manner: we create links among every pair of keywords that accompany a given article in the training split. We repeat the step for every article in the training split.

We next focus on community detection in the constructed graph. For that purpose, we use the Louvain algorithm (Blondel et al., 2008). The algorithm detects four distinct communities. The first one colored *green* is the most central community - the community with the highest amount of shared links with the three other detected communities. It contains general terms like *family*, *declaration*, *NKB(a bank)*, *sod*. Next one is *purple* and it talks about the trend of rising *taxes*, new *laws* and the petrochemical industry. The community colored in *blue* represents the economic news about *infrastructure* and *construction* industries. The last is the *yellow* community that talks about *financial help* from the government and the *European union*, accompanied by the *unemployment* and the slow rise of *GDP*. The graph and its detected communities are presented in Figure 1.

## 4. Methods

In our experiments, we follow the experimental setting proposed in Koloski et al. (2021) and Koloski et al. (2022b). The methods and the hyperparameters used are described below.

### 4.1. Unsupervised approaches

We evaluate three types of unsupervised keyword extraction methods, statistical, graph-based, and embedding-based, described in Section 2. Note that these models were already evaluated on the same corpus in Koloski et al. (2022b).

#### 4.1.1. Statistical methods

- **YAKE** (Campos et al., 2018): We consider n-grams with $n \in \{1, 2, 3\}$ as potential keywords.
- **KPMiner** (El-Beltagy and Rafea, 2009): We apply least allowable seen frequency of 3, while we set the *cutoff* to 400.

#### 4.1.2. Embedding-based methods

- **KeyBERT** (Grootendorst, 2020): For document embedding generation we employ sentence-transformers (Reimers and Gurevych, 2019), more specifically the *distiluse-base-multilingual-cased-v2* model available in the Huggingface library[2]. Initially, we tested two different KeyBERT configurations: one with n-grams of size 1 and another with n-grams ranging from 1 to 3, with *MMR*=$false$ and with *MaxSum*=$false$. The unigram model outscored the model that considered n-grams of sizes 1 to 3 as keyword candidates for all languages, therefore in the final report we show only the results for the unigram model.

#### 4.1.3. Graph-based methods

- **MultipartiteRank** (Boudin, 2018): We set the minimum similarity threshold for clustering at 74%.
- **RaKUn** (Škrlj et al., 2019): We use edit distance for calculating distance between nodes, and remove stopwords (using the *stopwords-iso* library[3]), a *bigram-count_threshold* of 2 and a *distance_threshold* of 2. An example graph of the RaKUn document representation and its predicted keywords are presented in Figure 2.

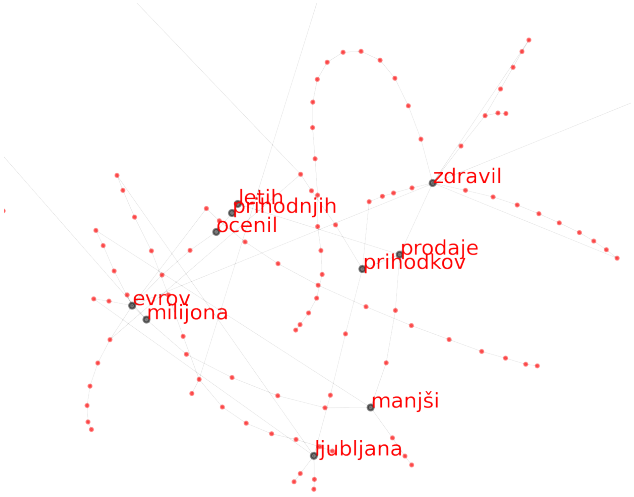


Figure 2: Visualization of one training example as it was seen by the RaKUn method. The visualization is generated via the Py3Plex (**?**) library. Top three extracted tokens here are *Ljubljana*, *Prihodki* and *Zdravil* - depicting that the article is about purchase of *medicine*.

We use the PKE (Boudin, 2016) implementations of *YAKE*, *KPMiner* and *MultiPartiteRank*. We use the official implementation for the RaKUn (Škrlj et al., 2019) and for the KeyBERT model (Grootendorst, 2020). For unsupervised models, the number of returned keywords need to be set in advance. Since we employ F1@10 as the main evaluation measure (see Section 4.3.), we set the number of returned keywords to 10 for all models.

### 4.2. Supervised approaches

We test two distinct state-of-the-art transformer-base models, BERT (Devlin et al., 2019) and TNT-KID (Martinc et al., 2020).

#### 4.2.1. BERT sequence labelling

As a strong baseline, we utilize the transformer-based BERT model (Devlin et al., 2019) with a token-classification head consisting of a simple linear layer for

[2] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2

[3] https://github.com/stopwords-iso/stopwords-iso

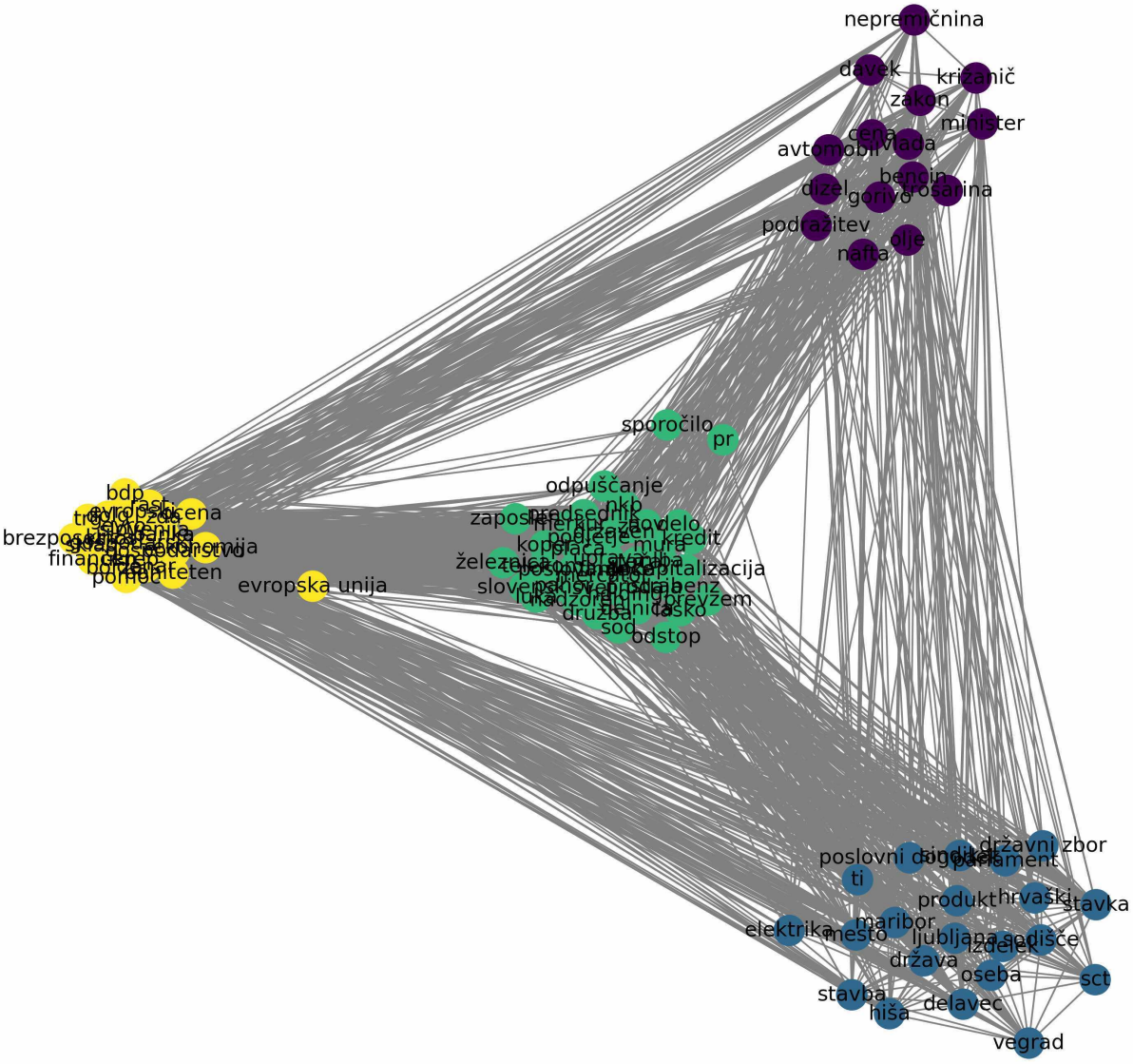


Figure 1: Visualization of the derived communities of the co-occurrence graph.

all our supervised approaches. We treat the keyword extraction task as a sequence classification task. We follow the approach proposed in Martinc et al. (2020) and predict binary labels (1 for 'keywords' and 0 for 'not keywords') for all words in the sequence. The sequence of two or more sequential keyword labels predicted by the model is always interpreted as a multi-word keyword. More specifically, we employ the *bert-uncased-multilingual* model from the HuggingFace library (Wolf et al., 2019) and fine-tune it on the SentiNews train split using an adaptive learning rate (starting with the learning rate of $3 \cdot 10^{-5}$), for up to 10 *epochs* with a batch-size of 8. Note that we chose this model since it is the best performing model on the Slovenian SentiNews dataset according to the study by Koloski et al. (2022b).

#### 4.2.2. TNT-KID sequence labelling

Same as for BERT, we follow the approach proposed in Martinc et al. (2020) and predict binary labels (1 for 'keywords' and 0 for 'not keywords') for all words in the sequence. Again, the sequence of two or more sequential keyword labels predicted by the model is always interpreted as a multi-word keyword. We first pretrain TNT-KID as an autoregressive language model on the domain specific news corpus containing *884,407* news articles crawled from websites of several Slovenian news outlets. The model was trained for 10 epochs. After that, the model was fine-tuned on the SentiNews train set for the keyword extraction task, again for up to 10 epochs. 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., 2020).

### 4.3. Evaluation setting

To evaluate the models, we compute F1, Recall, and Precision on 10 retrieved words. We next formally represent the Recall@10 metric:

$$Recall@10 = \frac{(\#\text{ of recommended relevant items @ 10})}{(\text{total }\#\text{ of relevant items}).}$$

and Precision@10 metric:

$$Precision@10 = \frac{(\#\text{ of recommended relevant items @ 10})}{(\#\text{ of recommended items @10})}$$

We omit the documents in which there are no keywords or which do not contain keywords. We do this because we only use approaches that extract words (or multi-word expressions) from the given document and cannot process keywords that do not appear in the text. All approaches are evaluated on the same monolingual test splits, which are not used for training the supervised models. Lower case and lemmatization are performed during the evaluation for both the gold standard and the extracted keywords (keyphrases).

## 5. Results

In this section we examine the results of the evaluation of the proposed models. We first study the results of the unsupervised methods and later the results of the supervised models.

### 5.1. Unsupervised methods

In this study we evaluate 5 different unsupervised methods: 2 statistical, 1 embedding-based and 2 graph-based methods. Comparing the two statistical methods, *KPMiner* outscored the *YAKE* method in terms of f1-score and precision. The embedding based *KeyBERT* method achieved the best results when compared to other unsupervised methods. From the graph-based methods, *RaKUn* performed the best in comparison with the *MPRU* method, achieving nearly $100\%$ relative improvement. Table 2 presents the results for all systems and evaluation metrics in detail.

### 5.2. Supervised methods

We use two different supervised methods based on the sequence labeling paradigm. BERT based model outperforms TNT-KID in terms of recall by about $5$ percentage points, achieving the best recall out of all models. In terms of precision, TNT-KID outscores the BERT model by $9.04$ percentage points and achieves the best precision@10 score - $38.58\%$. We believe this is due to the extensive language-model pretraining on a large domain specific Slovenian news corpus and the frequency of common co-occurrence patterns in the data, that TNT-KID has learned to exploit successfully.

| Model | precision@10 | recall@10 | f1-score@10 |
|---|---|---|---|
| *Statistical* | | | |
| KPMiner | *12.80* | 7.44 | 9.41 |
| YAKE | 5.91 | *12.13* | 7.94 |
| *Embedding-based* | | | |
| KeyBert | 12.13 | 12.00 | 11.53 |
| *Graph-based* | | | |
| RaKUn | *6.72* | *12.52* | *8.75* |
| MPRU | 3.39 | 6.96 | 4.55 |
| Sequence-labelling | | | |
| BERT | 29.54 | **47.81** | 32.59 |
| TNT-KID | ***38.58*** | 42.81 | ***40.59*** |

Table 2: Comparison of the evaluation of the proposed approaches. We report on the precision@10, recall@10 and f1-score@10. The scores of the best performing system of a specific type (i.e. statistical, embedding-based, graph-based or sequence-labelling based) are written in italic. The scores for the overall best-preforming model according to each metric are written in bold and presented in percents.

The final comparison of both the unsupervised and supervised models is presented in Table 2. The *TNT-KID* model performed the best in terms of precision and F1-score while *BERT* model performed the best out of all models in terms of recall. The supervised models outscored the unsupervised models by a large margin on the given task. The ranking of the models in terms of various metrics is given in Figure 3.

## 6. Conclusion and further work

In this study, we compared the performance of supervised and unsupervised keyword extraction methods on the new public benchmark for keyword extraction, derived from Slovenian SentiNews corpus. We have compared 8 different models, among them also TNT-KID, which has not been tested on Slovenian dataset yet. Five unsupervised approaches can be further divided into two graph-based, two statistical and one embedding-based approach. The embedding-based method *KeyBERT* showcased superior performance to the other unsupervised methods in terms of F1-score at 10 retrieved keywords.

When it comes to supervised approaches, we experimented with two transformer based models - one leveraging multilingual BERT and the other the TNT-KID method - that model keyword extraction as a sequence labelling task. The TNT-KID approach outperformed BERT-based approach (and all unsupervised models) in terms of precision and F1-score. These results therefore support the claims of the original study by (Martinc et al., 2020) that TNT-KID can be easily adapted for employment on less-resource languages, such as Slovenian, by domain specific unsupervised language model pretraining. By employing TNT-KID on the SentiNews dataset, we have advanced the state-of-the-art on the benchmark Slovenian keyword extraction dataset.

For further work, we plan to explore how potentially we can improve the results by constructing ensembles of keyword extractors. We will also propose testing several

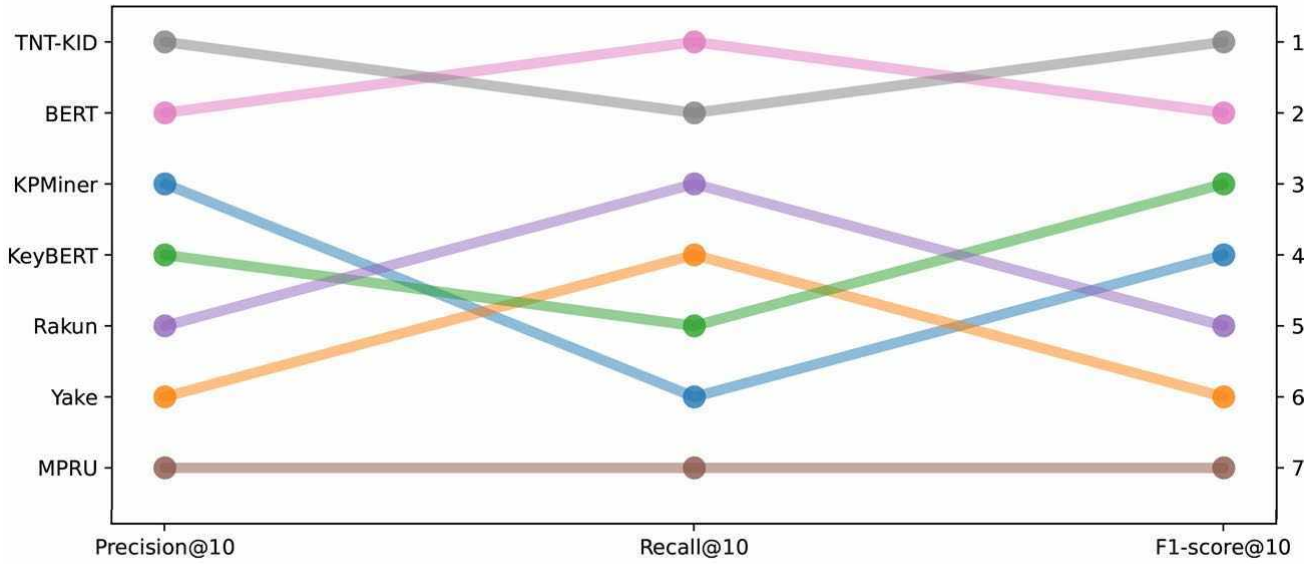


Figure 3: Comparison of the models ranking with respect to Precision@10, Recall@10 and F1-score@10.

different data splitting strategies, in order to study the possible effect of different splitting strategies on performance of different models and to establish the best possible split strategy. We also hypothesize that a possible improvement can be introduced by taking into account the co-occurrence of various pairs of keywords. Finally, in the future we plan to expand our experiments to also include the recently introduced monolingual massively pretrained model for Slovenian, SloBERTa (Ulčar and Robnik-Šikonja, 2020). We plan to fine-tune this model for the keyword extraction task and compare it to the TNT-KID, to check whether state-of-the-art can be advanced even further.

## 7. Availability

The best-performing *TNT-KID* based model is available as a docker model on the following link `https://gitlab.com/boshko.koloski/tnt_kid_app_slo`.

## 8. Acknowledgements

The authors acknowledge the financial support from the Slovenian Research Agency for research core funding for the programme Knowledge Technologies (No. P2-0103) and the project Computer-assisted multilingual news discourse analysis with contextual embeddings (CANDAS, J6-2581).

## 9. References

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium, October. Association for Computational Linguistics.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, oct.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Florian Boudin. 2016. PKE: an open source Python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 69–73, Osaka, Japan, December.

Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. *CoRR*, abs/1803.08721.

Jože Bučar. 2017. Manually sentiment annotated slovenian news corpus SentiNews 1.0. Slovenian language resource repository CLARIN.SI.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. Yake! collection-independent automatic keyword extractor. In *European Conference on Information Retrieval*, pages 806–810. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding.

Samhaa R El-Beltagy and Ahmed Rafea. 2009. KP-Miner: A keyphrase extraction system for English and Arabic documents. *Information systems*, 34(1):132–144.

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 (Volume 1: Long Papers)*, pages 1105–1115, Vancouver, Canada, July. Association for Computational Linguistics.

Ygor Gallina, Florian Boudin, and Béatrice Daille. 2019. Kptimes: A large-scale dataset for keyphrase generation on news documents. In *Proceedings of the 12th Inter-*

*national Conference on Natural Language Generation*, pages 130–135.

Maarten Grootendorst. 2020. KeyBERT: Minimal keyword extraction with bert.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August. Association for Computational Linguistics.

James M. Joyce, 2011. *Kullback-Leibler Divergence*, pages 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg.

Boshko Koloski, Senja Pollak, Blaž Škrlj, and Matej Martinc. 2021. Extending neural keyword extraction with TF-IDF tagset matching. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 22–29, Online, April. Association for Computational Linguistics.

Boshko Koloski, Matej Martinc, Ilija Tavchioski, Blaž Škrlj, and Senja Pollak. 2022a. Slovenian keyword extraction dataset from SentiNews 1.0. Slovenian language resource repository CLARIN.SI.

Boshko Koloski, Senja Pollak, BlaÅ¾ Å krlj, and Matej Martinc. 2022b. Out of thin air: Is zero-shot cross-lingual keyword detection better than unsupervised? In *Proceedings of the Language Resources and Evaluation Conference*, pages 400–409, Marseille, France, June. European Language Resources Association.

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*.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2641–2651, Copenhagen, Denmark, September. Association for Computational Linguistics.

Debanjan Mahata, John Kuriakose, Rajiv Ratn Shah, and Roger Zimmermann. 2018. Key2Vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 634–639, New Orleans, Louisiana, USA, June. Association for Computational Linguistics.

Matej Martinc, Blaž Škrlj, and Senja Pollak. 2020. Tnt-kid: Transformer-based neural tagger for keyword identification. *Natural Language Engineering*, pages 1–40.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Vancouver, Canada, July. Association for Computational Linguistics.

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.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November. Previous number = SIDL-WP-1999-0120.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

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, Lisbon, Portugal. Springer.

Claude Sammut and Geoffrey I. Webb, editors, 2010. *TF–IDF*, pages 986–987. Springer US, Boston, MA.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18*, page 33–40, Sapporo, Japan. Association for Computational Linguistics.

Matej Ulčar and Marko Robnik-Šikonja. 2020. Slovenian roBERTa contextual embeddings model: SloBERTa 1.0.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, Vancouver, Canada. Curran Associates, Inc.

Blaž Škrlj, Andraz Repar, and Senja Pollak. 2019. Rakun: Rank-based keyword extraction via unsupervised learning and meta vertex aggregation. *CoRR*, abs/1907.06458.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, page 254–255, Berkeley, California, USA. Association for Computing Machinery.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.