Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 400–409
Marseille, 20-25 June 2022

Out of Thin Air: Is Zero-Shot Cross-Lingual Keyword Detection Better Than Unsupervised?

Boshko Koloski, Senja Pollak, Blaž Škrlj, Matej Martinc
Jožef Stefan Institute, Jožef Stefan International Postgraduate School
Jamova cesta 39, Ljubljana, Slovenia
{boshko.koloski,senja.pollak,blaz.skrlj,matej.martinc}@ijs.si

Abstract

Keyword extraction is the task of retrieving words that are essential to the content of a given document. Researchers proposed various approaches to tackle this problem. At the top-most level, approaches are divided into ones that require training (supervised) and ones that do not (unsupervised). In this study, we are interested in settings, where for a language under investigation, no training data is available. More specifically, we explore whether pretrained multilingual language models can be employed for zero-shot cross-lingual keyword extraction on low-resource languages with limited or no available labeled training data and whether they outperform state-of-the-art unsupervised keyword extractors. The comparison is conducted on six news article datasets covering two high-resource languages, English and Russian, and four low-resource languages, Croatian, Estonian, Latvian, and Slovenian. We find that the pretrained models fine-tuned on a multilingual corpus covering languages that do not appear in the test set (i.e. in a zero-shot setting), consistently outscore unsupervised models in all six languages.
Keywords: keyword detection, cross-lingual learning, zero-shot learning

1. Introduction

Detecting keywords represents a crucial task in several text intensive applications. News industry relies on keywords for organization, linking and summarization of articles according to the content and topics they cover. With the current trend of fast-paced type of writing and an ever-growing amount of generated news, it becomes an infeasible task for the journalists to manually extract keywords and the development of tools for automatic extraction has become essential for speeding up the media production.

Keyword extraction can be tackled in a supervised or an unsupervised way. The current supervised state-of-the-art approaches are based on transformer-based (Vaswani et al., 2017) deep neural networks and employ large-scale language model pretraining. Despite being very successful in solving the task, they do require substantial amounts of labeled data which is expensive to obtain or non-existent for some low-resource languages and domains. To cope with this, researchers in most cases employ unsupervised keyword extraction in these low-resource scenarios. Unsupervised approaches require no prior training and can be applied to most languages, making them a perfect fit for domains and languages that have low to no amount of labeled data. On the other hand, they offer non-competitive performance when compared to supervised approaches (Martinc et al., 2020), since they can not be adapted to the specific language, domain and keyword assignment regime through training.

In this work, we explore another option for keyword extraction in low-resource settings, which has not been extensively explored in the past, a zero-shot cross-lingual keyword detection. More specifically, we investigate how multilingual pretrained language models, which have been fine-tuned to detect keywords on a set of languages, perform, when applied to a new language not included in the train set, and compare these results to the results achieved by several state-of-the-art unsupervised keyword extractors. In addition, we also investigate whether in a setting, where training data is available, supervised monolingual models can benefit from additional data from another language[1]. The main contributions are the following:

- We conduct and extensive zero-shot cross-lingual study of keyword extraction on six languages, four of them less-resourced European languages, and demonstrate that a multilingual BERT model fine-tuned on the training data not matching target language, performs better than state-of-the-art unsupervised keyword extraction algorithms.
- We evaluate the performance of supervised zero-shot cross-lingual models in comparison to the supervised monolingual models in order to better determine the decrease in the performance when no language specific data is available.
- We investigate if the performance of monolingual models can be improved by including additional multilingual data and whether there is a trade-off between the amount of data available and the language specificity of this data.
- We produce new supervised keyword extraction models for a new *Slovenian* dataset for keyword extraction, contributing to the development of new

[1]The code for all the experiments is available under the MIT license at `https://github.com/bkolosk1/CrossLingualKeywords`.

language resources for a less-resourced European language.

The rest of this paper is organized in the following way: Section 2 presents the related work in the field of keyword extraction, focusing also on the cross-lingual zero-shot learning. Section 3 describes the data used in our experiments and Section 4 explains our experimental settings. While Section 5 presents and discuses the results of our experiments, Section 6 concludes the paper and proposes further work on this topic.

2. Related Work

We can divide approaches for keyword extraction into supervised and unsupervised. As stated above, state-of-the-art supervised learning approaches have become very successful at tackling the keyword extraction task but are data-intensive and time consuming. Unsupervised keyword detectors can tackle these two problems and usually require a lot less computational resources and no training data, yet this comes at the cost of the reduced overall performance.

We can divide unsupervised approaches into four main categories, namely statistical, graph-based, embeddings-based, and language model-based methods. Statistical and graph based methods are the most popular and the main difference between them is that statistical methods, such as KPMiner (El-Beltagy and Rafea, 2009), RAKE (Rose et al., 2010), and YAKE (Campos et al., 2018), leverage various text statistics to capture keywords, while Graph-based methods, such as TextRank (Mihalcea and Tarau, 2004), Single Rank, KeyCluster (Liu et al., 2009), and RaKUn (Skrlj et al., 2019) build graphs and rank words according to their keyword potential based on their position in the graph. Among the most recent statistical approaches is YAKE (Campos et al., 2018), which we also test in this study. It is based on features such as casing, position, frequency, relatedness to context and dispersion of a specific term, which are heuristically combined to assign a single score to every keyword. KPMiner (El-Beltagy and Rafea, 2009) is an older, simpler method that focuses on the frequency and the position of appearance of a potential keyphrase. In order to enrich the quality of the retrieved phrases, the model proposes several filtering steps, e.g. removing rare candidate phrases that do not appear at least n-times and that do not appear within some cutoff distance from the beginning of the document.

TextRank (Mihalcea and Tarau, 2004), which we evaluate in this study, is one of the first graph-based methods for keyword detection. It leverages Google's PageRank algorithm to rank vertices in the lexical graph according to their importance inside a graph. Other method that employs PageRank is PositionRank (Florescu and Caragea, 2017). The so-called MultiPartiteRank algorithm (Boudin, 2018) encodes the potential candidate keywords of a given document into a multipartite graph structure, which also considers topic information. In this graph two nodes, representing keyphrase candidates, are connected only if they belong to different topics and the edges are weighted according to the distance between the two candidates in the document. In order to rank the verticies, the method leverages PageRank, similarly to Mihalcea and Tarau (2004). One of the most recent graph-based keyword extractors is RaKUn (Skrlj et al., 2019). The main novelty in this algorithm is the expansion of the initial lexical graph with the introduction of meta-vertices, i.e., aggregates of existing vertices. It employs *load centrality* measure for ranking vertices and relies on several graph redundancy filters.

Embedding-based keyword extraction methods are less popular but are nevertheless recently gaining traction. The first methods of this type were proposed by Wang et al. (2015), who proposed Key2Vec (Mahata et al., 2018), and Bennani-Smires et al. (2018), who proposed EmbedRank. Both of these methods employ semantic information from distributed word and sentence representations. The most recent state-of-the-art method of this type is KeyBERT proposed by Grootendorst (2020), which leverages pretrained BERT based embeddings for keyword extraction. In this approach, embedding representations of candidate keyphrases are ranked according to the cosine similarity to the embedding of the entire document.

Language model-based keyword methods, such as the ones proposed by Tomokiyo and Hurst (2003) use language model derived statistics to extract keywords from text. These type of keyword extraction models are quite rare and are not included in our study.

One of the first supervised approaches to keyword extraction was KEA proposed by Witten et al. (1999). It considers keyword identification as a classification task and employs Naive Bayes classifier to determine for each word or phrase in the text if it is a keyword or not. It uses only TF-IDF and the term's position in the text as classification features. A more recent non-neural supervised approach employs a sequence labelling approach to keyword extraction and was proposed by Gollapalli et al. (2017). The approach relies on Conditional Random Field (CRF) tagger. First neural sequence labeling approach was proposed by Luan et al. (2017), who proposed a neural network comprising of a bidirectional Long Short-Term Memory (BiLSTM) layer and a CRF tagging layer.

Keyword detection can also be considered as a sequence-to-sequence generation task. This idea was first proposed by Meng et al. (2017), who employed a recurrent generative model with an attention mechanism and a copying mechanism (Gu et al., 2016) based on positional information for keyword prediction. What distinguishes this model from others is that besides being able to detect keywords in the input text sequence, it can also potentially find keywords that do not appear in the text.

The most recent approaches tackle keyword detection with transformer architectures (Vaswani et al., 2017) and formulate keyword extraction task as a sequence labelling task. In the study by Sahrawat et al. (2020), contextual embeddings generated using BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and GPT-2 (Radford et al., 2019) were fed into a bidirectional Long short-term memory network (BiLSTM) with an optional Conditional random fields layer (BiLSTM-CRF). They conclude that contextual embeddings generated by transformer architectures outperform static. Another study employing transformer architecture and sequence labelling approach was conducted by Martinc et al. (2020). Their approach, named TNT-KID did not rely on massive pretraining but rather on pretraining the transformer based language model on much smaller domain specific corpora. They report good results employing this tactic and claim that this makes their model more transferable to low-resource languages with limited training resources.

Most keyword detection studies still focus on English. Nevertheless, recently several multilingual and cross-lingual studies, which also include low-resource languages, were conducted. One of them is the study by Koloski et al. (2021a) where the performance of two supervised transformer-based models, multilingual BERT with a BiLSTM-CRF classification head and TNT-KID were compared in a multilingual settings, on Estonian, Latvian, Croatian and Russian news corpora. The authors also explored if combining the outputs of the supervised models with the outputs of unsupervised models can improve the recall of the system.

Cross-lingual zero-shot transfer represents an arising hot-topic in the research community. The main idea behind this family of approaches is that models can benefit from transfer from one language to another and therefore be able to conduct tasks in new, 'unseen languages', on which they were not trained in a supervised way. These approaches are especially useful for low-resource languages without manually labeled resources. We are aware of two unsupervised cross-lingual approaches to keyword extraction. One of them is BiKEA (Huang et al., 2014), where the authors construct word graphs for documents in parallel corpora and rely on cross-lingual word statistics for keyword extraction. Another one is the study by Takasu (2010), where the focus is on building single latent space over two languages, and later extracting keywords, to be used as topic categories for the articles, from this common latent space.

Researchers conducted various studies on the effect of applying zero-shot cross-language modeling to multiple domains of NLP, with most of the experiments showing promising results. For example, a zero-shot approach, in which a model was trained on one language and applied on the other, for the task of automatic reading comprehension was carried out by Hsu et al. (2019). Phoneme recognition is another task that cross-lingual zero-shot learning seems to improve. In the work by Xu et al. (2021) they show that cross-lingual phoneme recognition offers performance comparable to the state-of-the-art unsupervised models for the task at hand.

Recently, masked language models based on transformers such as BERT (Devlin et al., 2019) have taken the field by the storm, achieving state-of-the-art results on many tasks. In a study by Wu and Dredze (2019) they explored how well does the multilingual variant of BERT performs when used in a zero-shot setting. The study included 39 languages and covered 5 different tasks, including document classification, natural language inference, named entity recognition, part-of-speech tagging, and dependency parsing. The results were very promising, with the model outscoring several unsupervised and non-transformer based cross-lingual approaches. A zero-shot approach relying on multilingual BERT was also adopted to tackle the tasks of news-sentiment classification (Pelicon et al., 2020), offensive speech detection (Pelicon et al., 2021) and abusive language detection (Glavaš et al., 2020). These studies concluded that pretrained models can be used in a cross-lingual fashion, serving as a strong baseline in the low-resource scenario. To the best of our knowledge, zero-shot transfer has not yet been investigated for the task of keyword extraction.

3. Data

For model evaluation we use six different datasets from the news domain. We include Russian, Croatian, Latvian, and Estonian news article datasets with manually labeled keywords from the Koloski et al. (2021b) dataset repository, using the same splits as in Koloski et al. (2021a). Additionally, we include a benchmark English dataset, the KPTimes dataset (Gallina et al., 2019), and a Slovenian SentiNews (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. Before feeding the datasets to the models, they are lowercased. Each dataset is split into three different splits: *train*, *validation* and *test*. For English, we use the data splits introduced in (Gallina et al., 2019), for other languages besides Slovenian we use the same data splits as in (Koloski et al., 2021a), while for Slovene we first removed the articles without keywords and randomly split the dataset into training, validation and test splits. We use the splits in the following manner:

- *train split* - used for fine-tuning of the cross-lingual supervised model. The procedure is explained in detail in Section 4.3.

- *valid split* - used for early stopping in order to prevent over-fitting during the fine-tuning phase of the supervised models.

	Train			Valid			Test		
Language	size	kw_per_doc	kw_present	size	kw_per_doc	kw_present	size	kw_per_doc	kw_present
Latvian	10506	3.2204	0.8691	2627	3.2687	0.8658	11641	3.1964	0.8624
Estonian	8600	3.8244	0.7809	2150	3.7386	0.7785	7747	4 944	0.8073
Slovenian	4796	4.0052	0.5991	1199	4.1643	0.6054	1519	3.8861	0.5995
Croatian	25778	3.5375	0.7047	6445	3.5469	0.6988	3582	3.5274	0.7009
English	207938	5.324	0.4599	51985	5.0350	0.4583	20000	5 349	0.6205
Russian	11064	5.6377	0.7779	2767.0	5.7311	0.7797	11475.0	5.4261	0.7918

Table 1: Number of documents (size), keywords per document (kw_per_doc) and percentage of keywords present in document's text (kw_present) per split in our experiments. Percentage of present keywords represents the percentage of keywords that appear in the text of the document.

- *test split* - used for evaluation of the supervised and unsupervised methods. This split is not used during training of any of the methods.

The dataset statistics are available in Table 1. For each split we report on the *size* (number of articles), the average amount of keywords per document (*kw_per_doc*) and finally the percentage of keywords that actually appear in the text of the news articles (*kw_present*). *Latvian* dataset has on average least keywords per document (3.22) while the English and Russian datasets contain most keywords per article, 5.32 and 5.64, respectively.

Note that some of the keywords accompanying an article in the data do not appear in the text of the document. For evaluation purposes we only use the **keywords present** in the documents. *English* has the lowest amount of present keywords (46%), while *Latvian* has the highest percentage of present keywords (87%). We consider keyword or keyphrase as present if a stemmed (English and Lativan) or lemmatized version (for other languages) appears in the stemmed or lemmatized version of the document. We use the NLTK's (Bird et al., 2009) implementation of the *PorterStemmer* for English and *LatvianStemmer*[2] for Latvian. For *Croatian, Slovenian, Estonian* and *Russian* we use the *Lemmagen3* (Juršic et al., 2010) lemmatizer.

4. Experimental Setup

In our experiments, we employ several unsupervised models to which we compare several supervised cross-lingual, multilingual and monolingual approaches.

4.1. Unsupervised Approaches

We evaluate three types of unsupervised keyword extraction methods, statistical, graph-based, and embedding-based, described in Section 2.

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.

[2] https://github.com/rihardsk/LatvianStemmer

4.2. 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[3]. 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.

4.2.1. Graph-based Methods

- **TextRank** (Mihalcea and Tarau, 2004): For languages supported by the PKE library (Boudin, 2016) (Russian and English), we employ stemming for normalization, and part-of-speech tagging during candidate weighting. 33% of the highest ranked words are considered as potential candidates.
- **MultipartiteRank** (Boudin, 2018): We employ part-of-speech tagging during candidate weighting for supported languages, and we set the minimum similarity threshold for clustering at 74%.
- **RaKUn** (Skrlj et al., 2019): We use edit distance for calculating distance between nodes, use language specific stopwords from the *stopwords-iso* library[4], a *bigram-count_threshold* of 2 and a *distance_threshold* of 2.

We use the PKE (Boudin, 2016) implementations of *YAKE*, *KPMiner*, *TextRank* and *MultiPartiteRank*. We use the official implementation for the RaKUn model (Skrlj 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

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

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

Section 4.4), we set the number of returned keywords to 10 for all models.

4.3. Supervised Approaches

We utilize the transformer-based BERT model (Devlin et al., 2019) with a token-classification head consisting of a simple linear layer for 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. We do not follow the related work (Koloski et al., 2021a) on adding a BilSTM-CRF classification head on top of BERT for sequence classification. Since the classification head needs to be randomly initialized (i.e. it was not pretrained during the BERT pretraining) and since, among others, we apply the model in a cross-lingual setting, we prefer to keep the token classification head simple, since the layers inside the head do not obtain any multilingual information during fine-tuning. The hypothesis is that using a simple one-layer classification head will result in a better generalization of the model in a cross-lingual setting.

More specifically, we employ the *bert-uncased-multilingual* model from the HuggingFace library (Wolf et al., 2019) and fine-tune it 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.

4.3.1. Cross-lingual Setup

Let C_k be the collection of all of the possible tuples of size k that can be constructed from the 6 languages. For example, C_2 denotes the collection of all possible two language combinations in a set of 6 languages, e.g.

$$C_2 = \{(English, Russian), (English, Latvian), \dots\}$$

We denote the *i-th* tuple of size k with C_k^i, e.g. for the previous example, C_2^1 would yield $(English, Russian)$. The cardinality of the collection C_k, $|C_k|$ is calculated as:

$$|C_k| = \binom{6}{k}$$

We create the *i-th* training dataset D from the collection of tuples C_k of size *k*, as a concatenation of datasets in the tuple, or more formally $D_{i,k}$:

$$D_{i,k} = \bigcup_{language \in C_k^i} \textit{train-split(language)}$$

where *train-split* represents the respective data-split of the given *language* as described in Section 3.

Dependent on the number of languages k included in the training set, and depending on what languages are the trained models employed, we define the following specific settings, for which we report results in Section 5:

- **MON** - monolingual ($k = 1$; $D_{i,1}$ for $i \in [1, |C_1|]$) - where we fine-tune the model on a single language (for example *English*). In this setting we train a total of 6 **monolingual** models[5] and **we train and test each model on the same language**. We use this setting as a baseline to which we compare unsupervised, cross-lingual and multilingual settings, i.e. for cross-lingual (LOO) and unsupervised settings, MON indicates how much we would gain, if language specific training data was available.
- **LOO** - *Leave One Out* ($k = 5$; $D_{i,5}$ for $i \in [1, |C_5|]$) - where we fine-tune the model on a concatenation of five languages (for example *Slovenian, Estonian, Latvian, Russian, Croatian*) and test it on the sixth language not appearing in the train set (e.g. *English*). In this manner we obtain 6 different models. This is the so-called **zero-shot cross-lingual** setting, since we do not include the test language at the training time. The main idea behind this setting is to test how well does a model do if no language specific training data is available. This setting represents the core of our experiments.
- **MUL** - multilingual ($k = 6$; $D_{i,6}$ for $i \in [1, |C_6|]$) - where we fine-tune just one model on all languages from the language set and apply it on all the test datasets. With this experiment we want to check if adding more domain-specific data from other languages improves the performance in comparison to the monolingual setting described above.

4.4. Evaluation Setting

In order to evaluate the models, we calculate F1, recall and precision at 10 retrieved words. We omit the documents that do not have present keywords or do not contain keywords. We do this since we only use approaches that extract words (or multi-word expressions) from the given document and cannot handle keywords not appearing in the text. All approaches are evaluated on the same monolingual test splits, which are not used for training of supervised models. Lowercasing and stemming (for English and Latvian) or lemmatization (for other languages) are performed on both the gold standard and the extracted keywords (keyphrases) during the evaluation.

5. Discussion of Results

Table 2 presents the results in terms of F1@10, Table 3 presents the results in terms of precision@10 and Table 4 presents the results in terms of recall@10.

[5]Note that even in this 'monolingual setting' we employ BERT pretrained on a multilingual corpus, since we are more interested in the comparison of fine-tuning regimes in this paper than in the comparison of different pretrained models.

All unsupervised approaches are outperformed by the cross-lingual approaches (see row LOO) across all of the datasets and according to all criteria. For all languages besides Slovenian, the cross-lingual model improves on the best performing unsupervised model by more than 10 percentage points in terms of F1@10, the improvement being the smallest for Slovenian (about 8 percentage points) and the biggest for Latvian and Estonian (about 15 percentage points). The best performing unsupervised model in terms of F1@10 is KeyBert, which outperforms other unsupervised models on all languages.

The difference in F@10 between the cross-lingual and monolingual models (see row MON) is substantial. If no training data for the target language is used, the performance is more than halved on three languages, Latvian, Estonian and Russian. For the other three languages, the drop is smaller yet still substantial. Similar drops can be observed according to two other measurements, precision@10 and recall@10.

The monolingual and multilingual models offer comparable performance according to all measures and across all languages. This indicates that including other languages into the train set, besides the target language, does generally not improve performance of the models, especially if the training dataset is sufficiently large. This finding supports the so-called curse of multilinguality (Conneau et al., 2019), i.e. a trade-off between the number of languages the model supports and the overall decrease in performance on monolingual and cross-lingual benchmarks. It is however very likely that the transfer between languages would be more successful if the language set would contain more similar languages.

Language		English	Slovenian	Croatian	Latvian	Estonian	Russian
Model	T	F1@10					
Without training data in the target language							
KPMiner	U	0.1584	0.0941	0.1043	0.131	0.0641	0.0578
YAKE	U	0.1449	0.0794	0.1248	0.095	0.0653	0.0966
KeyBert	U	0.1702	0.1153	0.1668	0.1330	0.0923	0.1352
TextRank	G	0.0440	0.0042	0.0041	0.0196	0.0239	0.0392
RaKUn	G	0.1176	0.0875	0.0902	0.0862	0.0605	0.0731
MPRU	G	0.1549	0.0455	0.0683	0.0821	0.0398	0.1171
LOO	C	**0.2856**	**0.2000**	**0.2883**	**0.2844**	**0.2368**	**0.2395**
With training data							
MON	S	0.4658	0.3259	0.4644	**0.6533**	**0.4920**	**0.5979**
MUL	S	**0.4702**	**0.3371**	**0.4674**	0.6532	0.4900	0.5943

Table 2: Performance of the models according to the F1@10. The *T* column denotes the type of model - *U* denotes unsupervised statistical model, *G* denotes unsupervised graph based model, *S* denotes the supervised BERT model and finally *C* denotes the cross-lingual *LOO* model. MPRU entry in the Model column denotes the MultiPartiteRank model.

5.1. Adding More Languages in a Cross-lingual Setting

Above we have showed that adding other languages into the train set already containing the data that matches the target language does generally not improve the performance. On the other hand, here we explore if it is worth adding more languages in a cross-lingual setting. We consider *English* as a testing language, and train on different combinations of languages that do not include English. Figure 1 presents the correlation between the number of languages and the performance of the model according to the F1@10. The Figure does indicate some positive correlation between the number of languages in the train set and the F1@10 improvement. The best was the model trained on Croatian (labeled as C) achieving F1@10 of 35%. Overall, the best performing model on English was trained on the concatenation of the Croatian and Estonian corpus (labeled as CE). Adding additional languages to the train set did not improve the performance further. It does however improve the stability of the models, i.e. the models trained on more languages tend to have higher performance minimum but also lower performance maximum, as can be clearly seen in Figure 2.

Language		English	Slovenian	Croatian	Latvian	Estonian	Russian
Model	T	precision@10					
Without training data in the target language							
KPMiner	U	0.1493	0.1280	0.0974	0.1243	0.0822	0.0578
YAKE	U	0.1068	0.0591	0.0818	0.0602	0.0432	0.0966
KeyBert	U	0.1640	0.1213	0.1428	0.0995	0.0747	0.1352
TextRank	G	0.0322	0.0036	0.0028	0.0120	0.0157	0.0392
RaKUn	G	0.0871	0.0672	0.0605	0.0550	0.0417	0.0731
MPRU	G	0.1151	0.0339	0.0462	0.0524	0.0273	0.1171
LOO	C	**0.3337**	**0.2728**	**0.2955**	**0.3158**	**0.3247**	**0.2395**
With training data							
MON	S	0.5278	0.2954	0.4514	**0.7056**	0.5053	**0.5979**
MUL	S	**0.5318**	**0.3429**	**0.4799**	0.7021	**0.5212**	0.5943

Table 3: Performance of the models according to the precision@10. The *T* column denotes the type of model - *U* denotes unsupervised statistical model, *G* denotes unsupervised graph based model, *S* denotes the supervised BERT model and finally *C* denotes the cross-lingual *LOO* model. MPRU entry in the Model column denotes the MultiPartiteRank model.

Language		English	Slovenian	Croatian	Latvian	Estonian	Russian
Model	T	recall@10					
Without training data in the target language							
KPMiner	U	0.1688	0.0744	0.1123	0.1384	0.0525	0.0578
YAKE	U	0.2251	0.1213	0.2625	0.2254	0.1336	0.0966
KeyBert	U	0.1768	0.1200	0.2001	0.2007	0.1206	0.1352
TextRank	G	0.0694	0.0051	0.0076	0.0536	0.0502	0.0392
RaKUn	G	0.1813	0.1252	0.1772	0.1995	0.1099	0.0731
MPRU	G	0.2367	0.0696	0.1310	0.1899	0.0734	0.1171
LOO	C	**0.2496**	**0.1579**	**0.2815**	**0.2586**	**0.1863**	**0.2395**
With training data							
MON	S	0.4169	**0.3634**	**0.4781**	0.6082	**0.4794**	**0.5979**
MUL	S	**0.4215**	0.3314	0.4556	**0.6107**	0.4624	0.5943

Table 4: Performance of the models according to the recall@10. The *T* column denotes the type of model - *U* denotes unsupervised statistical model, *G* denotes unsupervised graph based model, *S* denotes the supervised BERT model and finally *C* denotes the cross-lingual *LOO* model. MPRU entry in the Model column denotes the MultiPartiteRank model.

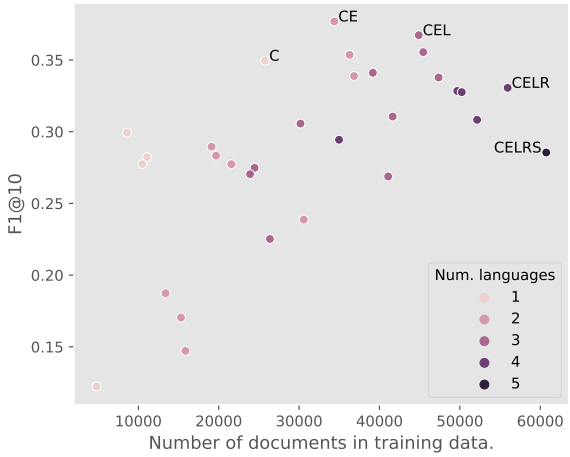

Figure 1: Correlation between the number of languages and the performance of the model according to the F1@10, when the model is tested on an unseen language (English). The best-performing combinations per language are labeled with a sequence of letters representing languages: Croatian - *C*, Slovenian - *S*, Estonian - *E*, Latvian - *L* and Russian - *R*.

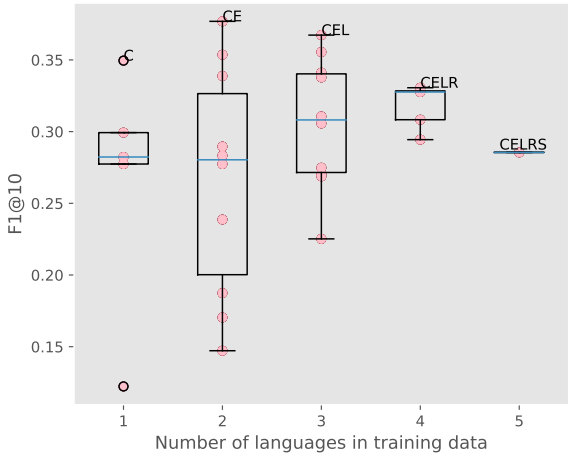

Figure 2: Correlation between the number of languages and the performance of the model according to the F1@10, when the models is tested on an unseen language (English). The models are split into groups according to the number of languages they were trained on and each group is represented by a boxplot. The best-performing combinations per language are labeled with a sequence of letters representing languages: Croatian - *C*, Slovenian - *S*, Estonian - *E*, Latvian - *L* and Russian - *R*..

5.2. Zero-shot Performance of the Monolingual Models

We explored how powerful are the monolingual (**MON**) models described in Section 4.3 in a cross-lingual zero-shot keyword extraction setting. Each of six trained monolingual models was tested on six languages to obtain a heatmap presented in Figure 3.

There was no single monolingual model that worked best for all of the remaining languages. For *English*, the best-performing model was trained on *Croatian*, most likely due to the fact that both datasets contain news from 2019, suggesting some topic intersection. The best performance on the *Estonian* dataset was achieved by the model trained on the *Latvian* dataset, most likely due to the fact that both of these datasets contain news from the same time period and were collected by the same media company, which covers news for both neighboring countries, Estonia and Latvia. Not surprisingly, the reverse correlation si also true: the best-performing model on the *Latvian* dataset was trained on the *Estonian* dataset. The best performance on the *Russian* dataset was achieved by the *Estonian* model due to both of the datasets coming from the same media-house stationed in Estonia, as reported by Koloski et al. (2021a). Finally, the best performance on the *Slovenian* data was achieved by the *Croatian* model, most likely because both of these languages belong to the Southern-Slavic language group and since Slovenia and Croatia are neighbouring countries.

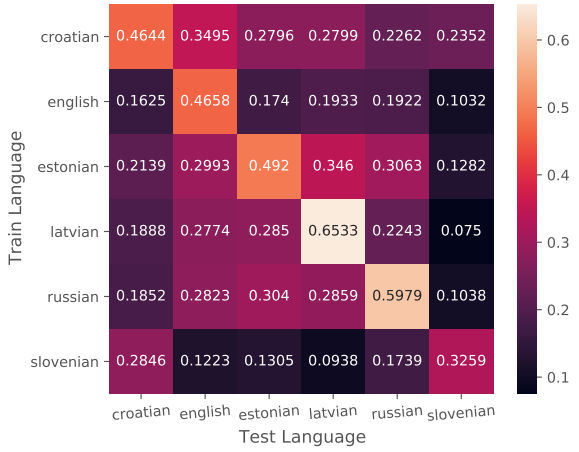

Figure 3: Evaluation of F1-score@10 retrieved keywords of the monolingual models in a setting of zero-shot cross-lingual learning. The rows represent the training language while the columns represent the testing languages.

We also conducted hierarchical clustering, using the cross-lingual scores of the monolingual models as affinities. We present the resulting dendrogram in Figure 4. The results mostly confirm relations between languages, countries and sources of data, pointed out above. *Estonian* and *Latvian* datasets seem to be most similar. *Russian* dataset is the natural addition to this cluster, most likely due to language and content similarity. Interestingly, *Croatian* and *English* form a separate cluster, most likely on the premises of both containing news from 2019, while the Slovenian dataset appears to be most dissimilar to other datasets.

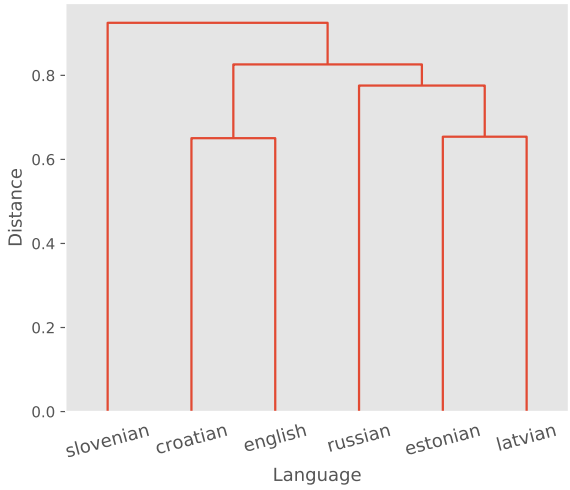

Figure 4: Dendrogram of the agglomerative clustering of the monolingual models applied in a cross-lingual setting.

6. Conclusions and Further Work

In this work, we have presented a comprehensive comparison study covering multiple unsupervised, cross-lingual, multilingual and monolingual approaches for keyword extraction. While we did not manage to improve the performance of the supervised monolingual models by adding additional foreign language data to the training dataset, the results clearly indicate that cross-lingual models outperform unsupervised methods by a large margin. This suggests that if a labeled keyword train set from a specific domain is not available for a specific low-resource language, one opts to try to train a supervised model on a dataset covering the same domain in some other (preferably similar) language and employ that model in a zero-shot setting, before employing the unsupervised methods.

While cross-lingual models tend to outperform unsupervised approaches by a large margin, the discrepancy in performance between the supervised cross-lingual setting and the supervised monolingual setting is nevertheless substantial and training the model on the target languages is still the preferred option in terms of performance. This is in line with further experiments conducted during the study, which suggest that the models perform really well for target languages similar to the languages on which the model was trained and when there is some intersection between the news content in the training and test datasets.

For further work we propose exploring few-shot shot scenarios, in which a small amount of target language data will be added to the multilingual train set. We plan to pinpoint the amount of needed target language data in order to bridge the gap in performance between the monolingual and cross-lingual models. Additionally, we propose ensambling multiple methods and explore how would that benefit the performance of the approach.

7. Acknowledgements

This work was supported by the Slovenian Research Agency (ARRS) grants for the core programme Knowledge technologies (P2-0103), the project Computer-assisted multilingual news discourse analysis with contextual embeddings (CANDAS, J6-2581), as well as the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The third author was financed via young research ARRS grant.

8. Bibliographical References

Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., and Jaggi, M. (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.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Boudin, F. (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.

Boudin, F. (2018). Unsupervised keyphrase extraction with multipartite graphs. *CoRR*, abs/1803.08721.

Bučar, J. (2017). Manually sentiment annotated slovenian news corpus SentiNews 1.0. Slovenian language resource repository CLARIN.SI.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A. M., Nunes, C., and Jatowt, A. (2018). Yake! collection-independent automatic keyword extractor. In *European Conference on Information Retrieval*, pages 806–810. Springer.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

El-Beltagy, S. R. and Rafea, A. (2009). Kp-miner: A keyphrase extraction system for english and arabic documents. *Information systems*, 34(1):132–144.

Florescu, C. and Caragea, C. (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.

Gallina, Y., Boudin, F., and Daille, B. (2019). Kptimes: A large-scale dataset for keyphrase generation on news documents. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 130–135.

Glavaš, G., Karan, M., and Vulić, I. (2020). XHate-999: Analyzing and detecting abusive language across domains and languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6350–6365, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Gollapalli, S. D., Li, X.-L., and Yang, P. (2017). Incorporating expert knowledge into keyphrase extraction. In *Thirty-First AAAI Conference on Artificial Intelligence*, page 3180–3187, San Francisco, California, USA. Association for Computing Machinery.

Grootendorst, M. (2020). Keybert: Minimal keyword extraction with bert.

Gu, J., Lu, Z., Li, H., and Li, V. O. (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.

Hsu, T., Liu, C., and Lee, H. (2019). Zero-shot reading comprehension by cross-lingual transfer learning with multi-lingual language representation model. *CoRR*, abs/1909.09587.

Huang, C.-C., Eskenazi, M., Carbonell, J. G., Ku, L.-W., and Yang, P.-C. (2014). Cross-lingual information to the rescue in keyword extraction. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6.

Juršic, M., Mozetic, I., Erjavec, T., and Lavrac, N. (2010). Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16(9):1190–1214.

Koloski, B., Pollak, S., Škrlj, B., and Martinc, M. (2021a). 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.

Koloski, B., Pollak, S., Škrlj, B., and Martinc, M. (2021b). Keyword extraction datasets for croatian, estonian, latvian and russian 1.0. Slovenian language resource repository CLARIN.SI.

Liu, Z., Li, P., Zheng, Y., and Sun, M. (2009). Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 257–266, Singapore, August. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luan, Y., Ostendorf, M., and Hajishirzi, H. (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.

Mahata, D., Kuriakose, J., Shah, R. R., and Zimmermann, R. (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.

Martinc, M., Škrlj, B., and Pollak, S. (2020). Tnt-kid: Transformer-based neural tagger for keyword identification. *Natural Language Engineering*, pages 1–40.

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., and Chi, Y. (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.

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

Pelicon, A., Pranjić, M., Miljković, D., Škrlj, B., and Pollak, S. (2020). Zero-shot learning for cross-lingual news sentiment classification. *Applied Sciences*, 10(17).

Pelicon, A., Shekhar, R., Škrlj, B., Purver, M., and Pollak, S. (2021). Investigating cross-lingual training for offensive language detection. *PeerJ Computer Science*, 7:e559.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. Technical report, OpenAi.

Reimers, N. and Gurevych, I. (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.

Rose, S., Engel, D., Cramer, N., and Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pages 1–20.

Sahrawat, D., Mahata, D., Kulkarni, M., Zhang, H., Gosangi, R., Stent, A., Sharma, A., Kumar, Y., Shah, R. R., and Zimmermann, R. (2020). Keyphrase ex-

traction 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.

Skrlj, B., Repar, A., and Pollak, S. (2019). Rakun: Rank-based keyword extraction via unsupervised learning and meta vertex aggregation. *CoRR*, abs/1907.06458.

Takasu, A. (2010). Cross-lingual keyword recommendation using latent topics. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '10, page 52–56, New York, NY, USA. Association for Computing Machinery.

Tomokiyo, T. and Hurst, M. (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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, Vancouver, Canada. Curran Associates, Inc.

Wang, R., Liu, W., and McDonald, C. (2015). Corpus-independent generic keyphrase extraction using word embedding vectors. In *Proceedings of the Workshop on Deep Learning for Web Search and Data Mining (DL-WSDM)*, page 39–46, Shanghai, China. Association for Computing Machinery.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., and Nevill-Manning, C. G. (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.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Wu, S. and Dredze, M. (2019). Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. *CoRR*, abs/1904.09077.

Xu, Q., Baevski, A., and Auli, M. (2021). Simple and effective zero-shot cross-lingual phoneme recognition. *CoRR*, abs/2109.11680.