EMBEDDIA

Cross-Lingual Embeddings for Less-Represented Languages in European News Media

Research and Innovation Action
Call: H2020-ICT-2018-1
Call topic: ICT-29-2018 A multilingual Next generation Internet
Project start: 1 January 2019 Project duration: 39 months

D6.10: Final report on EMBEDDIA Assistant Platform evaluation (T6.3)

Executive summary
This deliverable presents the results of EMBEDDIA Media Assistant Platform evaluation. We evaluate the performance of platform's tools and report the results of evaluations done by our media partners. The report contains usability tests and interviews of the media partners after they had tested the tools. The evaluation shows favourable performance of the tools and their positive reception by the users.

Partner in charge: TEXTA

Project co-funded by the European Commission within Horizon 2020 Dissemination Level		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	–
RE	Restricted to a group specified by the Consortium (including the Commission Services)	–
CO	Confidential, only for members of the Consortium (including the Commission Services)	–

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153

Deliverable Information

Document administrative information	
Project acronym:	**EMBEDDIA**
Project number:	**825153**
Deliverable number:	**D6.10**
Deliverable full title:	**Final report on EMBEDDIA Assistant Platform evaluation**
Deliverable short title:	**Final EMA Evaluation**
Document identifier:	**EMBEDDIA-D610-FinalEMAEvaluation-T63-submitted**
Lead partner short name:	**TEXTA**
Report version:	**submitted**
Report submission date:	**31/03/2022**
Dissemination level:	**PU**
Nature:	**R = Report**
Lead author(s):	**Linda Freienthal (TEXTA)**
Co-author(s):	**Birgitta Ojamaa (TEXTA), Ravi Shekhar (QMUL), Hannu Toivonen (UH), Silver Traat (TEXTA) and Matej Martinc (JSI)**
Status:	**__ draft, __ final, _x_ submitted**

The EMBEDDIA Consortium partner responsible for this deliverable has addressed all comments received. Changes to this document are detailed in the change log table below.

Change log

Date	Version number	Author/Editor	Summary of changes made
15/09/2021	v0.1	Linda Freienthal (TEXTA)	Chapter 3.1
01/03/2022	v0.2	Birgitta Ojamaa (TEXTA)	Chapter 2
03/03/2022	v0.3	Hannu Toivonen (UH)	Chapter 3.4
07/03/2022	v0.4	Linda Freienthal (TEXTA)	Introduction
07/03/2022	v0.5	Linda Freienthal (TEXTA)	Chapter 3.2
07/03/2022	v0.6	Silver Traat (TEXTA)	Chapter 3.6
10/03/2022	v0.7	Matthew Purver and Ravi Shekhar (QMUL)	Chapter 3.3
10/03/2022	v0.8	Senja Pollak, Matej Martinc (JSI)	Chapter 3.2
11/03/2022	v1.0	Linda Freienthal (TEXTA)	Conclusion, Executive summary, first version
12/03/2022	v1.1	Marko Robnik-Šikonja (UL)	Internal review.
15/03/2022	v1.2	Linda Freienthal and Birgitta Ojamaa (TEXTA)	Changes after the internal review.
18/03/2022	v1.3	Matthew Purver (QMUL)	Internal review
21/03/2022	v1.4	Linda Freienthal and Birgitta Ojamaa (TEXTA)	Changes after the internal review.
21/03/2022	v1.5	Senja Pollak, Andraž Pelicon (JSI)	Added a new Section 3.3.
21/03/2022	v1.6	Nada Lavrač (JSI)	Quality control.
30/03/2022	v1.7	Linda Freienthal (TEXTA)	Report updated after quality control.
31/03/2022	final	Anita Valmarska and Nada Lavrač (JSI)	Report finalised.
31/03/2022	submitted	Tina Anžič (JSI)	Report submitted.

Table of Contents

List of abbreviations

D	Deliverable
ExM	Ekspress Meedia Group
JSI	Institut Jožef Stefan
NLIB	National Library of Estonia
EMA	EMBEDDIA Media Assistant
SEO	Search Engine Optimisation
TTK	Texta Toolkit
UH	University of Helsinki
WP	Work Package
QMUL	Queen Mary University

1 Introduction

The EMBEDDIA Media Assistant (EMA) platform gathers a selection of tools and resources developed within WP1-WP6 into a joint open source platform. EMA addresses the lack of high-quality state-of-the-art tools for multilingual internet and text processing addressing under-resourced languages in Europe. Focusing on the news media industry, EMA enables journalists, editors, and researchers to search, link, and monitor news reports and editorial content, analyse and react to public user comments, and produce content semi-automatically. The content of EMA's first version has been described in Deliverable D6.7 "EMBEDDIA Media Assistant Platform v1.0 (T6.2)" and the final version is described in Deliverable D6.9 "Final EMBEDDIA Media Assistant Platform, packaged in docker container (T6.2)". EMA consists of four elements:

- Texta Toolkit GUI and API, which allow interactive user access (GUI) and programming access (API) to data exploration, building own classifiers and investigative journalism[1].
- API Wrapper, intended for system integrations, that includes comment and article analyzers[2].
- Demonstrator, showcasing a selection of the developed tools in a simple GUI for demonstration purposes, that includes news generation, comment and article analyzer tools in API Wrapper[3].
- Tools Explorer[4] showcases a larger selection of tools relevant to media industry and research, including the components which were not integrated through other elements.

This deliverable focuses on the evaluation of EMA and is a follow-up to the previous evaluation Deliverable D6.8 "Evaluation of EMBEDDIA Media Assistant Platform v1.0 (T6.3)". The deliverable is structured as follows. In Section 2, we present the results of the performance tests of EMA Tools, followed by the results of user evaluation tests (regarding accuracy and usability) of EMA tools in Section 3. Conclusions are presented in Section 4, followed by a list of associated outputs in Section 5. The three appendices provide additional information on the EMA evaluation.

2 Performance of EMA Tools

In lign with Deliverable D6.8 "Evaluation of EMBEDDIA Media Assistant Platform v1.0" (T6.3), we conducted performance tests to assess the functioning and speed of the tools in the EMA toolkit. The speed of prediction was measured without further analysis of output accuracy, due to the lack of objective reference values for judging quality of responses. Despite the testing datasets being the same, a complete comparison with previous performance results is not possible as some tools have changed. The performance tests were conducted for the following analyzers and generators:

- Article analyzers (see Section 2.1):
  - Keyword Extractor TNT-KID (Estonian)
  - Keyword Extractor RaKUn (Multilingual)
  - Named Entity Extractor TEXTA MLP (Multilingual)
- Comment analyzers (see Section 2.2):
  - Comment Moderator MBERT (Cross-lingual)
  - Comment Moderator FEBERT (English, Estonian)

[1] Texta Toolkit is available for interactive use at `rest.texta.ee`, its backend is at `https://github.com/EMBEDDIA/texta-rest` and the frontend at `https://github.com/EMBEDDIA/texta-rest-front`. Its documentation is available at `docs.texta.ee`.

[2] `https://github.com/EMBEDDIA/embeddia-toolkit`

[3] Demonstrator's repository is available at `https://git.texta.ee/texta/embeddia-demo` and its live version (supported at least until autumn of 2023) at `https://embeddia-demo.texta.ee/`.

[4] Available from EMA landing page `https://embeddia.texta.ee/`

  - Comment Moderator CSEBERT (English, Slovenian, Croatian)
  - Comment Moderator BERT (Estonian)
- Article generators (see Section 2.3):
  - Natural Language Generator (Finnish, Croatian, Russian, Estonian, Slovenian, English)

Testing was performed via the API Wrapper, which accesses models run on three 8-core servers with 16-32 GB of RAM, of which two have 6th generation Intel processors and one has a 7th generation Intel processor. Comment Moderators use GTX 1080 GPU. We describe the results in the subsections below.

2.1 Article Analyzers

The properties of the testing dataset are described in Deliverable D6.8, and the data is presented in Deliverable D4.1 "Datasets, benchmarks and evaluation metrics for cross-lingual content analysis" (T4.4). During our experiments, we accessed the following three news article analyzers via the API Wrapper:

- keyword extractor TNT-KID (Martinc et al., 2021) (Estonian)[5],
- keyword extractor RaKUn (Škrlj et al., 2019) (Multilingual)[6], and
- named entity extractor (Multilingual)[7].

The keyword extractors try to extract the most relevant keywords based on the analyzed text, while the named entity recogniser extracts names of people and organizations.

Similarly to the experiments in D6.8, we created a list of 13 values containing powers of two for all analyzers. The powers were set as follows: $2^n : 2^3 ... 2^{15}$. For each value, we collected from 2-12 articles with lengths closest to these values, altogether 100 articles. The collection was used as the article testing dataset to which we applied each of the EMA tools. After the processing, we calculated average, maximum and minimum tagging times for each of the article lengths. The relation between prediction time and text length is depicted in Figure 1. Similar to the experiments in deliverable D6.8, the line graph shows an approximately linear increase in time as article length increases, as well as maximum and minimum times in transparently coloured areas. Although a linear axis might show the relationship more clearly, the graph is shown in binary logarithmic scale for coherence and consistency with the previous experiments.

We can see that the processing time of all analyzers were similar and relatively low up to the article length of 2^{11}, after which RaKUn seems to be the slowest, Named Entity Recognition in the middle, and TNT-KID being the fastest. Compared to previous experiments, TNT-KID has an improved speed in tagging longer documents, possibly due no longer using lemmatisation.

Tagging a very long article (39 189 characters) caused a timeout error for RaKUn in 1% of total requests. The error occurs when the tools do not respond within 60 seconds and by default the connection is shut down by the server. This error is included in the calculation of averages. In the experiments described in deliverable D6.8, there was one timeout error for TNT-KID and even more errors for RaKUn, therefore the results have slightly improved.

2.2 Comment Analyzers

The properties of the testing dataset are described in Deliverable D6.8. We analyze the performance of the following four comment moderation models, whose function is to detect whether analyzed comments

[5] https://gitlab.com/matej.martinc/tnt_kid
[6] https://github.com/SkBlaz/RaKUn
[7] https://pypi.org/project/texta-mlp/

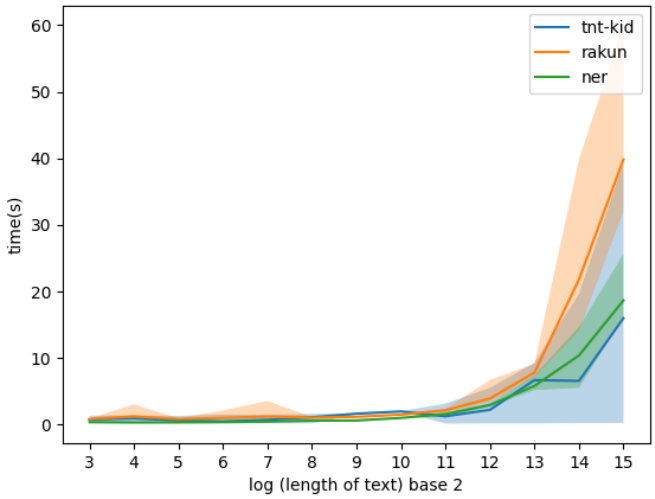

Figure 1: Analysis of article processing time relative to article length (in characters) for the two keyword extractors (TNT-KID and RaKUn), and named entity extractor (NER).

are offensive or not.

- MBERT (Cross-lingual)[8],
- FEBERT (English, Estonian)[9],
- CSEBERT (English, Slovenian, Croatian)[10], and
- Comment Moderator BERT (Estonian)[11].

Similarly to the experiments in D6.8 and in Section 2.1 above, we created a list of ten values containing powers of two for all analyzers. The powers were set as follows: $2^n : 2^3 ... 2^{12}$. For each value, we collected from 2-12 articles with lengths closest to these values, altogether 145 articles. The collection was used as the article testing dataset to which we applied each of the EMA tools. After the processing, we calculated average, maximum and minimum tagging times for each of the article lengths. The relation between prediction time and text length is shown in Figure 2.

[8] https://github.com/EMBEDDIA/comment-filter-mbert-multi
[9] https://github.com/EMBEDDIA/comment-filter-finest-bert-engee
[10] https://github.com/EMBEDDIA/comment-filter-csebert-cse
[11] https://pypi.org/project/texta-bert-tagger/

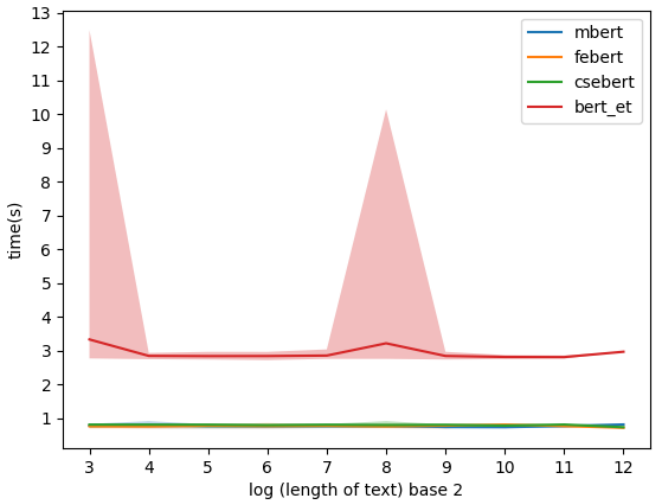

Figure 2: Analysis of comment classification processing time relative to comment length (in characters) for the four tested BERT models. *febert*, *mbert* and *csebert* gave so similar results they are hard to differentiate in the graph.

During the experiment no timeouts occurred and all requests were answered in 12 seconds or less. As shown in the graph, all the multilingual models (MBERT, FEBERT, CSEBERT) were fast (response times under one second) while the monolingual BERT Estonian model was slower, returning results in 3 seconds on average, with maximum up to 12 seconds when dealing with a few shorter-to-average length comments. Compared with the previous experiments there seems to be an increase in speed for multilingual models. During testing we identified unreasonable health checks that slowed down the performance of the Estonian BERT model, so there is an opportunity for optimizing API Wrapper behaviour to improve speed for this model.

2.3 Natural Language Generation

In the article generation analysis the aim is to generate articles given the input data and parameters. The article generator EU NLG[12] creates news articles in six different languages. Three different datasets can be used to choose a more specific topic and thirty eight locations are provided for article generation. Choosing a location parameter means that a unique country-specific article is generated from the data specific to the location country. Different combinations of possible user choices were tested via Wrapper API to generate 352 articles. From the processing, we calculated average, maximum and minimum generation times for each combination of possible parameters and the data.

Generating the articles took from 1.2 seconds to 47.4 seconds, with an average time of 14.6 seconds. As shown in Figure 3, average article generation times for most languages were around 21 seconds. English and Finnish seemed to have faster article generation times, around 10 seconds, but this is due to the different topics and datasets available for these two languages. For topics, using the *health care funding* dataset for article generation was the most efficient time-wise (~2 seconds on average) and Eurostat data (*cphi*) the slowest (~21 seconds on average). *Cphi* data are available for all languages, while *health costs*, and *health funding* are only available for English and Finnish. Therefore, generating articles based on *cphi* data increases average generation times for the rest of the languages.

[12] https://github.com/ljleppan/eu-nlg-prod

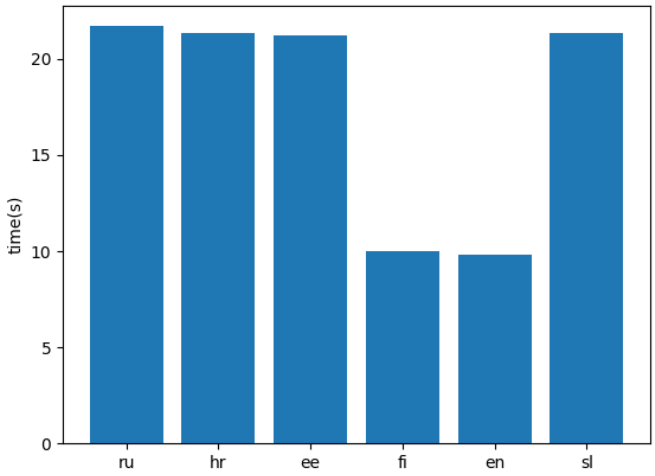

Figure 3: Average article generation times for different languages: Russian (ru), Croatian (hr), Estonian (ee), Finish (fi), English (en), and Slovene (sl).

The location parameter allows the user to choose the country that the article will describe, subject to the availability of data. Out of 38 locations articles were generated fastest for LI (Liechtenstein) and BA (Bosnia and Herzegovina), only available for the *health* datasets. The slowest text generation was for location *all* with an average of 30 seconds, this is expected due to more complexity and availability for all languages. Other locations' averages ranged from 13-20 seconds.

3 Accuracy and usability of EMA Tools

In this section, we present user evaluation tests applied to the EMA tools. In Section 3.1 we focus on Texta Toolkit usability test and a workshop carried out during the development. Sections 3.2, 3.4, and 3.5 introduce the results of user evaluations for keyword extraction tools (Section 3.2), comment analyzer tools (Section 3.4) and natural language generation tools (Section 3.5). EMA usage outside the EMBEDDIA project is then outlined (Section 3.6). In Section 3.7 we present a brief outline of the interviews with our media partners after they had evaluated our tools.

3.1 Texta Toolkit

The usability of EMA's Texta Toolkit (TTK)[13] has been a permanent consideration throughout its development. The EMBEDDIA project organized three events in order to evaluate EMA TTK usability: usability testing, user workshop, and evaluation seminar, described below.

Usability testing. On 26th February 2021 we conducted usability testing. Two usability tests were designed in a way that users are given information about the required task but not how to perform it (i.e. not "click there", but rather "create a time aggregation"). In this way, we investigated how a user without any prior knowledge handles TTK, we could design improvements. The tests (given in Appendix A on page 22) were designed on the Croatian dataset. A Croatian professor of applied linguistics, who studies data science, agreed to try out TTK. The testing was done via video call during which the test user commented her thoughts of what she is trying to find or do and the tester noted down all usability-related issues that occurred.

[13] `docs.texta.ee`

Overall the feedback was positive: "It is going to be a great tool, very useful. Some things need to be added, I am looking forward to this". The test user was interested in TTK, asked if it is open-source, and saw herself using it. TTK proved to be somewhat complicated at first use. The test notes were taken into account by the development team, who made several improvements, including adding more helping texts into TTK. We used the results to make improvements immediately, following the RITE (Rapid Iterative Testing and Evaluation) method (Wixon, 2003).

User workshop. On 21st June 2021 Texta organized **an engaging 3-hour-long workshop on TTK**. Around 20 corpus linguists, computational linguists, NLP-related researchers, and journalists across Europe participated. They first received an overview of previous TTK use cases (for example, TTK was applied to automated comment moderation in media industry web platforms) and how to use it. Afterwards, everyone could test TTK using the Aylien Covid news dataset[14], answering research questions like "What are the higher risk factors of developing a severe case of coronavirus?", "How has China or Wuhan related news frequency changed over time?". While participants agreed that TTK is complex and beginners need to have some experience before using it confidently, the overall feedback was positive.

At the end of the workshop, we asked participants to fill in a questionnaire in Google Forms (see the questionnaire in Appendix B on page 25). Below, we quote some of the feedback received:

- "The Texta Facts tool seems like a very helpful tool, especially given the way it interacts dynamically with the search function. It could help researchers to quickly identify texts written on particular topics."
- "Media and crime analyses would be absolutely brilliant with TEXTA Toolkit. If only I had a "For children" tutorial on the basic tasks that you can perform with the toolkit. Step-by-step walkthroughs are a good way for that, like in the workshop. Obviously, there are a million options with the Toolkit, but getting started is confusing (because there are so many options)."

In the first part of the questionnaire (see Appendix B on page 25), we asked participants to rate their agreement to different statements. These statements are used to calculate the System Usability Scale (SUS) scores (Brooke, 1995). SUS is especially useful for comparing different user groups or versions of the Toolkit. The SUS score is a rating rather then a percentage so it needs additional interpretation. Across many such scores and tests, the average is found to be 68. The score can be converted to an A-F rating (A being the highest and F the lowest grade), where the average response we received was 48 (F). One user rated the tool with the score 82.5 (A) but the rest of answers were below average. This indication of usability issues for the user demographics we addressed with the workshop, led to significant further improvements in the usability aspect, with the focus on better intuitivity for first-time users. Using this initial score as a reference, we intend to compare future improvements quantitatively in post project activities.

Developers' evaluation seminar. On 9th August 2021 Texta organized **an internal six hour long evaluation seminar**, where every tool in TTK was discussed regarding the usability based on the feedback received from clients and experience of the developers. Around 70 issues were identified for the front-end and majority of them are solved by now. Some of the issues were just adding more guiding texts and making error messages clearer. Others were about improving the workflow and creating functionalities that TEXTA's language technologists felt like missing based on their previous TTK using experience, such as autofilling new task parameters based on the previous ones in different tools, overwriting saves searches, adding stop words to significant words aggregation, asking confirmation before retraining a language model or a classificator, adding "tag random doc" option for testing purposes to all taggers.

[14] https://aylien.com/blog/free-coronavirus-news-dataset

3.2 Keyword extractors

Adding keywords for articles helps to find related articles and create statistics of published articles. It is a common practise and the time consuming task often falls on the journalists. Automatic keyword extractors can help with this task either by suggesting suitable keywords or taking the annotation over entirely.

In Deliverable D4.8 "Final evaluation report on cross-lingual content analysis technology (T4.4)" Section 3.1.1, we presented the evaluation by our media partner Ekspress Meedia Group (ExM). From the perspective of ExM, the results of the TNT-KID keyword extractor, combined with an unsupervised TF-IDF based keyword extractor to improve the recall, were satisfactory and the company implemented it into their live product. See Section 3.7 for further information about ExM feedback on EMA tools.

In addition to the evaluation scores from 1 to 5[15] reported in D4.8 Section 3.1.1, we here report precision, recall and F1 score for TNT-KID. We compare the scores on the predicted keywords with the gold standard keywords. As thee gold standard we took the union of the predicted keywords that the human evaluators found suitable and additional non-predicted keywords that the reviewers manually added to each article as missing but needed.

To asses the performance of the model on the evaluation corpus, we followed the same procedure used for the quantitative evaluation of the TNT-KID model (presented in Deliverable D2.6). We report the F1@k score, a harmonic mean between Precision@k and Recall@k, calculated for the first k returned keywords. If the system returns more than k keywords, only the keywords ranked equal to or better than k are considered and the rest are disregarded. Precision@k is the ratio of the number of correct keywords returned by the system divided by the number of all keywords returned by the system:

$$Precision@k = \frac{\#correct \quad returned \quad keywords@k}{\#returned \quad keywords}$$

Recall@k is the ratio of the number of correct keywords returned by the system and ranked equal to or better than k divided by the number of correct ground truth keywords:

$$Recall = \frac{\#correct \quad returned \quad keywords@k}{\#correct \quad keywords}$$

Due to the high variance in the number of ground truth keywords, this type of recall becomes problematic if k is smaller than the number of ground truth keywords, as in this case it becomes impossible for the system to achieve a perfect recall. Similar can happen to precision@k, if the number of keywords in a gold standard is lower than k, and the returned number of keywords is fixed at k.

We formally define F1@k as a harmonic mean between Precision@k and Recall@k:

$$F1@k = 2 * \frac{Precision@k * Recall@k}{Precision@k + Recall@k}$$

All tested instances (generated keywords and gold standard ones) were converted to lower-case and lemmatized using the Lemmagen tool (Juršič et al., 2010)). The evaluation was conducted on 281 documents[16]. TNT-KID obtained Precision@5 of 0.37, Recall@5 of 0.31, F1@5 0.34, Precision@10 of 0.29, Recall@10 of 0.46, and F1@10 of 0.35.

We can compare these results to the previous quantitative evaluation results of the model applied to several media partner datasets (see Deliverable D2.6), in Estonian, Latvian, Russian, and Croatian.

[15] 1 being "the keywords are not relevant to the article at all and don't give proper overview of the content" and 5 "the keywords are relevant to the content and give a proper idea/overview of it"

[16] 20 documents were removed from the original corpus of 301 labeled documents since they contained no gold standard keywords. The evaluation corpus contains duplicated news articles, since some of the documents were evaluated by more than one reviewer, each of them proposing different gold standard keywords. In our evaluation, we treat these distinct reviews of the same document as distinct testing examples.

The results for TNT-KID, TF-IDF and the combination of both for all media datasets are presented in Table 1, as well as additional results with other state-of-the-art approaches (CopyRNN Meng et al. (2019), CatSeqD Yuan et al. (2019), and BERT + BiLSTM-CRF Sahrawat et al. (2020)) for Croatian and Estonian (which cover the primary needs of the EMBEDDIA partners). In that study the results vary between datasets. While TF-IDF alone performs poorly on all datasets besides Croatian, combining TNT-KID and TF-IDF improves recall@5 and recall@10 on all datasets, since this combination mostly returns more keywords than the TNT-KID alone.

If we compare the results of the current study on a new evaluation corpus to results in Table 1, a direct comparison can be done only for the TNT-KID + TF-IDF model on the Estonian corpus, since the language and the model are the same. In terms of F1@5 and F1@10 scores, the results are comparable, but precision and recall show a different picture. While precision is much better in the new evaluation, the recall@10 dropped significantly, from about 0.71 on the Estonian media partner dataset to 0.46. This could be explained by the fact that the new evaluation corpus is significantly newer than the used training corpora.

ExM considered TNT-KID + TF-IDF results meeting their needs, and is now integrating it in their production workflow.

Table 1: Results on the media partner datasets.

Model	P@5	R@5	F1@5	P@10	R@10	F1@10
Croatian						
TF-IDF	0.1518	0.3404	0.2100	0.1289	0.5607	0.2096
TNT-KID	0.3485	0.5359	0.4223	0.3354	0.5594	0.4194
TNT-KID + TF-IDF	0.2793	**0.6517**	0.3911	0.2034	**0.9230**	0.3334
CopyRNN	0.2277	0.3166	0.2418	0.2263	0.3193	0.2409
CatSeqD	0.1580	0.3561	0.2040	0.1389	0.4052	0.1887
BERT + BiLSTM-CRF	**0.4728**	0.4585	**0.4655**	**0.4724**	0.4602	**0.4662**
Estonian						
TF-IDF	0.0377	0.0785	0.0510	0.0388	0.1523	0.0619
TNT-KID	0.5067	0.5649	**0.5343**	0.5055	0.6035	**0.5502**
TNT-KID + TF-IDF	0.2956	**0.5924**	0.3944	0.1864	**0.7061**	0.2949
CopyRNN	0.4706	0.3517	0.3611	0.4703	0.3523	0.3611
CatSeqD	0.3650	0.3910	0.3332	0.3515	0.4037	0.3271
BERT + BiLSTM-CRF	**0.5221**	0.4528	0.4850	**0.5199**	0.4681	0.4927
Russian						
TF-IDF	0.0822	0.1086	0.0936	0.0817	0.1686	0.1101
TNT-KID	**0.6896**	0.5906	**0.6363**	**0.6897**	0.6196	**0.6528**
TNT-KID + TF-IDF	0.4329	**0.6384**	0.5160	0.2932	**0.7468**	0.4211
Latvian						
TF-IDF	0.0518	0.1036	0.0690	0.0419	0.1417	0.0647
TNT-KID	**0.3718**	**0.4120**	**0.3909**	**0.3715**	0.4208	**0.3946**
TNT-KID + TF-IDF	0.1415	0.3417	0.2001	0.1230	**0.5089**	0.1982

3.3 Sentiment analyzer

In D4.7 and D4.8 we evaluated the performance of our classifier on sentiment analysis in Slovene (in cross-validation setting), as well as on Estonian and Croatian in zero-shot learning setting (e.g. without

any training data) on media partners' data. Here, we summarise these results and present the evaluation on additional languages, namely Bosnian, Macedonian and Serbian.

In the same fashion as for Croatian and Estonian, the testing for Bosnian, Macedonian and Serbian was also done in a zero-shot setting e.g. without any additional training in those languages. The news articles were selected by a Slovenian media monitoring company Kliping, a company external to the consortium with whom we are discussing future collaboration. They manually annotated about 200 articles per language, from various topics. The model was evaluated using four standard classification metrics: macro recall, macro precision and macro F1 score. Its performance was compared to a simple majority baseline classifier which classifies all examples into the majority class. The results are presented in Tables 2 and 3. We observe that the results on Bosnian, Macedonian and Serbian languages follow a similar trend than on Croatian and Estonian - our sentiment model outperforms the majority baseline by a significant margin. Moreover, we note that the results for Macedonian and Serbian languages in zero-shot setting are higher than on Slovenian in a standard training setting. We attribute this result to a different sampling method, used for evaluation of usability by Klipping, which produced test sets with a clearer distinction between the sentiment classes which consequently makes the examples easier to classify.

Table 2: Results (from D4.7 and D4.8) on Slovenian (cross-validation) and Croatian and Estonian in zero-shot learning. For each language, the results of the model (column 'sentiment model') are compared to the majority baseline classifier (column 'majority').

	Slovenian		Croatian		Estonian	
	majority	sentiment model	majority	sentiment model	majority	sentiment model
Recall	0.3333	0.6600	0.3300	0.5490	0.3300	0.5400
Precision	0.1734	0.6719	0.2000	0.5632	0.1300	0.7000
F1	0.2276	0.6633	0.2500	0.5477	0.1900	0.5500

Table 3: Results on the test sets provided by Klipping. For each language, the results of the model (column 'sentiment model') are compared to the majority baseline classifier (column 'majority').

	Bosnian		Macedonian		Serbian	
	majority	sentiment model	majority	sentiment model	majority	sentiment model
Recall	0.2500	0.5281	0.3333	0.6822	0.3333	0.7458
Precision	0.0875	0.5586	0.2003	0.6673	0.1150	0.7531
F1	0.1296	0.5333	0.2503	0.6773	0.1710	0.7452

Kliping considers the results of sufficient quality for further usage. In a longer run, JSI will adapt the sytem for target-based sentiment analysis, which would be their core interest.

3.4 Comment analyzers

Due to the high volumes of user comments, media outlets must currently devote significant human resources to moderating comments under their news and removing those which violate media rules. In WP3, we developed automatic comment filtering classifiers designed to help moderators do their job faster and reduce the required human effort. Below, we first present a quantitative evaluation of comment filtering tools via lab experiments on real-world data, followed by an evaluation of the tools when actually used by industry users in a production setting.

3.4.1 Lab-based evaluation

Over the course of WP3 development, we have tested multiple models on real-world data provided by media partners in Croatian (24sata) and Estonian (ExM). Shekhar et al. (2020) developed the first

models, setting a baseline using multilingual BERT (Devlin et al., 2019) and evaluating the performance showing an F1-score circa 63% on the Croatian data. With the introduction of EMBEDDIA trilingual BERT models (Ulčar & Robnik-Šikonja, 2020), we obtained further improved comment moderation performance, increasing to circa 75% F1-score by using other languages' data via cross-lingual training Pelicon et al. (2021). We showed that performance varies between different news sections (where different topics are discussed and different vocabulary is used). By incorporating overall comment semantics as topics, the overall performance further increased, with a boost of c.5% in absolute F1-score (Zosa et al., 2021). This led to the absolute F1-score of c.68% without using a large language model (i.e. compared to the 63% baseline described above). More detail on these evaluations, including details of precision, recall, and F1-score, is given in Deliverable D3.7.

In all these works, we tested the performance of our models on real industry data from media partners; namely, the figures above are for the performance on 24sata's Croatian news comment data. We tuned the performance to obtain an optimal F1 score, as the dataset is highly unbalanced (comments that should be blocked are much more rare than comments that should not be blocked). However, the models could easily be tuned for other objectives specific to the end-users' needs, e.g., different trade-offs between the need for high precision (lower likelihood of false positives, flagging innocent comments as needing to be blocked) and the need for high recall (lower likelihood of false negatives, failing to flag comments that should be blocked).

3.4.2 Real-world evaluation

To understand the effectiveness of the EMBEDDIA comment moderation system in a real industry setting, we worked with the 24sata newspaper (a member of Styria Media Group, and contributor of data to the project via partner Styria/Trikoder) to integrate our classifiers into their production system. The system was put into use in the real day-to-day work of 24sata's moderators in December 2021 and used for close to eight weeks. At the end of this period, we analysed the quantitative performance of the system compared to the final decisions of the moderators; and interviewed two of the moderators to get qualitative user feedback. The interview was unstructured, focusing on three main aspects:

1. How do the moderators perform moderation, when using the produced system and without it?
2. Does the EMBEDDIA comment moderation system help?
3. How could the system be further improved?

Below we present the main findings.

Quantitative evaluation: During the test period while moderators used the EMBEDDIA model, 24sata recorded the model output and the annotators' decision and provided us with this data for more than 527K comments. Taking the annotators' decisions as the gold standard, the EMBEDDIA model had overall macro-averaged F1 Score of 49.3%. Since moderators are interested in finding all possible comments to be blocked, we also report recall and precision for blocked comments, as 55.6% and 7.4% respectively. This is a drop from the results obtained on our previous test dataset, where we achieved F1 scores from 54-62%, depending on the year from which the data was taken (Shekhar et al., 2020). On further analysis of the data, we found that the main reason for the comparatively low accuracy was a significant shift in topics discussed in comments. Specifically, we trained our model on comment data up until 2019, while the evaluation was performed from December 2021 to February 2022. The majority of the blocked comments during the evaluation were related to "Vaccination' and the "Russian-Ukraine war". Of course, no comments related to these topics were present in the training dataset, and problematic comments on these topics were therefore hard for the model to recognise. To incorporate these new topics into the model, we re-trained the EMBEDDIA model on part of the new data, improving performance by 3.5% (49.3% to 52.9%) F1 score. This suggests that while topic drift in news can cause significant drops in performance, periodic fine-tuning on the latest data allows the model to adapt to the current situation.

Moderation processes: Moderators generally work in shifts and perform the moderation live as the comments are posted. The volume of comments varies from 700 to 2000 comments in an hour. Specifically, they have more comments at the start of the morning shift, when there is some breaking news or a big event like a football match. During these times, they might have 2000 to 3000 comments per hour. However, due to the volume of the comments, they have to operate very fast, and at the same time, they learn to expect the types of comments they would get based on the article. In most cases, to decide, they only read a comment without the context, and here their experience plays a significant role. They used the EMBEDDIA system, which flags whole comments as potentially needing moderation, to prioritise which comments to examine first. They also use an internal list of banned words for the decision and keep adding new words to this list when they encounter new bad words. Finally, they pay careful attention to the comments that violates "major" policy rules, because this results in users being blocked from the site (violating only "minor" rules results only in comments being blocked).

Overall impression: Before the interview, the moderators internally discussed the working process and came prepared with the overall impression of the system. Overall, they liked the system and felt that it made them more efficient in performing their job. They found that the system was most effective when they had a large volume of comments to deal with. Initially, they found it challenging to use the system, especially when the system made mistakes. However, after using the system for some time, they became more used to it and learned how to use the output better, including being able to anticipate the types of errors made by the model. Overall they rated it 3 out of 5 and want to keep using the system. We expect that their impression will improve once the main issues are fixed (see below).

Rule variability and mismatch: They found that the model performs better for some rules (24sata rule numbers 6, 7, and 8) than the other rules. They also noticed that the model often correctly flags comments as needing to be blocked, but assigns the rule number wrongly. During the interview, we realized that this is due to a mismatch between the numbering systems being used: there is a difference in the rule number used internally by the moderators and the numbers provided to us and used to develop the system. (Specifically, our rules 1, 2, 3, and 4 were 24sata's internal rules 5, 1, 2, and 3, respectively). Apart from our rule 1, the others are the "major" rules which lead to user blocking, and moderators pointed out that this need modification. Fortunately, this mismatch was easy to fix once identified, and we have now fixed this in the next version of the system. We note that even with this issue, the moderators were positive about the system and wanted to keep using it (see above); clearly this will only improve.

Possible improvements: One of the advantages of the EMBEDDIA model is that it could provide *decision confidence*. Therefore, we ask moderators whether they think having classifier confidence would be helpful in their moderation process (e.g., by showing flagged comments with varying degrees of shading in their interface). According to them, having this information might not be beneficial and might take more processing time. However, they were open to try before making the final decision.

Another possible improvement suggested is based on the fact that the classifier is easily tunable based on the end-user requirement of *precision or recall*. The moderators thought it would be better to see more potentially problematic comments (i.e. having higher recall) than to only see highly accurate decisions (higher precision). In this way, they would focus only on the comments selected by the EMBEDDIA system for the blocking decision and save their time. We intend to tune the EMBEDDIA model for higher recall in future releases.

One of the significant improvements they suggested is incorporating *user feedback* from the moderators like adding new bad words and adding evolving local context. Commentators find new ways to bypass the moderation, and it would be good to incorporate those into the model. Another suggestion was that instead of assigning the rule, the system only suggests whether a comment should be blocked or not, and the moderators do rule assignment manually. We believe this is due to the mismatch in rules described above (and now resolved), so will wait for further testing to see if this issue is still raised. One moderator indicated that comments marked as violating a major rule by the EMBEDDIA system shall

not be published until checked by a moderator. This will allow less nasty comments in the feed. Some of these suggestions could be incorporated using the active learning techniques we are investigating and will be included in the next version.

Overall, moderators were positive about the system and wanted to keep using it. They have continued to use it since that test, and a range of improvements is currently being integrated into the next version for further testing.

3.5 Natural Language Generator

The final evaluation report on the multilingual text generation technology is contained in Deliverable D5.7: we refer the interested reader to that deliverable. Here, we summarize it briefly.

In Deliverable D5.7, we evaluate the three main natural language generation (NLG) components of EMBEDDIA: the multilingual natural language generation method (from Task T5.1), the document planning and content selection methods (T5.2), and the headline generation method (T5.3). The evaluations are based on a combination of qualitative and quantitative methods and analysis of the software. The results indicate that the fundamental technology developed for natural language generation is sound and fits the design goals, and that journalists find the results useful. The approaches developed for document planning and content selection support a range of different use cases. For headline generation, our results show that pre-trained multilingual NLG models are a good choice for low-resourced languages.

3.6 Evaluation outside the EMBEDDIA project

During the course of the EMBEDDIA project, the National Library of Estonia (NLIB) announced two tenders for solving automatic subject indexing (i.e. keyword tagging in the library terms) for books, booklets, articles and other kind of text pieces.

In 2019-2020, partner TEXTA won **the first tender** "Automaatse märksõnastamise KRATT: detailanalüüs, sh prototüübi loomine"[17] for creating a prototype for the automatic subject indexer. TEXTA used Hybrid Tagger (Vaik et al., 2020), which is introduced as an associated output in Deliverable D6.8 and received promising results with it. Further, Asula et al. (2021) describe the work done and the results of the tender, including the evaluation on methods using Hybrid Tagger. The preprint version can be found in Appendix C and at ArXiv[18].

In D4.8 Section 3.1.2, we mentioned **the second tender** "Kratt "Automaatne artiklite märksõnastamine""[19] also won and carried out by TEXTA in 2021-2022. This follow-up tender focused on developing different methods for article tagging in Estonian and evaluation of the results with both regular library users and the library's cataloguers. By then, TEXTA testing with RaKUn revealed that with appropriate preprocessing (lemmatization) and hyperparameters the results were good enough with Estonian texts for production and had added RaKUn to Texta Toolkit.

In the second tender, seven different methods were tested, including some that used either Hybrid Tagger or RaKUn (with different pre- and postprocessing) developed within the EMBEDDIA project. As stated in D4.8 "the evaluation showed that RaKUn was the best method out of all the methods tested out in this tender". However, methods using Hybrid Tagger were not far behind. Figure 4 illustrates the evaluation architecture of the seven tested methods. Although the results of the tender, including the detailed analysis and evaluation of the methods are not yet publicly accessible as the tender ends at the end of March 2022, we can already specify the ranking of the tested methods:

1. RaKUn (M3)

[17] https://riigihanked.riik.ee/rhr-web/#/procurement/1597549/general-info
[18] https://arxiv.org/abs/2203.12998
[19] https://riigihanked.riik.ee/rhr-web/#/procurement/3224632/documents?group=B

2. Hybrid Tagger (M2)
3. RaKUn (M5)
4. RaKUn (M4)
5. Phrase detection (M6)
6. Phrase detection (M7)
7. Hybrid Tagger (M1)

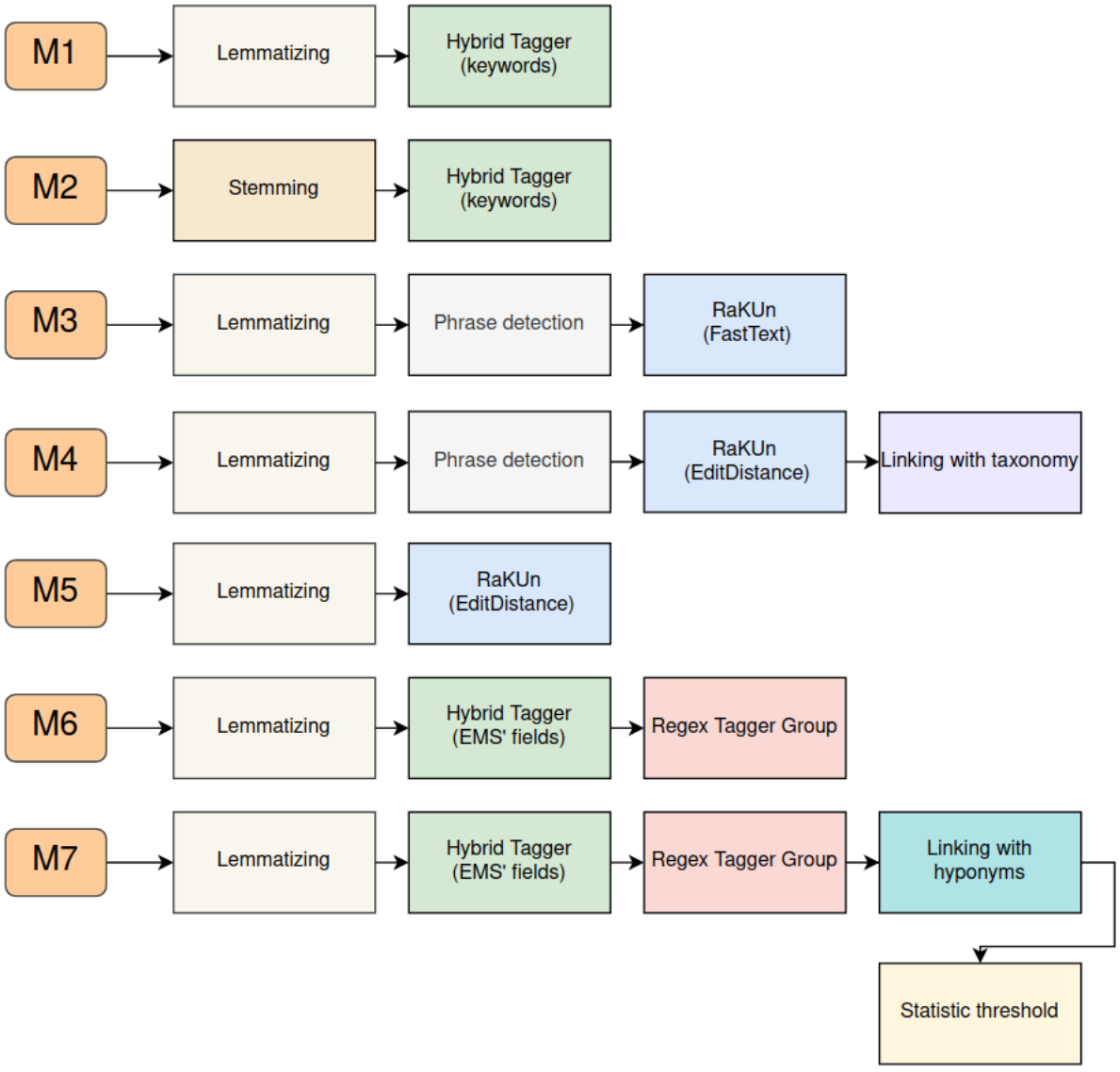

Figure 4: The evaluation pipelines of seven keyword extraction methods tested in the NLIB tender for the Estonian language.

NLIB published the tool developed within the second tender in `https://marta.nlib.ee/`. The tool is publicly accesible and processes Estonian texts.

3.7 Interviews with media partners

As part of the EMA evaluation (and in collaboration with Task T7.3 on exploitation), we conducted interviews with representatives from EMBEDDIA media partners STT (Finland), Ekspress Meedia Group (ExM, Estonia) and 24sata (Styria, Croatia). The three interviewees (one from each company) were from the business divisions of the partners and the goal was to introduce them the tools created in the EMBEDDIA project and discuss their usefulness from the business perspective. All interviewed companies are working with comments. STT provides comment moderation as a service while the other two handle their own user comments. A common problem with comments is the quantity of unwanted online text (hate speech, offensive language, etc). All participants agreed that the user comment moderation API built in EMBEDDIA could be useful for them so that they would gain speed in moderation, catch more malicious comments and increase the quality of the online conversations.

Meta tagging of articles is also done by all three partner companies. STT and 24sata do it manually while ExM has a semiautomatic solution that is not producing satisfactory results. Common problems of the current manual approach are that taxonomies of keywords grow too large or existing systems are not flexible enough and need too many user clicks. In this sense, the keyword extraction tools built in the project were found useful by 24sata and ExM. STT found it useful but not compatible with their existing solution which is old and does not allow API connections. With the help of automatic meta tagging, the media houses could get better reader engagement, have better search engine optimisation (SEO) and can find articles faster. They would also have better control over the taxonomies and journalists would be able to find/add the relevant tags faster.

All participants had heard of the produced natural language generation tool but only STT had tried to find its business value. It found that the tool could be used for generating business news, sports news or covering elections. The question is how easy the tools could be adapted to new languages and new topics.

The analysis of results is also presented in deliverable D7.5 (as Section 5.4 of deliverable PEDR-3: Final Dissemination Report and Plan for Results Exploitation), including also the actual answers to the interview questions in Appendix B of deliverable D7.5.

4 Conclusion

The qualitative and quantitative evaluation of EMA showed that media partners found the tools useful, implemented them to their systems, and are using them in their day-to-day production. Keyword tagging tools Hybrid Tagger and RaKUn have been tested and deployed in the National Library of Estonia, giving good results.

During the EMA development, we conducted usability tests and a workshop to find out how to make the TTK more user-friendly. We used that information in further development. We conducted performance tests on tools available through the API Wrapper to check how the updates in the second half of the project have influenced the speed and found it satisfactory. We also present recent results on the usability and user evaluations. Finally, we summarized the interviews with our media partners from the business perspective after they had tested our tools in their production.

Overall, we can consider EMA successful, achieving its technical and business objectives. Following the requests for information we received, many media companies will be able to test EMA and Tools Explorer at `embeddia.texta.ee`.

5 Associated Outputs

Citation	Status	Appendix
Asula, M., Makke, J., Freienthal, L., Kuulmets, H.-A., & Sirel, R. (2021). Kratt: Developing an automatic subject indexing tool for the National Library of Estonia. In *Cataloging & Classification Quarterly , 59(8)*, 775-793.	Published	Appendix C

References

Asula, M., Makke, J., Freienthal, L., Kuulmets, H.-A., & Sirel, R. (2021). Kratt: Developing an automatic subject indexing tool for the national library of estonia. *Cataloging & Classification Quarterly*, *59*(8), 775-793. doi: 10.1080/01639374.2021.1998283

Brooke, J. (1995, 11). Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, *189*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186).

Juršič, M., Mozetič, I., Erjavec, T., & Lavrač, N. (2010). Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, *16*(9), 1190–1214.

Martinc, M., Škrlj, B., & Pollak, S. (2021). TNT-KID: Transformer-based neural tagger for keyword identification. *Natural Language Engineering*, 1–40. doi: 10.1017/S1351324921000127

Meng, R., Yuan, X., Wang, T., Brusilovsky, P., Trischler, A., & He, D. (2019). Does order matter? An empirical study on generating multiple keyphrases as a sequence. *arXiv preprint arXiv:1909.03590*.

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

Sahrawat, D., Mahata, D., Kulkarni, M., Zhang, H., Gosangi, R., Stent, A., ... Zimmermann, R. (2020). Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings. In *Proceedings of European Conference on Information Retrieval (ECIR 2020)* (pp. 328–335).

Shekhar, R., Pranjić, M., Pollak, S., Pelicon, A., & Purver, M. (2020). Automating news comment moderation with limited resources: Benchmarking in Croatian and Estonian. *Journal of Language Technology and Computational Linguistics*, *to appear*.

Ulčar, M., & Robnik-Šikonja, M. (2020). FinEst BERT and CroSloEngual BERT: less is more in multilingual models. In *Proceedings of Text, Speech, and Dialogue, TSD 2020.*

Vaik, K., Asula, M., & Sirel, R. (2020). *Hybrid Tagger – An Industry-driven Solution for Extreme Multilabel Text Classification.* Zenodo preprint. doi: 10.5281/zenodo.4306169

Wixon, D. (2003). Evaluating usability methods: why the current literature fails the practitioner. *Interactions*, *10*(4).

Yuan, X., Wang, T., Meng, R., Thaker, K., Brusilovsky, P., He, D., & Trischler, A. (2019). One size does not fit all: Generating and evaluating variable number of keyphrases. *arXiv preprint arXiv:1810.05241*.

Zosa, E., Shekhar, R., Karan, M., & Purver, M. (2021, September). Not all comments are equal: Insights into comment moderation from a topic-aware model. In *Proceedings of the 13th biennial International Conference Recent Advances in Natural Language Processing (RANLP).*

Škrlj, B., Repar, A., & Pollak, S. (2019). RaKUn: Rank-based keyword extraction via unsupervised learning and meta vertex aggregation. *Lecture Notes in Computer Science*, 311–323. doi: 10.1007/978-3-030-31372-2_26

Appendix A: TTK usability tests

Test 1. Searching and aggregating data

You are a journalist at a media company in Croatia. Your boss comes to you and asks: "What do you think, have people been mentioning less and less the president Kolinda Grabar-Kitarović in our comments?". You decide to check it up before answering on your gut feeling. Furthermore - you'll check if the comments mentioning her are mainly suitable comments or the ones that have been deleted by the moderators since it is violating rules of good commenting (is too rude, offensive etc.)

Use TEXTA Toolkit to gain knowledge about the following questions:

a) How has the frequency of mentioning the president Kolinda Grabar-Kitarović in the comments changed in time?
b) Are those comments mostly hate speech or consist of threats, bad language, insults that let us believe the attitude towards her is negative?
c) Has the attitude change over time? If yes, how?

As an experienced TEXTA Toolkit user, you already know that:

- All these questions can be answered via Search in TEXTA Toolkit which currently works best in Chrome at *rest.texta.ee*.
- Your user *Usability_testing* (password: *ITesTUsabiliTy*) has access to a project with Croatian comments data index *embeddia_styria_comments* in it.
- This index has many fields, including:
    - *content*, where you can search for the mentionings of the president in several different variations,
    - *created_date* on which you can use the Aggregations panel to visualize the changes of frequency of given (searched out) comments,
    - *texta_facts*, which have two facts:
        - no_violation, with values:
            - no_violation (comment is not problematic in the eyes of the moderatiors)
        - rule_violated, with values:
            - minor 1
                - (original): "Oglašavanje, netematski sadržaj, spam, kršenje autorskih prava, citiranje uvredljivih komentara ili bilo kakvih drugih komentara koji nisu dopušteni na portalu".
                - (translated): "Advertising, content unrelated to the topic, spam, copyright infringement, citation of abusive comments or any other comments that are not allowed on the portal")
            - major 2
                - (original): "Izravno prijetiti korisnicima, novinarima, administratorima ili subjektima članaka, koje mogu rezultirati i kaznenim progonom"

    - (translated): "Direct threats to other users, journalists, admins or subjects of articles, which may also result in criminal prosecution”
- major 3
    - (original): "Vrijeđanje, omalovažavanje i napad na temelju nacionalne, rasne, spolne ili vjerske pripadnosti, govor mržnje te propagiranje nasilja"
    - (translated): "Verbal abuse, derogation and verbal attack based on national, racial, sexual or religious affiliation, hate speech and incitement"
- major 4
    - (original): "Prikupljanje i objavljivanje osobnih podataka, upload, distribucija ili objava sadržaja pornografskog, obscenog, nametljivog ili nezakonitog karaktera te korištenje prostačkog ili uvredljivog nicka te nicka u kojem je sadržano ime i prezime drugih osoba"
    - (translated): "Collecting and publishing personal information, uploading, distributing or publishing pornographic, obscene, immoral or illegal content and using a vulgar or offensive nickname that contains the name and surname of others"
- minor 5
    - (original): "Objavljivanje lažnih informacija s ciljem zavaravanja ili klevete, te "trollanje" - namjerno provociranje drugih komentatora" Description in
    - (translated): "Publishing false information for the purpose of deception or slander, and "trolling" - deliberately provoking other commentators"
- minor 6
    - (original): "Upotreba psovki, osim u slučaju kada se koriste kao stilski izraz, odnosno nisu nekome direktno upućene"
    - (translated): "Use of bad language, unless they are used as a stylistic expression, or are not addressed directly to someone"
- minor 7
    - (original): "Pisanje bilo kojim drugim jezikom osim hrvatskog ili pismom osim latinice i pisanje isključivo velikim slovima"
    - (translated): "Writing in other language besides the Croatian, in other scripts besides Latin or writing with all caps"
- minor 8
    - (original): "Vrijeđanje ostalih korisnika i njihovih komentara, autora članaka, te izravnih ili neizravnih

subjekta članaka te prozivanje administratora ili polemiziranje s administratorom na bilo koji način"
- (translated): "Verbaly abusing of other users and their comments, article authors, and direct or indirect article subjects, calling the admins out or arguing with them in any way"

Test 2. Training a classifier

You are a technical helper in a news media company in Croatia. One of the mundane tasks the journalists have to do is to tag their articles with meaningful tags which helps to find related articles. This is a demanding task, though: journalists are under time pressure and are producing several articles a day without paying much attention to choosing keywords on a tag set that is growing rapidly with new tags and different versions of the same thing (capital of England and London are two different tags). You decide to help with that and create a helper that will suggest suitable tags for the journalists, eliminating the variety of words and loss of time.

You already know that this kind of helper can be done with the help of the TEXTA Toolkit. You can train several tag recognisers (called Taggers). You can train them together when the articles are already tagged and one by one based on a search result or the tags. You are not yet sure if the tags already given to the articles need to be cleaned before training the helper. That is why you decide to create your own subset of positive samples for a certain tag.

Create a Tagger (under Models > Tagger) for tagging documents mentioning Eurovision in the datset *embeddia_styria_articles_lemmatized*. Test it out with Tag random doc or other options under Actions.

As a TEXTA Toolkit expert, you already know that you can create a subset of documents via Search (by searching the documents containing mentionings of European Union and saving it) and use it in training the Tagger.

Appendix B: TTK workshop feedback questions

TEXTA Toolkit Feedback Form

1. Usability Questions (mandatory)

Users had to choose „Strongly Disagree", „Disagree", „Neutral", „Agree" and „Strongly Agree" to the following statements:

* I think that I would like to use this system frequently
* I found the system unnecessarily complex.
* I thought the system was easy to use.
* I think that I would need the support of a technical person to be able to use this system.
* I found the various functions in this system were well integrated.
* I thought there was too much inconsistency in this system.
* I would imagine that most people would learn to use this system very quickly.
* I found the system very cumbersome to use.
* I felt very confident using the system.
* I needed to learn a lot of things before I could get going with this system.

2. Open-text questions (optional)

* What is your domain of expertise?
* Are there any tasks or scenarios in your work where you would choose to use this tool? (please describe the tasks if applicable)
* Are there any features that you find particularly useful?
* Are there any features that you find particularly difficult to use?
* Are there any key features missing that would be useful for your work?
* Does this tool provide something not available in tools you have previously used?
* Do you have any additional feedback or comments?

Appendix C: Kratt: Developing an Automatic Subject Indexing Tool for The National Library of Estonia

Kratt: Developing an Automatic Subject Indexing Tool for The National Library of Estonia

Marit Asula, Jane Makke, Linda Freienthal, Hele-Andra Kuulmets & Raul Sirel

28 May 2021

This is a preprint version of Marit Asula, Jane Makke, Linda Freienthal, Hele-Andra Kuulmets & Raul Sirel (2021) Kratt: Developing an Automatic Subject Indexing Tool for the National Library of Estonia, Cataloging & Classification Quarterly, 59:8, 775-793, DOI: 10.1080/01639374.2021.1998283

Abstract

Manual subject indexing in libraries is a time-consuming and costly process and the quality of the assigned subjects is affected by the cataloguer's knowledge on the specific topics contained in the book. Trying to solve these issues, we exploited the opportunities arising from artificial intelligence to develop Kratt: a prototype of an automatic subject indexing tool. Kratt is able to subject index a book independent of its extent and genre with a set of keywords present in the Estonian Subject Thesaurus. It takes Kratt approximately 1 minute to subject index a book, outperforming humans 10-15 times. Although the resulting keywords were not considered satisfactory by the cataloguers, the ratings of a small sample of regular library users showed more promise. We also argue that the results can be enhanced by including a bigger corpus for training the model and applying more careful preprocessing techniques.

1 Keywords

national libraries, automated subject indexing, machine learning, natural language processing, cataloguing

2 Introduction

As a national bibliographic agency, the National Library of Estonia is responsible for the registration of the publications issued in Estonia or outside Estonia by Estonians. Bibliographic descriptions are primarily definitive containing all the mandatory elements set out in the ISBD. Normally, the cataloguing process also includes the subject indexing for which Estonian Subject Thesaurus (Eesti Märksõnastik, EMS)[1] is used along with UDC Summary for classifying the resources. EMS was launched in 2009, it currently contains about 61 000 terms, among which there are approximately 40 000 preferred terms and 21 000 non-preferred terms. The terms in EMS are in Estonian, however, an English translation is given for each term.

According to some authors, the cataloguing process is considered to be time-consuming and expensive in the library work[2]: cataloguing is manual work and it is estimated that a cataloger is able to describe around 3-4 books per hour[3]. At the National Library of Estonia, it is estimated that the cataloguer should produce 10 national bibliographic records[4] in a day depending on the material type.

Subject access data is usually created by a cataloguer who discovers the topics of the book and performs a content analysis. Eventually, keywords expressing the subject of the resource will be generated. It is estimated that the subject indexing and classifying of resources takes approximately 15 minutes depending on the material, topic and the level of experience of the librarian. Indeed, one of the problems with subject description performed by catalogers is the limited ability to understand and describe the subject. The quality of description depends on the intellectual assumptions of a cataloger and is affected by some amount of subjectivity[5]. Pokorny argues: "When processing scientific books, a cataloger who is not an expert in a given discipline is not able to correctly and precisely understand and describe the topics contained in the book."[6]

Wishing to address the above-mentioned issues (speed, subjectivity, cost and quality), the National Library of Estonia decided to test automating the process of subject indexing with the help of machine learning and text mining tools. Since Estonian Government is actively supporting the development of artificial intelligence[7] to reduce costs, raise quality and save time in the public sector, the Library found financial support for the project from the government programs. It also sought help from the private sector to find relevant knowledge and experience.

The project was initiated in 2019 and it took 6-7 months to build a prototype of the Kratt[8] (it is a common name for AI applications in Estonia) for automated subject indexing of books in Estonian language. Library's goal was to test: (a) whether the automation of the process would be possible, (b) if it might save time and money, (c) if Kratt could help to raise the quality, and (d) if Kratt could be integrated into the daily cataloguing workflows.

3 Related work

Several national libraries have over the years reported their attempts to exploit text mining and machine learning methods in order to automate cataloguing and indexing tasks and reduce the amount of human workload needed.

In general, there are two options for libraries to approach the issue. Either they build their own solution from scratch or purchase the software from the market. The later option for example has been used by Deutsche Nationalbibliothek[9]. The subject indexing tool that they evaluated uses unsupervised methods to extract terms and later matches them to the controlled vocabulary. The results of their experiments, which focused on online publications (mostly doctoral theses) were not considered satisfactory, mainly because of the low precision of assigned subjects.

A different approach was chosen by the National Library of Finland who developed its own tool for subject indexing and text classification.[10] The tool, called Annif, is built on top of existing open-source algorithms, allowing users to choose from multiple unsupervised and supervised algorithms, including ensemble methods. Their experiments, conducted on more or less academic articles, Master's and Doctoral theses, question-answer pairs of any topic and a regional newspaper, showed that ensemble methods perform better than individual methods. Annif performed best on theses with an average f1-score of 0.46 and worst on newspaper articles with an average f1-score of 0.28. It is also reported that in The University of Jyväskylä, where Annif was adapted, approximately one half of the subjects suggested were selected as final subjects by students who were uploading their Master's theses to the repository. Librarians who were reviewing uploads selected 53% of the same suggestions.

4 Description of the Training Data

The National Library of Estonia (NLE) provided 7668 publically available books with corresponding subject indices for developing the prototype. All the books used for developing the prototype are available in NLE's digital archive DIGAR[11] and the corresponding subject indices in the Estonian National Bibliography (ERB)[12]. The books were in various languages and consisted of a wide variety of forms, including dissertations, reports, brochures, manuals, textbooks, collections of articles, transcripts, dictionaries, novels, short stories etc. Each book had been subject indexed by a professional cataloguer and the total number of unique preferred terms (also referred to as "subject indices", "labels" or "keywords" in the subsequent text) was 10 098. Each preferred term belonged to one of the seven major categories: genre and form (e.g. "fiction", "memorial"), time (e.g. "21st century", "2020"), location (e.g. "Latvia", "London"), topic (e.g. "sewing", "economy"), person ("Barack Obama", "Herman Hesse"), collective/organization (e.g. "The European Union", "The University of Tartu"), and temporary collective or event (e.g. "The European Capital of Culture", "Black Nights Film Festival"). As categories "person", "collective", "temporary collective or event" are not based on EMS or any other thesaurus and would have needed additional preprocessing, e.g. merging differently written preferred terms referring to the same entity, we excluded the labels belonging to these categories from the prototype. After removing the aforementioned keywords, the total number of labels was reduced to 8928. As EMS contains about 40 0000 preferred terms, it reveals the first shortcoming of the training data: only 22% of all the possible labels were included. This means that supervised machine learning methods would be able to predict only the same subset of labels excluding 78% of all the possible labels.

Furthermore, 8928 labels might constitute only one fifth of all the preferred terms in EMS, but it is still a very large set of possible targets to consider while constructing the keyword assignment models. Another difficulty arising from the data was a very sparse distribution of unique labels: the median frequency of unique labels was 2, with most of the labels occurring only 1-4 times in the whole set (Figure 1). This is an extremely small number considering the fact that most machine learning-based classification methods require hundreds of examples to accurately learn the features corresponding to each label. If we set the minimum number of required examples to 50 - a relatively low bar to cross, only 111 labels would have exceeded the threshold.

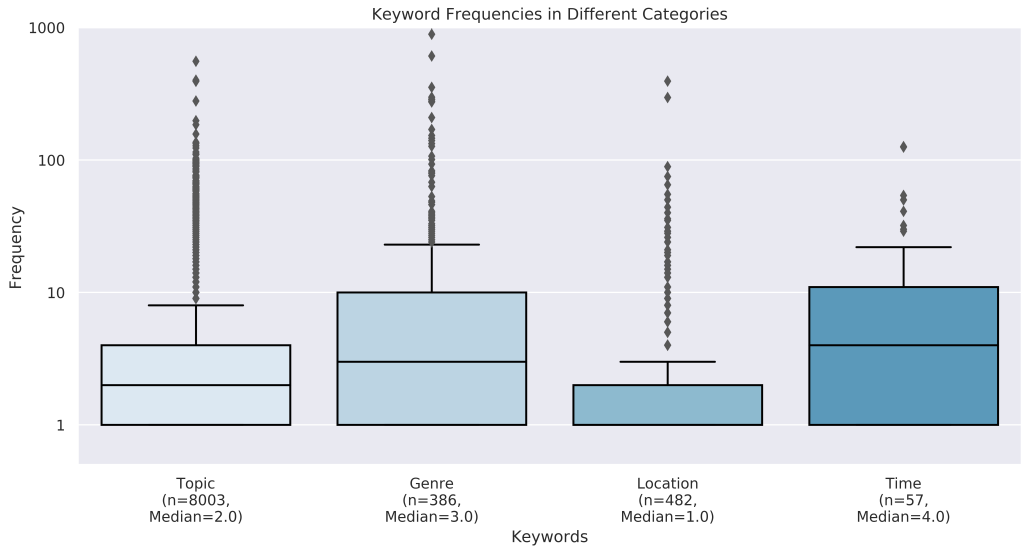

Figure 1: The frequency distributions of unique keywords in the training set. Each boxplot represents the distribution of the keywords in a specific category.

There is no specific limit set on the number of labels per one document in the library's subject indexing methodology, but the set of labels should be sufficiently large to provide an adequate overview of the book's content and key topics, yet as succinct and compact as possible. The number of labels per document in the training set varies from 1 to 35 with an average number of 5.

86.5% of the books in the training set were in Estonian, 9.8% in English, 2.5% in Russian and 1.2% in other languages, including Finnish, German, French, Latvian, Swedish etc.

The extent of the books varied from 1 to 1936 pages, with a mean number of 86 and a median number of 52.

The subject index contained the publishing dates only for a specific, often translated, edition of the books and did not have direct information of the original publishing dates. However, it did include the author's date of birth, so we could approximate the original publishing dates based on this data. It was necessary to obtain this information as the disparity between two-word distributions grows with their temporal difference and as word frequencies influence the statistical prediction models, this could thus have an effect on the results[13]. For example, if most of the training data is published after the year 2000, it would be safe to assume that the model would perform poorer on books originally published in the 15th century. The oldest author in the training data was Niccolò Machiavelli (born in 1469), followed by Wilhelm Christian Friebe (born in 1762) and Carl Friedrich von Ledebour (born in 1785). However, all the other authors present in the training set were born after the year 1800 and the majority (93.5%) in the 20th century. Assuming that the published books are usually written by authors who are at least 17 years old, we can derive that 93% of the books in the training set were originally published after the year 1917, i.e. in the last 100 years.

As the performance and the results of machine learning methods strongly depend on the underlying data, it is important to be aware of the limitations arising from it. Most of the training documents were in Estonian (86.5%) and 93% of the books were published in the last 100 years. It should also

be taken into account that the distribution of the genres and forms of the books in the public domain might differ from the distribution of genres and forms of the copyrighted books.

5 Constructing the Prediction Models

Keyword assignment methods can be roughly divided into two parts: 1) keyword extraction, where keywords are chosen from words that are explicitly mentioned in the original text and 2) keyword assignment, where keywords are chosen from a controlled vocabulary or taxonomy. As all the chosen keywords should be present in EMS, the latter technique is more suitable for following the methodology used for subject indexing by the National Library of Estonia.

As the training data is labelled, it could otherwise be an ideal input for supervised machine learning methods, but we would first need to solve the following problems:

1. How to make the machine learning methods that require hundreds of examples to acquire the necessary information if the average number of examples per label are two?

2. How to time-efficiently predict a label set of size 5-10 from more than 8000 possible label candidates?

To overcome the issue of the low number of training examples per label, we split each book into pages and linked all the labels in the book's subject index with all of its pages. For instance, a book with 456 pages would have contributed 456 examples to every label it had been assigned. It should be noted, however, that this approach had a drawback: some of the books in the training set consisted of multiple unrelated sections, e.g. article collections and short story collections. Moreover, each label assigned to the book might have not represented every page of the book equally, even if the content distribution in the book was more or less uniform. This means that some of the labels for some of the pages might have been inaccurate, but we expected that the noise arising from these mismatches would not have a noticeable impact on the grand scale.

After splitting the documents and treating every page as a separate instance, the number of examples per keyword increased considerably, with the median frequency of 393. We set the minimum number of examples for each classifier model to 50 to guarantee sufficient number of examples for each keyword and obtained the final label set of size 8003.

The text from each page is thereafter extracted with Apache TIKA[14] - an open-source tool supporting text extraction from a wide variety of formats, including images requiring optical character recognition.

Each text is then passed through quality control by feeding it to a Hidden Markov model based on the distribution of character sequences to distinguish unsuccessful text extraction (texts consisting of meaningless character sequences like "AXwQkKSj4G") from correctly extracted texts.

The texts passing the quality check are processed with a multilingual preprocessing tool Texta MLP, which uses EstNLTK[15] and spaCy[16] for identifying the language of the text, lemmatizing it and extracting part-of-speech (POS) tags. Lemmatization is a process of converting words into dictionary forms (running, ran and runs into run), which helps to reduce the size of the vocabulary and therefore provide more succinct and precise features for the machine learning models. POS tags (e.g. "noun", "verb", "adverb" etc) provide morphological information about the words based on their context and definition and have been successfully used to improve the results of classifying text genres. [17]

The extracted lemmas and POS tags were used as input features for 8003 binary Logistic Regression classifiers - one for every label in the training set.

To time-efficiency predict the labels from 8003 possible candidates, we used Hybrid Tagger - a tool aimed at extreme multi-label classification tasks. Hybrid Tagger is part of TEXTA Toolkit[18] and it enables to reduce the set of candidate tags significantly by comparing the input document to the other documents from the same domain indexed in Elasticsearch.[19] The set of candidate tags is constructed of the top n most frequent tags present in the m most similar documents. The binary classifiers corresponding to each candidate tag are then applied to the input document and tags with positive classification results are returned as the predicted labels. We used configuration n = 10 and m = 20, which means that each input document has at most 20 candidate tags reducing the total number of candidates 8003/20 = 400.5 times and thus making the prediction process notably faster. The workflow of Hybrid Tagger is depicted on Figure 2.

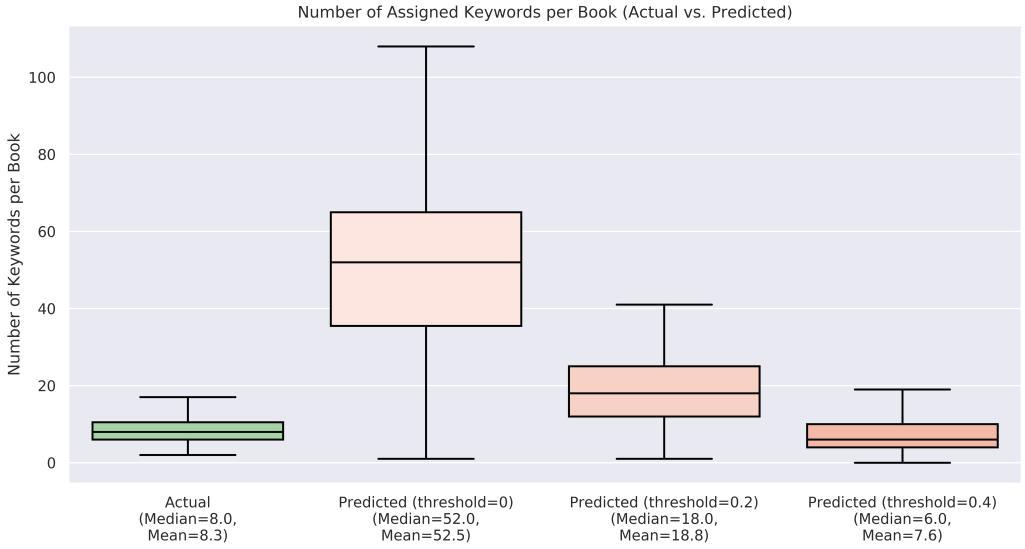

Figure 2: The number of keywords assigned to the books by a) the cataloguers and b) the automatic subject indexing tool on different frequency thresholds.

Extent	**Average number of pages to reach 90% of the maximum recall**	**Average number of pages to reach the maximum recall**
1-49 pages	10	14
50-99 pages	8	30
100-299 pages	4	39
300-499 pages	5	42
¿= 500 pages	6	45
Average	6.6	34.0

Table 1: The average number of pages it takes to reach 90% of the maximum recall and the number of pages it takes to reach the maximum recall. The maximum recall is considered the recall obtained after annotating all the pages of the book.

If the book to annotate is 500 pages long, it might not be necessary to use all the pages for the subject indexing process as a) the time it takes the automatic subject indexing tool to predict the keywords grows with the number of pages to process and b) the results stabilize after a certain page limit or even get worse if the number of irrelevant labels grows. To determine how many pages of the book we should pass to the automatic subject indexing tool, we analysed how the recall scores changed while incrementing the number of pages. We ignored f1-score-based performance at this stage as we used another method addressed in the next paragraph for specifically optimizing precision. As the number of pages it takes for the results to converge might differ depending on the extent of the book, we constructed 5 classes based on the extent and selected 20 random examples for each class. On average, it took 34 pages to fully converge, but only 6.6 to reach 90%. Moreover, the 90% was always reached after using at most 10 segments regardless of the book's extent. The results for each extent class are presented in Table 1. We have estimated that processing and annotating one page takes about 6 seconds, although the time may vary depending on the available computational power of the machine hosting the subject indexing tool. This means that it takes on average 3 additional minutes to fully converge after reaching the 90% (34.0*6 - 6.6*6 = 168 seconds). Based on these results, we decided to set the default number of pages to annotate to 10 as it is sufficient to reach at least 90% of highest possible recall on average, yet ensures a fairly time-efficient subject indexing process. However, we decided to leave the user an option to modify it.

The average number of keywords added by the cataloguers is 8, meanwhile, the average number of

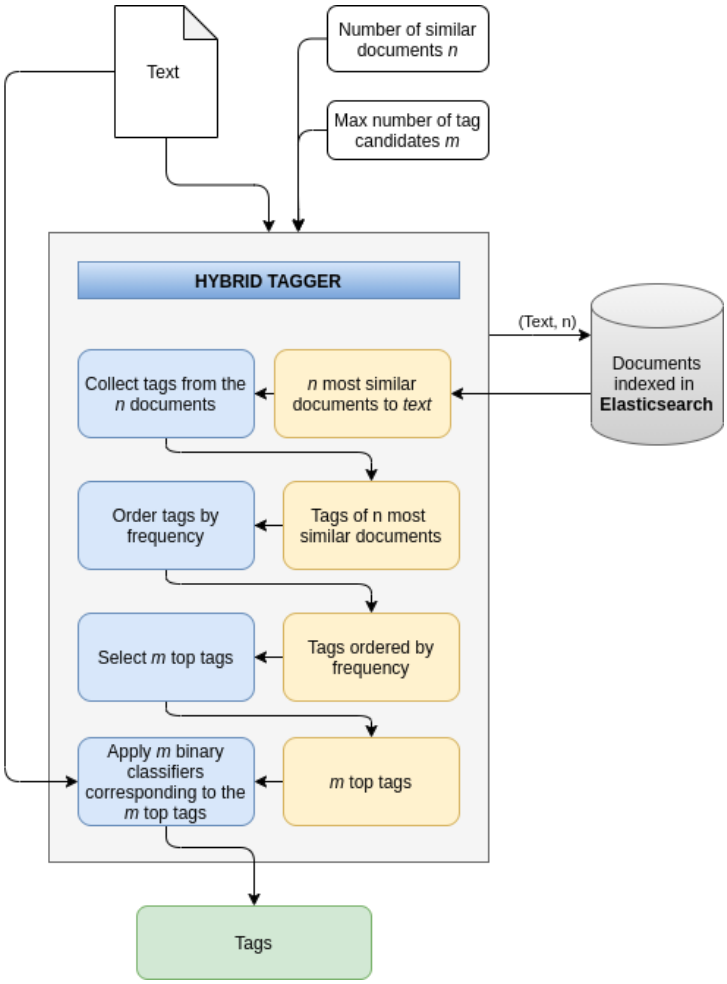

Figure 3: The workflow of Hybrid Tagger tool used for predicting the keywords.

keywords added by the keyword assignment model is 52. To retrieve fewer and more precise results, all the predicted keywords are first sorted by their frequency ($f = t/n$, where t is the number of pages the keyword occurred in and n is the number of pages used) and only the keywords passing a configurable threshold are presented to the user. The default threshold is set to 0.4 as it provides the highest F1 score on average and the number of keywords passing the threshold is most similar to the number of keywords added by the cataloguers on average (Figure 3).

6 Architecture

The prototype consists of a back-end implemented as a RESTful Python application based on Django Rest Framework and an Angular front-end communicating with the backend via API endpoints. The prototype uses Celery for handling task queues and Redis as a message broker. Besides the main component, which is responsible for handling the subject indexing tool's I/O, Kratt also includes Texta Toolkit containing the Hybrid Tagger used for predicting the labels, Texta MLP responsible for language extraction and preprocessing the data, and Elasticsearch for storing the data used by the Hybrid Tagger. All components, excluding Elasticsearch, are wrapped inside separate Docker containers to make them platform-independent and easily deployable.

The tool can be used via graphical user interface or by passing the data directly to the API.

6.1 User Interface

The books to annotate can be uploaded to the automatic subject indexing tool Kratt from the user's computer or from an external resource by providing the URL of the resource. The user can additionally modify the number of randomly selected pages n used for further processing (by default 10).

The subject indexing process takes about 1 minute, depending on the number of pages the user has selected. While the tool is processing the input, the user is displayed a progress chart with the information of the current step (e.g. "Detecting languages").

After the subject indexing tool has finished processing, the detected keywords passing the current threshold (by default = 0.4, but easily modifiable with a slider) are displayed along with the language distribution of the randomly selected pages. The user can deselect irrelevant keywords and then copy all selected keywords in MARC21 format to the clipboard.

6.2 Workflow

The workflow of Kratt consists of the following steps (Figure 4):

1. The uploaded book is converted into PDF and divided into pages.
2. Then $n + 5$ pages (the additional 5 pages are used as a buffer in case some of the selected pages do not pass quality control) are randomly selected and passed to the text extractor.
3. The extracted plaintexts undergo quality control and n texts passing the control are sent to Texta MLP.
4. The lemmas and POS tags extracted with Texta MLP are then passed to the Hybrid Tagger, which predicts the keywords for each page.
5. The keywords are then sorted by their frequency ($f = t/n$, where t is the number of pages the keyword occurred in and n is the number of pages used).
6. All keywords that exceed the configured threshold (by default 0.4) are presented to the user.

7 Results and Discussion

To evaluate the results, we applied the automatic subject indexing tool with default parameters to 315 new books subject indexed by the cataloguers during the 6 month period after finishing the prototype. The distribution of forms, genres and extent of the books was similar to the training data. However, the language distribution deviated from the original with only 59.0% of the books in the test set being in Estonian - 27.5% less compared to the training data. The other languages in the test set included English (28.9%), Russian (7.3%) and a small percentage of others (4.8%). The books contained 1474 unique preferred terms of which 1222 (82.9%) were present in the training data while developing the automatic subject indexing tool and 252 (17.1%) were not. We excluded all the keywords not present in the training set while evaluating the results to get a better grasp of how well the model predicts keywords it should be able to predict and treat the rather large number of unseen labels as a separate problem.

We then compared the subject indices added by the cataloguers with the keywords predicted by the automatic subject indexing tool and calculated precision, recall and f1-score for each book. The overall scores were then calculated by averaging all the individual books' scores. The distribution of resulting prediction scores on thresholds 0, 0.2, and 0.4 can be seen in Figure 5. The best average f1-score was 0.3 and it was reached equally on thresholds 0.2 and 0.4, with the first providing higher average recall and the latter providing higher average precision.

Although the scores are rather underwhelming, there are several possible explanations and ways of improvement:

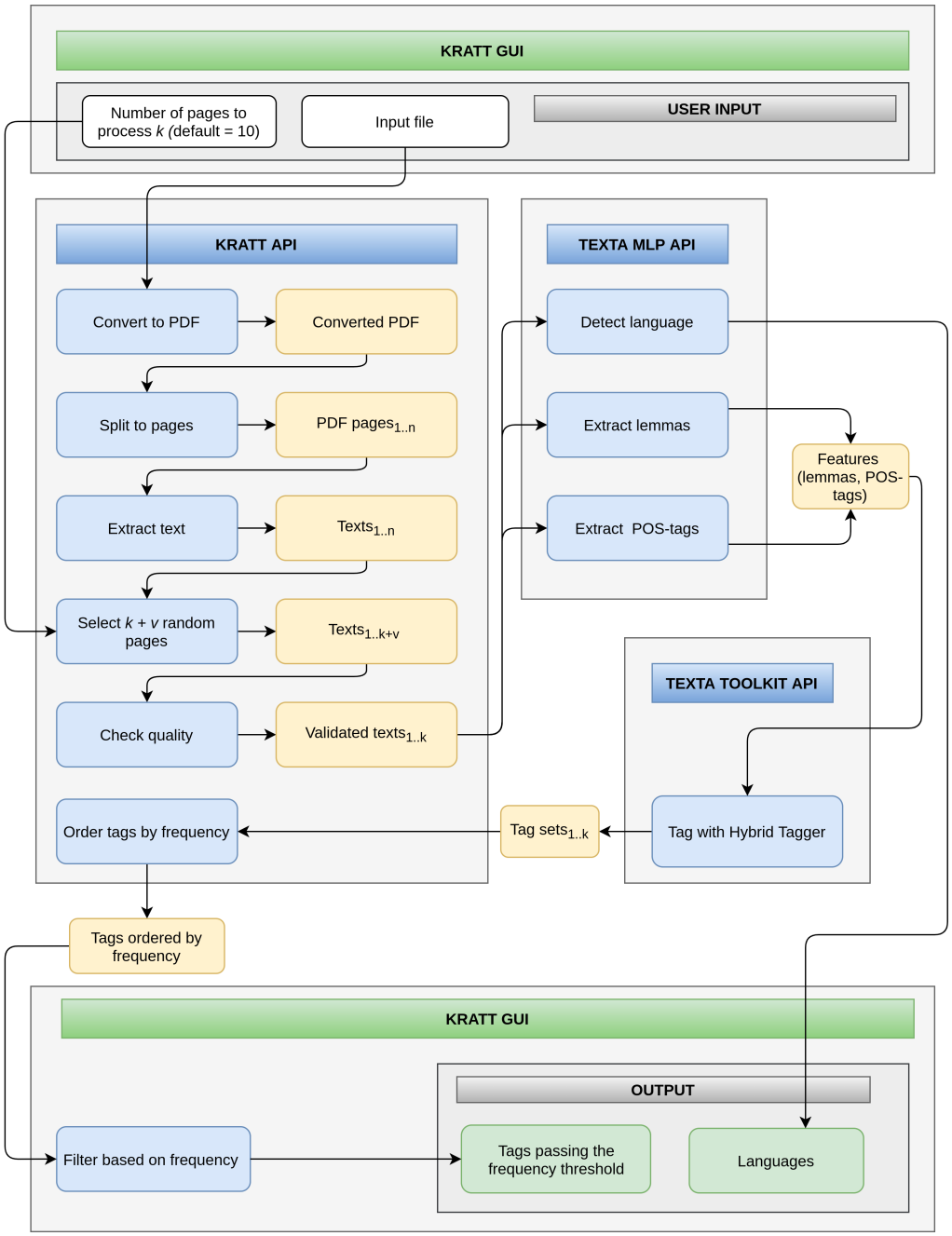

Figure 4: The workflow of the automatic subject indexing tool Kratt.

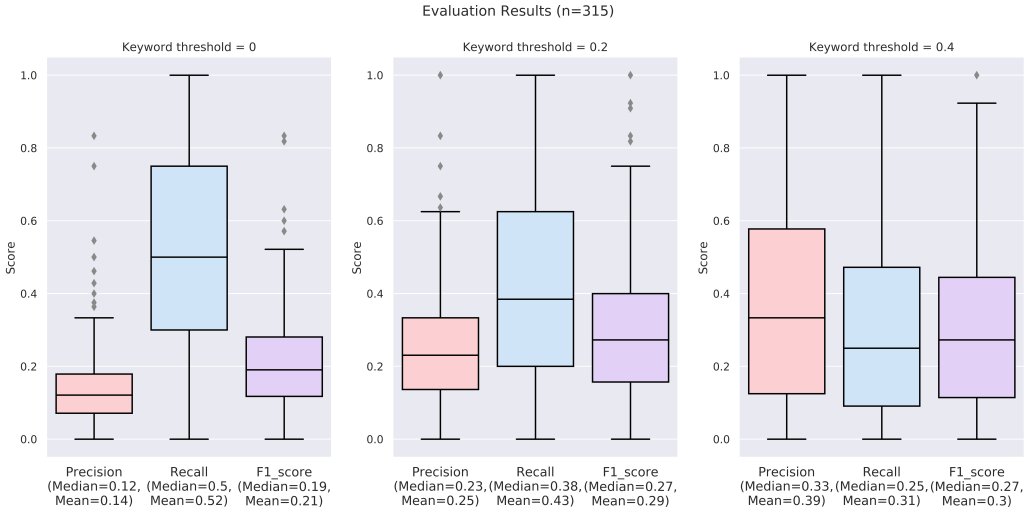

Figure 5: The evaluation scores based on comparing the automatic subject indexing results of 315 books to the keywords added by the cataloguers. (1) Evaluation results without a threshold for keyword frequencies; (2) evaluation results with the keyword frequency threshold set to 0.2; (3) evaluation results with the keyword frequency threshold set to 0.4.

1. The noise generated by assigning keywords to all pages might have more considerable effect than we originally anticipated. One possible solution for improvement is to try mapping only a relevant subset of the labels with each page and retrain the models.

2. Although the number of examples per keyword increased remarkably after splitting each book into pages, the number of unique books linked with an average keyword stayed the same. This might have led to overfitting on the features tied to the content in the limited number of books. The best solution for this problem is to use more data, which could be accomplished by including the copyrighted books, which were not available during the development of the prototype due to legal complications. It is also worth exploring opportunities arising from large pretrained BERT[20] models as they require less training data compared to the classical machine learning methods like logistic regression or support vector machines.

In addition to the aforenamed explanations, we should also consider that the labels added by the cataloguers used for calculating the scores are subjective to some extent: even the labels chosen by two different cataloguers might not fully coincide. Furthermore, the subject indices are not usually 100% correct or 100% incorrect - there exists a grey area consisting of keywords, which are more or less adjacent to the topic, but considered excessive by the subject indexing methodology followed by the cataloguers.

To get a better overview of the results regarding these aspects, we surveyed a small number of professional cataloguers and regular library users in addition to the automatic evaluation.

Five cataloguers working in the National Library of Estonia tested the tool with 20 books each. The cataloguers did not consider the results satisfactory and claimed that the results are rather causing extra confusion: if the prototype suggests a keyword originally not added by the cataloguer, it takes the cataloguer additional time to find out if the keyword proposed by the tool is actually relevant to the book's content and if it should be used in the subject index or not. These results are in alignment with a survey investigating the attitudes of German- and English speaking librarians towards automatic subject indexing. The findings of the survey showed that the librarians assess the quality of automatic subject indexing systems with scepticism and over 60% of the respondents believed that machines will never be able to outperform humans in this task.[21]

As the methodology of cataloguing and subject indexing follows a strict set of rules, the library workers may perceive the quality of the keywords differently from a regular user. To determine how the

regular users perceive the results, we chose 10 random books from the test set and let 6 users evaluate the results based on the criteria presented in Table 2. To make sure that the randomly chosen books are representative of the test sample, we plotted the data points of the books on a figure representing the average prediction scores (f1-score, recall, precision) depending on the extent of the book. As most of the data points stay below the lines representing the average scores, we are confident that the chosen books did not distort the perception of the results of the automatic subject indexing tool in a favorable way. We presented three sets of subject indexing results for each book: 1) the original labels added by the cataloguers; 2) the labels added by the automatic subject indexing tool after using 10 pages of a book; and 3) the labels added by the automatic subject indexing tool after using all the pages of a book. The threshold was set to 0.2 to enhance recall rather than precision. The results grouped by different types of keywords are presented in Table 3. On average, all users evaluated the keywords added by the cataloguers higher than the keywords assigned by the automatic subject indexing tool. However, there was a noticeable difference between different subcategories. While the topic keywords added by the cataloguers were considered better than the subject indexing tool's results, Kratt outperformed cataloguers slightly with the prediction of genres. It is also worth mentioning that meanwhile the time keywords added by the cataloguers had a higher average score than the keywords suggested by the automatic subject indexing tool in the same category, the cataloguers added time keywords only to 2 books out of 10. Kratt predicted time keywords to 5-6 books out of 10 (depending on the number of pages used) and still received relatively high evaluation scores (4.5-4.6) from the users. As Kratt's average results predicting the keywords is slightly over 4, we can conclude that the user's perception of adequate keywords does not necessarily mirror the cataloguers' perception. However, it should be noted that this specific evaluation process measured only how well the keywords were able to summarize the book, but this is not the only purpose of the keywords. They are also used for searching information from the library's databases and while a couple of false positives or false negatives may not have a significant impact on the general overview of the book, they can negatively affect the user's experience while using the database. This also helps to explain the disparity between the users' and cataloguers' opinions as the latter group is used to considering all possible implications of the labelset.

Rating	**Description**
1	The keywords are irrelevant and do not represent the content of the book.
2	Some of the keywords are accurate, but the larger majority does not represent the content of the book
3	Fair amount of the keywords are relevant to the topics covered in the book, but there exists a sufficiently significant amount of irrelevant keywords to cause confusion.
4	Most of the keywords are relevant, but a few are not.
5	The keywords give a decent overview of the book and are relevant to the topics covered in the book.

Table 2: The description of the ratings used for surveying the users about the keywords added by a) the cataloguers b) the automatic subject indexing tool Kratt.

	Topic	**Location**	**Time**	**Genre**	**All keywords**
Cataloguer	4.59	4.69	5	4.28	4.64
Kratt / 10 pages	3.43	4.32	4.5	4.31	4.14
Kratt/ All pages	3.98	4.38	4.6	4.33	4.32

Table 3: The average user ratings of different types of keywords added by a) the cataloguers b) the automatic subject indexing tool Kratt.

As one of the goals of developing the automatic subject indexing tool is to save resources, we measured the time Kratt spends for subject indexing a book and compared it with the performance of an average cataloguer. It currently takes Kratt about 1 minute to predict keywords for a book of any size with the default settings. There is no additional time cost for longer books as only 10 random pages are used for predictions (by default) and the time consumption of processes like converting and

splitting the file is negligible. It should also be noted that increasing the training data and the set of keywords does not affect the time consumption as the prediction time of Hybrid Tagger does not depend on the number of possible targets. As it takes an average cataloguer about 15 minutes to subject index a book, the speed of automatic subject indexing tool outperforms human specialists. Furthermore, the automatic subject indexing tool does not need to rest and can work 24 hours a day, 7 days a week.

As one of the goals of developing the automatic subject indexing tool is to save resources, we measured the time Kratt spends for subject indexing a book and compared it with the performance of an average cataloguer. It currently takes Kratt about 1 minute to predict keywords for a book of any size with the default settings. There is no additional time cost for longer books as only 10 random pages are used for predictions (by default) and the time consumption of processes like converting and splitting the file is negligible. It should also be noted that increasing the training data and the set of keywords does not affect the time consumption as the prediction time of Hybrid Tagger does not depend on the number of possible targets. As it takes an average cataloguer about 15 minutes to subject index a book, the speed of automatic subject indexing tool outperforms human specialists. Furthermore, the automatic subject indexing tool does not need to rest and can work 24 hours a day, 7 days a week.

8 Conclusion

The goal for developing the prototype of an automatic subject indexing tool Kratt was to test whether the automation of the process would be possible, if it might save time and money, if it would help to improve the quality and if Kratt could be integrated into the daily cataloguing workflows. Our results demonstrated that while the automation process is possible and more time efficient than manual cataloguing, the quality of the predicted subjects is currently not sufficient for integrating the automated process into the library's daily workflows. However, we do not rule out the possibility of including it in the future, if the proposed methods of enhancing the quality of the models prove to be successful.

9 Funding

The project of developing the prototype of an automatic subject indexing tool was supported and financed from the European Regional Development Fund (ERDF). The work described in this paper has also been supported by the language technology research and development program "Estonian Language Technology 2018–2027" of the Ministry of Education and Research under grant EKTR3, by European Union's Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for LessRepresented Languages in European News Media) and Enterprise Estonia project No. EU48684, Research Project No. 1.11 (Deep neural models and cross-lingual embeddings in TEXTA Toolkit).

10 Data Availability Statement

The processed data used for developing the prototype described in this study are available in DIGAR at http://data.digar.ee/#kratt. These data were derived from the following resources available in the public domain: https://www.digar.ee/arhiiv/en (book files), https://erb.nlib.ee/?lang=eng (metadata)

Notes

[1]Estonian Subject Thesaurus https://ems.elnet.ee/

[2] Kont, K.-R., 2015. "How Much Does It Cost to Catalog a Document? A Case Study in Estonian University Libraries". Cataloging & Classification Quarterly, 53:7 (2014), 825-50, DOI: 10.1080/01639374.2015.1020463

[3]Pokorny, J. "Automatic Subject Indexing and Classification Using Text Recognition and Computer-Based Analysis of Tables of Contents". Toronto, ELPUB, 2018. DOI: 10.4000/proceedings.elpub.2018.19

[4]According to the Estonian National Bibliography, the annual production of publications is approximately 11 000 items (both print and e-publications).

[5]Cloete, L. M.; Snyman, R.; Cronje, J.C., "Training Cataloguing Students Using a Mix of Media and Technologies". Aslib Proceedings, 55:4 (2003), 223–233. DOI: 10.1108/00012530310486584

[6]Pokorny, "Autmatic Subject Indexing"

[7]National artificial intelligence strategy for 2019-2021 https://en.kratid.ee/

[8]In Estonian mythology, Kratt is a magical creature. Essentially, Kratt was a servant built from hay or old household items by its master who then had to cede three drops of blood for the devil to bring life to the kratt. The kratt was notable for doing everything the master ordered. Therefore, the Estonian government uses this character as a metaphor for AI and its complexities

[9]Junger, U., 2014. "Can Indexing Be Automated? The Example of the Deutsche Nationalbibliothek". Cataloging & Classification Quarterly, 52:1, 102-9. DOI: 10.1080/01639374.2013.854127

[10]Suominen, O., 2019. "Annif: DIY Automated Subject Indexing Using Multiple Algorithms". LIBER Quarterly, 29 (1), 1–25. DOI: 10.18352/lq.10285

[11]DIGAR stores online publications, print files and digitised copies of publications. https://www.digar.ee/arhiiv/en

[12]ERB registers all publications issued in Estonia, as well as Estonian-language publications, works by Estonian authors and their translations, and foreign-language publications about Estonia and Estonians issued abroad. https://erb.nlib.ee/?lang=eng

[13]Jatowt, A.; Tanaka, K., 2012. "Large Scale Analysis of Changes in English Vocabulary over Recent Time". CIKM '12: Proceedings of the 21st ACM international conference on Information and knowledge management, 2523-26. DOI: 10.1145/2396761.2398682

[14]https://tika.apache.org/

[15]EstNLT is an open-source tool for Estonian natural language preprocessing. https://estnltk.github.io/estnltk/1.4.1/

[16]spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. https://spacy.io/

[17]Feldman, S.; Marin, M. A.; Ostendorf, M; Gupta, M. R., 2009. "Part-of-speech Histograms for Genre Classification of Text", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009, 19-24 April 2009, Taipei, Taiwan. DOI: 10.1109/ICASSP.2009.4960700

[18]Texta Toolkit is an open-source framework for building and executing machine learning pipelines and analysing textual content.

[19]Vaik, K.; Asula, M.; Sirel, R., 2020. "Hybrid Tagger – An Industry-driven Solution for Extreme Multi-label Text", Proceedings of the LREC2020 Industry Track, 26–30.DOI: 10.5281/zenodo.4306169

[20]https://github.com/google-research/bert

[21]Keller. A. "Attitudes among German- and English-Speaking Librarians toward (Automatic) Subject Indexing", Cataloging & Classification Quarterly, 53:8 (2015), 895-904. DOI: 10.1080/01639374.2015.1061086