Further Keyword Generation Experiment in Hungarian with Fine-tuning PULI LlumiX 32K Model

Réka Dodé
Hungarian Research Centre for Linguistics
Budapest, Hungary
dode.reka@nytud.hun-ren.hu

Zijian Győző Yang
Hungarian Research Centre for Linguistics
Budapest, Hungary
yang.zijian.gyozo@nytud.hun-ren.hu

***Abstract*—Our research continues an investigation using neural models to generate and extract keywords from lengthy texts, using data from the REAL repository and author-provided keywords.**

Previously, we tested three models: fastText for keyword extraction as a multi-label classification baseline, a fine-tuned Hungarian language model PULI GPT-3SX for keyword generation, and a further trained Llama-2-7B-32K model.

In this study, we fine-tuned a new model, the PULI LlumiX 32K model with the same data, combining Hungarian language knowledge with Llama-2-7B-32K's 32,000-token input capacity.

We assessed the generation of new, relevant keywords by the models compared to author-provided keywords and those not present in the text. The PULI LlumiX 32K model outperformed both the PULI GPT-3SX language model and Llama-2-7B-32K model. For keywords not present in the text, PULI LlumiX 32K and Llama-2-7B-32K generated approximately 20%, similar to author keywords. PULI GPT-3SX had a higher ratio of about 30%. Some new keywords were relevant, while others were inaccurate due to erroneous phrases.

***Index Terms*—PULI LlumiX 32K, generated keywords, fine-tuning, author-provided keywords, Llama-2-7B-32K, PULI GPT-3SX, Hungarian language model**

I. Introduction

Keywords play a significant role in structuring texts, categorizing information, and facilitating easier navigation between texts. In the case of scientific publications, articles, and professional texts, keywords not only serve to summarize content but also aid in the discovery of texts for search engines and library catalogs. Additionally, they can be useful in terminological work. However, selecting and determining keywords is not always a simple task.

Keywords manually provided by authors often rely on their own professional background knowledge [1] [2] and are not necessarily based solely on the frequency of occurrence in texts, which can sometimes lead to keywords that are not necessarily characteristic or common. This is important because keywords play a prominent role in mapping the conceptual framework of texts and the specific domain.

In recent years, neural large language models, such as GPT-3 [3] and BERT [4] have brought revolutionary changes to language processing. These models are capable of interpreting language in a complex manner and making predictions about the relationship between words and texts.

The application of generative systems offers the opportunity to create keywords that may not or only partially appear in the text, which could represent a breakthrough in this field.

II. Related work

Keyword extraction has a long history. Various methods have emerged to address the problem, following technological advancements tailored to specific goals (e.g., information extraction, text mining, or even terminology extraction to facilitate terminological work).

Keyword extraction, using author-provided keywords, was already explored from Hungarian scientific publications in 2010 by Berend and Farkas [5]. The author-provided keywords were expanded with a feature set, and supervised machine learning was applied with this training data.

Keyword extraction can also be interpreted as a classification task, where the keywords are elements of a large or even open-ended label set in the training data [6]. An example of this is the Hungarian keyword and label extraction system developed by Yang and colleagues [7], which was fine-tuned with texts from the weekly HVG.

Over the years, there has been a growing demand for keywords not only to be extracted from the text but also to incorporate external knowledge to obtain new insights beyond extraction. An example of this is the MAUI method [8] where external knowledge is sourced from Wikipedia. In the Hungarian context, research has used Wikipedia as external knowledge in labeling tasks [5].

However, large language models go beyond solely relying on the text or using an external database (e.g., Wikipedia or even a terminology database) for keyword generation. Among multilingual models, Llama models are currently popular, which also have Hungarian knowledge. Currently, there are two Llama families, LLaMA [9] and Llama-2 [10]. Both families contain multiple differently sized large language models. The Llama-2 models were trained on a corpus of 2 trillion tokens.

979-8-3503- 8788-9/24/$31.00 © 2024 IEEE

Currently, there are four large language models available for the Hungarian language. HILANCO-GPTX[1] is a 6.7 billion parameter, English-Hungarian bilingual GPT-NeoX model as well as the PULI large language models, including the Hungarian-language, also 6.7 billion parameter PULI GPT-3SX [3]. Additionally the trilingual (Hungarian-English-Chinese) 7.67 billion parameter PULI GPTrio [11] GPT models, and the PULI LlumiX 32K, a Llama-2 model fine-tuned with a 32K context window for the Hungarian language can come in handy.

Our research predecessors [12] compared the keywords extracted by fastText with those generated by the Hungarian language model PULI (GPT-3) and the Llama 2 model. The results indicated that the outcomes of the language models were more promising than those of fastText. Additionally, it was found that the Llama 2 model with a 32,000-token input, by being able to learn from entire long texts, achieved significantly higher accuracy and coverage than the 2048-token input PULI model.

III. Training data

The teaching material coincided with the teaching material used in the previous study [12] in order to compare the results. For the sake of completeness of the study, we also describe here how we compiled the teaching material. The REAL repository's articles[2] were used. During compilation, materials from after 2010 were processed, assuming that keyword addition became common around this time. A total of 29,502 files were obtained, which were Hungarian-language articles published after 2010 on various scientific topics; from these, 9,226 articles contained the ”*kulcssz**” (keyw*) pattern based on our pre-filtering. The materials from the REAL repository are available for download in PDF format. Using the Tesseract OCR engine version 5.0[3] developed by Google, we converted the PDF materials into txt format. Among the 9,226 articles, we used 1,146 texts where the script found and extracted the keywords provided by the author from the OCR-ed text. In cases where the number of keywords was very high, we manually verified them and excluded the conference proceedings from the corpus. The texts that remained alongside the extracted keywords no longer contained the keywords.

For our research, we divided the corpus into training and test data. To do this, we shuffled the corpus and separated 1,000 articles for training and 145 articles for testing.

The key characteristics of the author-provided keywords/phrases found in the articles are as follows:

- Number of unique keywords/phrases: 5,546
- Number of unique words: 5,382
- Average length of key expression: 1.63
- Average number of keywords per document: 4.70

Table I and Table II displays the key characteristics of the articles, as well as the properties after tokenization using the models we employed. It can be observed that we worked with articles averaging around 5,000 words. To determine the number of sentences, we utilized the HuSpaCy tool [13]. No tokenizer was used during the original text measurement; we only relied on HuSpaCy for sentence counting.

TABLE I
Key properties of documents 1.

	Tokens	Avg. token count/ doc.	
		Average	Median
Original text	5,632,804	4,915.19	4,611.5
PULI GPT-3SX	11,078,832	9,667.39	9,181.0
Llama-2-7B-32K	18,029,175	15,732.26	14,924.0
PULI LlumiX 32K	18,029,175	15,732.26	14,924.0

TABLE II
Key properties of documents 2.

	Sentences	Avg. sent. count/ doc.	
		Average	Median
Original text	295,900	258.20	242.5

In Table I, the token counts after tokenization for the models are shown. There is a significant difference between the text segmentation of the two models; the Hungarian language PULI model breaks down the text much less compared to the English-centric Llama-2 and PULI LlumiX 32K models.

IV. Methods

The models used in previous research were fastText, PULI GPT-3SX, and Llama-2-7B-32K. We provide their descriptions based on the published study [12].

- **fastText** [14], [15]: The development of Meta Research[4] aimed at efficient training of word representation and text classification models. Its performance in text classification competes with other deep learning-based solutions and is extremely fast. Pre-trained word vectors are available on the platform for 294 different languages, trained from Wikipedia texts. For our experiment, we utilized the pre-trained model for the Hungarian language. It is capable of processing texts of any length.
- **PULI GPT-3SX** [3]: The Hungarian language GPT-NeoX model [16] trained by the HUN-REN Language Research Centre[5]. The model is comprised of 6.7 billion parameters and was pre-trained on a corpus of over 32 billion words. It is capable of handling 2048 input tokens.
- **Llama-2-7B-32K** [10]: Together[6] is equipped with 7 billion parameters. During fine-tuning, the model's input length was extended to 32768 using the position interpolation method [17]. This allows for the processing of long documents.

[1] https://hilanco.github.io
[2] http://real.mtak.hu
[3] https://github.com/tesseract-ocr
[4] https://research.facebook.com
[5] https://nytud.hu
[6] https://www.together.ai

A. PULI LlumiX 32K model

PULI LlumiX 32K[7] model was trained by the HUN-REN Hungarian Research Centre for Linguistics[8]. The main advantages of this model include an expanded context window of 32,768 tokens, enabling the processing of entire documents for our task. Additionally, it offers superior Hungarian language capabilities compared to the original Llama2 model.

The PULI LlumiX 32K model is a multiple further pretrained Llama-2 7B model [9]. First, the Together[9] fine-tuned a LLaMA-2-7B-32K long context language model from the Llama-2 7B model. The model has been extended to a context length of 32,768 with position interpolation. Subsequently, the LLaMA-2-7B-32K model was continuously pretrained on a Hungarian dataset. The corpus consisted of 7.9 billion words, exclusively comprising long documents exceeding 5000 words in length. During the Hungarian pretraining phase, the first half epoch exclusively utilized the Hungarian dataset. However, in the second half epoch, English corpora were mixed into the training data. The English dataset[10] consisted of 2 billion words from the Long Context QA and 78 million words from BookSum.

V. Experiments

The experiments conducted with the three models are summarized based on the published study [12].

- **fastText**: To train the model, we continued training the pre-trained Hungarian language model for the classification task. Vector representation size to 300 dimensions, 'wordNgrams' parameter: 3, multi-label classification feature set, learning rate: 1.0 and the model trained up to 100 epochs.
- **PULI GPT-3SX**: We fine-tuned the model utilizing the implementation of Stanford Alpaca (see [18]) and employed the instruction-following prompt template (see Table III). We reduced the text of the articles to 512 words. Hyperparameters: 4 batch/GPU (8 GPUs total); 'gradient accumulation steps': 4; learning rate: 2e-5; 'warmup ratio': 0.03; deepspeed optimization; bf16. 3 epochs achieved the best result.
- **Llama-2-7B-32K**: We did not perform preprocessing. For the experiment, we fine-tuned the model named togethercomputer/LLaMA-2-7B-32K[11] using the implementation of OpenChatKit [19]. Hyperparameters: 4 batch/GPU (8 GPUs total); learning rate: 2e-5; fp16; epochs: 10. We finally set it to 10 epochs. The prompt template can be seen in Table III.

A. Experiment of PULI LlumiX 32K model

In our experiment, we fine-tuned the PULI LlumiX 32K model for the keyword generation task. For this task, we utilized the same prompt template as in the Llama2 experiment (see Table III). For the fine-tuning task, we utilized the OpenChatKit implementation[12]. The hyperparameters were set similar to those used in the Llama-2-7B-32K experiment: 4 batches per GPU (8 GPUs in total); learning rate of 2e-5, mixed precision training (fp16), and 10 epochs.

TABLE III
Prompt template

Stanford Alpaca
Az alábbiakban egy utasítást találsz, amely leír egy feladatot, amelyhez egy bemenetet is mellékelünk, hogy további összefüggéseket adjon. Írj egy választ, amely megfelelően teljesíti a feladatot! (Below you'll find an instruction describing a task, along with an input provided to offer further context. Write a response that adequately fulfills the task!) ### Instruct: Generálj kulcsszavakat az alábbi szöveg alapján! (Generate keywords based on the provided text!) ### Input: [content of the article] ### Answer: [keywords]
OpenChatKit
[content of the article] <Q>: Generálj kulcsszavakat a megadott szöveg alapján! (Generate keywords based on the provided text!) <A>: [keywords]

VI. Results

Table IV, the results of our models are shown. For comparability, we configured fastText to generate 5 labels for each document. It can be observed that all generative models learned approximately how many labels to generate for the articles. The results indicate that PULI LlumiX 32K achieved the best result, although the improvement is no longer outstanding, unlike the improvement between PULI GPT-3SX and fastText, as well as between Llama-2-7B-32K and PULI GPT-3SX.

TABLE IV
Results

	Precision	Recall	F-score	Avg # of tags
fastText (100 epoch)	5.57	14.1	7.99	5
PULI GPT-3SX	14.64	13.89	14.26	4.94
Llama-2-7B-32K	38.45	35.12	36.71	4.77
PULI LlumiX 32K	**41.05**	**38.79**	**39.89**	**4.92**

A. New keywords

Author-provided keywords often do not rely on term frequency - unlike statistically-based keywords - and do not adhere to the vocabulary found in the text corpus [1]. Dodé also points out that the occurrence of manually provided keywords in the text is not necessarily frequent [2]. In the corpus he examined, 37% of the keywords appear fewer than twice in the texts.

In our research, we examined the generated keywords that were new compared to the author-provided keywords, as well as those that did not appear in the text. In our text

[7] https://huggingface.co/NYTK/PULI-LlumiX-32K

[8] https://nytud.hu

[9] https://www.together.ai

[10] https://huggingface.co/datasets/togethercomputer/Long-Data-Collections

[11] https://huggingface.co/togethercomputer/LLaMA-2-7B-32K

[12] https://github.com/togethercomputer/OpenChatKit

searches, we counted verbatim matches after converting the text to lowercase. Table V shows the number of new keywords compared to the author-provided keywords.

TABLE V
NEW KEYWORDS COMPARED TO THE AUTHOR-PROVIDED KEYWORDS

	New keywords (total)	New keywords/ doc (avg)
PULI GPT-3SX	536	4.19
Llama-2-7B-32K	384	2.62
PULI LlumiX 32K	391	2.88

The Table VI provides an example of new but still relevant keywords generated by language models, primarily PULI GPT-3SX. The Llama-2-7B-32K and PULI LlumiX 32K models also generated new keywords, but they were not as innovative or creative as PULI GPT-3SX. These models tended to adhere more closely to the keywords provided by the author.

TABLE VI
EXAMPLE 1.

	Keywords
author-provided	hígtrágya, edc, ösztrogénhatás, hormonkészítmények, ivarzásindukálás
fastText	talaj, prevenció, talajosztályozás, prevalencia, megújuló energia
PULI GPT-3SX	hormonok, endoösztrogének, ösztrogénhatású vegyületek, xenoösztrogének
Llama-2-7B-32K	trágya, edc, ösztrogénhatás, hormonkezelések, ivarzásindukció
PULI LlumiX 32K	trágya, edc, ösztrogénhatás, hormonkészítmények, ivarzásindukció

As shown in Table VII, PULI GPT-3SX generated more new keywords that did not appear in the text: 10% more than what is characteristic for the author's keywords. Both Llama-2-7B-32K and PULI LlumiX 32K show similar ratios to the author's keywords, with approximately 20% of the keywords not appearing in the text.

TABLE VII
RATIOS OF GENERATED KEYWORDS (OCCUR AND DO NOT OCCUR IN THE TEXT)

	occur in the text	not occur in the text
author-provided	**583 (76.4%)**	**180 (23.6%)**
PULI GPT-3SX	486 (67.3%)	236 (32.7%)
Llama-2-7B-32K	396 (76.6%)	163 (23.4%)
PULI LlumiX 32K	391 (79.2%)	145 (20.2%)

In Table VIII there are a few examples of generated keywords that do not appear in the text. In Table VIII, the cases where new keywords are generated compared to the author's provided keywords and do not appear in the text are highlighted in bold. These results are also shown as models in Table IX.

TABLE VIII
EXAMPLE 2.

	occur in the text	not occur in the text
author-provided	*herpes zoster, övsömör, bárányhimlő*	*reinfekció, reaktíválódás*
PULI GPT-3SX	*herpes zoster, humán herpeszvírus*	***humán herpeszvírus-vírusok, humán herpeszvírus-fertőzés, humán herpeszvírus-***
Llama-2-7B-32K	*herpes zoster, bárányhimlő*	***újfertőzés, reaktiváció***
PULI LlumiX 32K	*herpes zoster, bárányhimlő, reaktiválódás*	***újrafertőződés***

TABLE IX
NUMBER OF KEYWORDS THAT NEITHER APPEAR IN THE TEXT NOR MATCH THE AUTHOR'S KEYWORDS

	No. of new keywords not occur in texts
PULI GPT-3SX	236 (32.7%)
Llama-2-7B-32K	148 (21.2%)
PULI LlumiX 32K	129 (18%)

B. Conclusion

The PULI LlumiX 32K and the Llama-2-7B-32K models learned to identify keywords from long texts, while the PULI GPT-3SX model generated many creative and mostly relevant keywords. As for the generated keywords not present in the text, in the cases of PULI LlumiX 32K and Llama-2-7B-32K, the ratio was approximately 20%, similar to the author's keywords (23.6%). For PULI GPT-3SX, the ratio was higher (32.7%). Within these, we examined how many keywords neither appeared in the text nor among the author's keywords. The ratios were similar here as well: around 20% for the Llama-2-7B-32K and PULI LlumiX 32K models, and around 30% for PULI GPT-3SX. This means that the majority of the keywords generated by the models which are not in the texts (in the case of PULI GPT-3SX, all of them) do not appear among the author's keywords either, so they are novel compared to both the author and the text.

Among the new keywords not occurring in the text, some were relevant (for example, synonyms of the keywords provided by the author), but there were some cases where the search was inadequate, and erroneous phrases (truncated endings, spelling errors or foreign language elements) were why they did not appear in the text. Additionally, the generation of multi-element structures (e.g., *jánoshalmi késő középkori templom* 'late medieval church in Jánoshalom') can also be a reason for the novelty.

These models offer promising solutions to the challenges of keyword extraction. Keyword generation opens up new horizons for most applications that use keywords, but in the long run, also for terminology through the representation of texts and domain content.

REFERENCES

[1] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Conference on Empirical Methods in Natural Language Processing*, 2003. [Online]. Available: https://api.semanticscholar.org/CorpusID:5723599

[2] R. Dodé, "Kulcsszavak és terminusok vizsgálata a REAL repozitóriumának anyagán – pilot kutatás," in *Tudásmegosztás, Információkezelés, Alkalmazhatóság. A MANYE Kongresszusok Előadásai*, 2023, p. in press.

[3] Z. Gy. Yang, R. Dodé, G. Ferenczi, E. Héja, K. Jelencsik-Mátyus, A. Kőrös, L. J. Laki, N. Ligeti-Nagy, N. Vadász, and T. Váradi, "Jönnek a nagyok! BERT-Large, GPT-2 és GPT-3 nyelvmodellek magyar nyelvre," in *XIX. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2023)*. Szeged, Hungary: Szegedi Tudományegyetem, Informatikai Intézet, 2023, pp. 247–262.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[5] G. Berend and R. Farkas, "Kulcsszókinyerés magyar nyelv tudományos publikációkból ," in *VII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2010)*. Szeged, Hungary: Szegedi Tudományegyetem, Informatikai Intézet, 2010, pp. 47–55.

[6] N. Firoozeh, A. Nazarenko, F. Alizon, and B. Daille, "Keyword Extraction: Issues and Methods," *Nat. Lang. Eng.*, vol. 26, no. 3, pp. 259–291, 2020.

[7] Z. G. Yang, A. Novák, and L. Laki, "Automatikus tematikuscímke-ajánló rendszer sajtószövegekhez," in *XVI. Magyar Számítógépes Nyelvészeti Konferencia*. Szeged, Magyarország: Szegedi Tudományegyetem, Informatikai Intézet, 2020, pp. 155–168.

[8] O. Medelyan, "Human-competitive automatic topic indexing," Ph.D. dissertation, The University of Waikato, Hamilton, New Zealand, 2009. [Online]. Available: https://hdl.handle.net/10289/3513

[9] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.

[10] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open Foundation and Fine-Tuned Chat Models," 2023.

[11] Z. Gy. Yang, L. J. Laki, T. Váradi, and G. Prószéky, "Mono- and multilingual GPT-3 models for Hungarian," in *Text, Speech, and Dialogue*, ser. Lecture Notes in Computer Science. Plzeň, Czech Republic: Springer Nature Switzerland, 2023, pp. 94–104.

[12] R. Dodé and Z. Gy. Yang, "Kulcsszógenerálás magyar nyelvű, hosszú szövegekből nagy nyelvi modellekkel," in *XX. Magyar Számítógépes Nyelvészeti Konferencia*, G. Berend, G. Gosztolya, and V. Vincze, Eds. online: Szegedi Tudományegyetem, 2024, pp. 257–267.

[13] G. Orosz, Z. Szántó, P. Berkecz, G. Szabó, and R. Farkas, "HuSpaCy: an industrial-strength Hungarian natural language processing toolkit," in *XVIII. Magyar Számítógépes Nyelvészeti Konferencia*, 2022, pp. 59–73.

[14] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 427–431.

[15] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[16] A. Andonian, Q. Anthony, S. Biderman, S. Black, P. Gali, L. Gao, E. Hallahan, J. Levy-Kramer, C. Leahy, L. Nestler, K. Parker, M. Pieler, J. Phang, S. Purohit, H. Schoelkopf, D. Stander, T. Songz, C. Tigges, B. Thérien, P. Wang, and S. Weinbach, "GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch," 9 2023. [Online]. Available: https://www.github.com/eleutherai/gpt-neox

[17] S. Chen, S. Wong, L. Chen, and Y. Tian, "Extending context window of large language models via positional interpolation," 2023.

[18] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford Alpaca: An Instruction-following LLaMA model," 2023. [Online]. Available: https://github.com/tatsu-lab/stanford_alpaca

[19] T. Computer, "OpenChatKit: An Open Toolkit and Base Model for Dialogue-style Applications," 3 2023. [Online]. Available: https://github.com/togethercomputer/OpenChatKit