

A Comparative Analysis of Static Word Embeddings for Hungarian

Máté Gedeon

Abstract—This paper presents a comprehensive analysis of various static word embeddings for the Hungarian language, including traditional models such as Word2Vec, FastText, as well as static embeddings derived from BERT-based models using different extraction methods.

We evaluate these embeddings on both intrinsic and extrinsic tasks to provide a holistic view of their performance. For intrinsic evaluation, we employ a word analogy task, which assesses the embeddings' ability to capture semantic and syntactic relationships. Our results indicate that traditional static embeddings, particularly FastText, excel in this task, achieving high accuracy and mean reciprocal rank (MRR) scores. Among the BERT-based models, the X2Static method for extracting static embeddings demonstrates superior performance compared to decontextualized and aggregate methods, approaching the effectiveness of traditional static embeddings.

For extrinsic evaluation, we utilize a bidirectional LSTM model to perform Named Entity Recognition (NER) and Part-of-Speech (POS) tagging tasks. The results reveal that embeddings derived from dynamic models, especially those extracted using the X2Static method, outperform purely static embeddings. Notably, ELMo embeddings achieve the highest accuracy in both NER and POS tagging tasks, underscoring the benefits of contextualized representations even when used in a static form.

Our findings highlight the continued relevance of static word embeddings in NLP applications and the potential of advanced extraction methods to enhance the utility of BERT-based models. This piece of research contributes to the understanding of embedding performance in the Hungarian language and provides valuable insights for future developments in the field. The training scripts, evaluation codes, restricted vocabulary, and extracted embeddings will be made publicly available to support further research and reproducibility.

Index Terms—word embeddings, BERT, FastText, Word2Vec, NLP, intrinsic evaluation, extrinsic evaluation

I. INTRODUCTION

Teaching machines to understand human language is a crucial step toward developing intelligent systems. While one-hot encoding can be effective for small-scale classification tasks, it becomes impractical when the aim is to represent thousands of words and their variations. This limitation led to the development of word embeddings—dense vector representations of words in a continuous space. These embeddings not only represent similar words with similar vectors but also are able to capture complex semantic relationships between them.

Bengio et al. [1] introduced feedforward neural networks with a single hidden layer for language modeling in 2003. These models were capable of learning distributed word representations but suffered from significant scalability issues,

limiting their ability to handle large vocabularies. Collobert and Weston [2] addressed this bottleneck in 2008 by refining the training objective and demonstrating the effectiveness of word embeddings pre-trained on large corpora for downstream tasks. Their neural network architecture also served as an inspiration for many subsequent approaches.

Building on the idea that words can be effectively predicted from their context, several word embedding models emerged, including Word2Vec [3], GloVe [4], and FastText [5]. These models proved highly effective in various natural language processing (NLP) tasks. However, their primary limitation lies in their static nature: each word is assigned a single representation, making it impossible to capture context-dependent meanings (e.g., polysemy). This shortcoming paved the way for the development of dynamic, contextualized word embeddings, exemplified by Transformer-based models [6] such as BERT [7], GPT [8], and T5 [9], which generate word representations that adapt based on surrounding text.

Despite the success of dynamic models, static word embeddings remain a viable option for numerous applications due to their significantly lower computational requirements compared to context-dependent embeddings. In this paper, we analyze the performance of multiple pre-trained static word embeddings for Hungarian, as well as static embeddings extracted from BERT-based models, on both intrinsic and extrinsic tasks.

The main contributions of this paper are as follows:

- A comprehensive analysis of the performance of multiple static word embeddings for Hungarian.
- An evaluation of static embeddings extracted from BERT-based models.
- Public release of the code used in the study¹.
- Public release of the embeddings extracted from BERT-based models².

The rest of the paper is organized as follows: Section 2 provides an overview of related work, Section 3 presents the embedding models employed, Section 4 introduces the datasets used in the study, Section 5 describes the experiments conducted, and Section 6 summarizes findings and outlines future directions.

II. RELATED WORK

Multiple methods exist for extracting static embeddings from BERT-based models. The simplest approach involves averaging the token embeddings of a word, when inputting only that word to the model (later referred to as *decontext-*

Máté Gedeon is with the Faculty of Natural Sciences, Budapest University of Technology and Economics, Budapest, Hungary (e-mail: gedeonm01@gmail.com).

DOI: 10.36244/ICJ.2025.2.4

¹https://github.com/geodeonmate/hungarian_static_embeddings

²https://huggingface.co/geodeonmate/static_hungarian_bert

tualized method). Although easy, this method presents an input to the model, which is different from the training data, potentially leading to suboptimal embeddings. Therefore, more sophisticated methods have been proposed. In this piece of research, we employed two such methods: the *aggregate* method [10] and the *X2Static* method [11]. The *aggregate* method pools embeddings of different contexts the word appears in. This presents the model a more natural input than the *decontextualized* method, but as it uses the embedding of the whole context and not just the word itself, it still can lead to suboptimal results. The *X2Static* method, on the other hand, uses a CBOW-inspired static word-embedding approach, leveraging contextual information from a teacher model to generate static embeddings. This method has been shown to outperform the *aggregate* method in multiple tasks.

A Turkish study [12] conducted an in-depth analysis of static word embeddings in the language, which had a substantial impact on the chosen methodology in this study. Their findings were similar, finding the *X2Static* method to be the most effective for extracting static embeddings from BERT-based models.

Several studies have contributed to the evaluation of word embeddings in Hungarian. A 2019 paper [13] assessed the semantic accuracy of word embeddings through word analogy tasks, revealing a substantial performance drop of 50–75% compared to English. The authors attributed this decline to the high morphological variation in Hungarian and the less stable semantic representations that result from it.

Research on Hungarian contextual embeddings has also gained traction. A comparative study [14] evaluating huBERT against multilingual BERT models demonstrated that huBERT outperformed its multilingual counterparts in morphological probing, POS tagging, and NER tasks. On the same tasks, but with a focus on the impact of subword pooling, Ács et al. [15] conducted a comprehensive cross-linguistic analysis of pooling strategies over several languages.

III. EMBEDDING MODELS

A. Word2Vec

Word2Vec [3] is one of the most influential static word embedding models. It is based on the idea that words appearing in similar contexts have similar meanings. Unlike models focused solely on predicting the next word, Word2Vec considers both preceding and succeeding words within a fixed context window (e.g., five words). The model has two architectures: Continuous Bag of Words (CBOW) and Skip-gram. In CBOW, the goal is to predict a target word from its context, disregarding word order. In Skip-gram, the objective is to predict context words based on a given target word.

Several pre-trained embeddings exist for various languages, including Hungarian. In this study, we used pre-trained embeddings developed for the EFNILEX³ project [16], trained on the combined Hungarian Webcorpus [17] and the Hungarian National Corpus [18] with 600 dimensions.

B. FastText

In 2018, Bojanowski et al. [5] introduced FastText, which provides word embeddings for 157 languages. The model was trained on Wikipedia dumps and the Common Crawl. FastText follows a training procedure similar to Word2Vec but incorporates subword information. The Skip-gram model uses character n-grams, assigning a vector representation to each, and constructs word representations by summing the vectors of the character n-grams present in the word. The full word is included to maintain a unique vector for each word. The CBOW model represents words as bags of character n-grams with position-dependent weights to capture positional information. For this study, we used the 300-dimensional Hungarian embeddings published by FastText.

C. SpaCy

SpaCy [19] is a Natural Language Processing library implemented in Python and Cython, supporting over 70 languages. It provides pretrained pipelines for tasks such as tagging, parsing, named entity recognition, and text classification. HuSpaCy [20] is the Hungarian adaptation of SpaCy, including pre-trained embedding models. We employed the 300-dimensional Hungarian CBOW embeddings⁴ provided by HuSpaCy, trained on the Hungarian Webcorpus 2.0.

D. ELMo

ELMo (Embeddings from Language Models) [21] generates contextualized word embeddings using a bidirectional long short-term memory (LSTM) language model. Unlike static embeddings, ELMo captures polysemy and context-dependent meanings, improving performance across multiple NLP tasks. We used the Hungarian ELMo embeddings⁵ provided by HIT-SCIR, trained on a Hungarian Wikipedia dump and the Hungarian portion of the Common Crawl.

E. BERT-based models

BERT (Bidirectional Encoder Representations from Transformers) [7] is a transformer-based model that, like ELMo, generates contextualized word embeddings. It is trained using a masked language model (MLM) objective, where the goal is to predict masked words within a sentence, and a next sentence prediction (NSP) objective, where the aim is to determine whether two sentences are consecutive. As a base model, we used huBERT [22], trained on the Hungarian Webcorpus 2.0. To have a baseline, we also included XLM-RoBERTa (later referred to as *XLM-R*) [23] with the same strategies.

For extracting static embeddings from BERT-based models, we employed the three methods mentioned in *Related Work*. *Decontextualized*, where only the word is inputted to the model, *Aggregate*, which pools embeddings of different contexts the word appears in, and *X2Static*, which uses a CBOW-inspired static word-embedding approach, leveraging contextual information from a teacher model to generate static embeddings. For both the *Aggregate* and *X2Static* methods, we used the same training data, which was a text file, containing

³<http://corpus.nytud.hu/efnilex-vect/>

⁴https://huggingface.co/huspa/hu_vectors_web_lg

⁵<https://github.com/HIT-SCIR/ELMoForManyLangs>

A Comparative Analysis of Static Word Embeddings for Hungarian

one sentence in a line, collected from Hungarian Wikipedia and the Hungarian Webcorpus⁶.

Table I provides an overview of the models used in the study, including their dimensions and original vocabulary sizes.

TABLE I
MODEL SPECIFICATIONS

Model	Dimensions	Original Vocabulary
FastText	300	2,000,000
huBERT	768	-
EFNILEX	600	1,896,111
HuSpaCy	300	1,524,582
XLM-R	768	-
ELMo	1024	-

IV. DATASETS

A. Intrinsic Evaluation Datasets

Intrinsic evaluations assess the quality of embeddings independently, rather than through downstream tasks. They typically measure word similarity or evaluate embeddings on word analogy tasks.

Since no Hungarian word similarity datasets are available, we used the word analogy dataset developed by Makrai [24] for intrinsic evaluation. Inspired by the widely used analogy task introduced by Mikolov et al. [25], this dataset consists of structured word analogy questions in the form of $A:B :: C:D$, where the relationship between words A and B is expected to be analogous to the relationship between words C and D.

The dataset encompasses multiple linguistic and semantic relations, including morphological (e.g., singular-plural forms, verb conjugations), semantic (e.g., country-capital pairs), and syntactic analogies (e.g., grammatical roles).

To make comparisons fair, we restricted the embeddings to the vocabulary of their intersection, arriving at a vocabulary of 256 808 words. Table II provides a comparison of number of questions by category as they were originally, opposed to after the restrictions. There were three categories, where the ratio of the restricted and original questions fell below 90%: capital-world, family, and gram5-present-participle. For example, the capital-world category contained all the country-capital pairs. As smaller countries capital were not present in the vocabulary, the number of questions in this category significantly reduced.

B. Extrinsic Evaluation Datasets

For extrinsic evaluation, we used NYTK-NerKor [26], a Hungarian gold-standard named entity-annotated corpus containing 1 million tokens. This dataset also includes POS tags, making it suitable for both Named Entity Recognition (NER) and Part-of-Speech (POS) tagging tasks. The corpus is divided into training, development, and test sets using an approximately 80-10-10 split. It comprises texts from diverse genres, including fiction, legal documents, news, web sources, and Wikipedia.

The fiction subcorpus includes novels from the Hungarian Electronic Library (MEK) and Project Gutenberg, as well as subtitles from OpenSubtitles. Legal texts originate from

⁶<http://mokk.bme.hu/resources/webcorpus/>

TABLE II
COMPARISON OF ORIGINAL AND RESTRICTED NUMBER OF QUESTIONS BY CATEGORIES

Category	Original	Restricted	Ratio (%)
capital-common-countries	190	190	100.0%
capital-world	13695	5995	43.8%
county-ceter	171	171	100.0%
currency	435	406	93.3%
family	190	136	71.6%
gram1-adjective-to-adverb	780	780	100.0%
gram2-opposite	435	435	100.0%
gram3-comparative	780	780	100.0%
gram4-superlative	780	780	100.0%
gram5-present-participle	780	496	63.6%
gram6-nationality-adjective	820	741	90.4%
gram7-past-tense	780	780	100.0%
gram8-plural-noun	780	780	100.0%
gram9-plural-verb	780	780	100.0%

sources such as the EU Constitution, documents from the European Economic and Social Committee, DGT-Acquis, and JRC-Acquis. The news subcorpus draws from the European Commission's Press Release Database, Global Voices, and the NewsCrawl Corpus. Web texts are sourced from the Hungarian Webcorpus 2.0.

Named entity annotations are divided into four categories: Person, Location, Organization, and Miscellaneous. Although an updated version of the corpus [27] introduces approximately 30 entity types, we used the original version, as these four categories are most commonly utilized in NER tasks.

V. EXPERIMENTS

A. Intrinsic Evaluation

To assess performance on the analogy task, we employed two evaluation metrics. The first metric was overall accuracy, defined as the proportion of correctly answered questions, where the most likely prediction aligns with the correct answer. The second metric was the Mean Reciprocal Rank (MRR), a widely adopted evaluation measure that rewards not only the best prediction but also cases where the correct answer appears among the top-ranked candidates. Given a set of queries, MRR is computed as the average of the reciprocal ranks of the first relevant result for each query:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

where $|Q|$ denotes the total number of queries, and rank_i represents the rank position of the first relevant document for the i th query. To ensure computational feasibility, we restricted analysis to the top 10 ranked answers per query, assigning a score of 0 when the correct answer was absent. Given that different categories contained varying numbers of questions, we also report the average MRR across categories along with the overall MRR (which is the weighted sum).

For static embedding extraction from BERT-based models, we utilized three previously described methods: *decontextualized*, *aggregate*, and *X2Static*. These are indicated by subscripts in the model names. For instance, $\text{huBERT}_{\text{de}}$ refers

to static embeddings extracted using the *decontextualized* method, while huBERT_{agg} and huBERT_{x2} correspond to the *aggregate* and *X2Static* methods, respectively.

Table III presents the results of the analogy tasks. FastText exhibited superior performance, achieving an overall accuracy of 71% and an MRR score of 0.77. The second-best model, huBERT, attained 49% accuracy and an MRR of 0.58. The remaining models formed two distinct groups based on overall accuracy. The first group, comprising EFNILEX, HuSpaCy, and XLM-R_{x2}, yielded an MRR of approximately 0.46 and an accuracy of around 38%. The second group, including huBERT_{agg}, XLM-R_{agg}, ELMo, and huBERT_{de}, exhibited an MRR around 0.23 with an accuracy near 18%. XLM-R_{de} performed the worst, with both metrics falling below 2%, indicating that the *decontextualized* embedding extraction was unsuitable for this task.

Analyzing overall accuracy, we observe that among the top four models are the three static embedding models, along with only one BERT-based model. This suggests that static embeddings remain competitive for intrinsic tasks despite their lack of contextual adaptability compared to transformer-based models. Notably, huBERT_{x2} performed comparably to static embeddings, indicating that the *X2Static* method is a promising technique for deriving static representations from BERT-based models.

The average MRR scores follow a similar pattern to overall accuracy but reveal a more nuanced ranking, as it considers every category equally important, regardless of its size. While the top five models remain consistent, the performance gaps narrow. This discrepancy is likely due to category imbalances, where some categories contain significantly more questions than others. Among the lower-performing models, ordering shifts: ELMo surpasses both *aggregate* models, while huBERT_{de} outperforms huBERT_{agg}.

Examining the performance of the three embedding extraction methods in terms of overall accuracy, a clear trend emerges. The *decontextualized* method consistently underperforms in both huBERT and XLM-R. The *aggregate* method offers a marginal improvement over *decontextualized* in huBERT, yet it demonstrates a substantial boost in XLM-R. This improvement may stem from XLM-R's multilingual nature, where isolated Hungarian words may provide insufficient input, whereas contextualized sentences enable better representations. The *X2Static* method outperforms both alternatives across both models, approaching the effectiveness of dedicated static embeddings. However, when considering average MRR scores, while *X2Static* remains the best performer, huBERT_{de} surpasses huBERT_{agg}.

Category-level results reveal substantial variation in model performance. FastText dominates, achieving the highest scores in 7 out of 14 categories, with the largest margins observed in *capital-common-countries*, *capital-world*, and *gram2-opposite*. HuBERT prevails in three categories (*family*, *gram9-plural-verb*, and *gram7-past-tense*), drawing with EFNILEX in the latter. EFNILEX also leads in *gram4-superlative*. HuSpaCy stands out in *gram5-present-participle*, significantly outperforming other models. Notably, this category experienced the most substantial question reduction

(37%) due to vocabulary constraints. HuSpaCy achieved a 72% MRR score in this category, despite its average MRR of 0.56.

XLM-R_{x2} demonstrates strong performance, leading in the *gram1-adjective-to-adverb* category and slightly outperforming others in *gram3-comparative*. An interesting observation is that while ELMo and huBERT_{de} exhibit similar overall performance, their category-level results diverge significantly, with each model excelling in different areas. In total, static embeddings win 10 of 14 categories, BERT-based models obtain three victories, and one category results in a tie. These findings underscore that category-level performance can vary considerably, making overall scores an incomplete representation of a models strengths.

When analyzing embedding extraction techniques for huBERT at the category level, *X2Static* consistently outperforms the other methods in all but one category (*gram8-plural-noun*), where the *decontextualized* method proves to be superior. Comparing *aggregate* and *decontextualized* approaches, results are less conclusive. For huBERT, *aggregate* excels in only four categories, while *decontextualized* leads in nine, with one tie. XLM-R follows a more distinct hierarchy, with *X2Static* outperforming all other methods, *aggregate* ranking second, and *decontextualized* consistently underperforming.

These results collectively highlight the effectiveness of static embeddings for analogy tasks, the viability of the *X2Static* approach for extracting static embeddings from transformer-based models, and the varying impact of embedding extraction techniques across different linguistic categories.

B. Extrinsic Evaluation

For the extrinsic evaluation, we employed a single-layer bidirectional LSTM with a dropout rate of 0.5. The bidirectional LSTM was chosen for its ability to capture contextual information from both directions, which is particularly beneficial for tasks such as Named Entity Recognition (NER) and Part-of-Speech (POS) tagging. To assess the performance of the embeddings, we experimented with varying hidden sizes (1, 2, 4, 8, 16, 32, and 64). While increasing the hidden size generally enhances model performance, the objective was to evaluate the embeddings themselves, including their behavior under constrained settings.

To handle out-of-vocabulary words in the training data, we represented them using vectors sampled from a normal distribution with a mean of 0 and a standard deviation of 0.6 ($\mathcal{N}(0, 0.6)$). The model architecture concluded with a softmax activation function. Training was conducted for five epochs with a batch size of 32, utilizing the Adam optimizer with its default parameter settings. Categorical cross-entropy was employed as the loss function.

Table IV summarizes the performance of the NER models on the test set. As the primary objective was to assess the quality of the embeddings rather than to achieve state-of-the-art results, accuracy was selected as the evaluation metric. While suitable for the purposes of our comparative analysis, it should be noted that this metric is not directly comparable to those reported by leading models in the field, which are typically evaluated using the F1 score [28].

A Comparative Analysis of Static Word Embeddings for Hungarian

TABLE III
COMPARISON OF PERFORMANCE ACROSS MODELS (MRR)

Category	FastText	huBERT _{x2}	EFNILEX	HuSpaCy	XLM-R _{x2}	huBERT _{agg}	XLM-R _{agg}	ELMo	huBERT _{de}	XLM-R _{de}
capital-common-countries	0.77	0.58	0.45	0.44	0.40	0.25	0.26	0.09	0.17	0.01
capital-world	0.83	0.50	0.28	0.25	0.30	0.23	0.17	0.03	0.08	0.00
county-center	0.88	0.76	0.31	0.47	0.25	0.18	0.07	0.09	0.24	0.00
currency	0.31	0.10	0.19	0.15	0.09	0.07	0.09	0.12	0.07	0.00
family	0.66	0.67	0.40	0.59	0.46	0.22	0.25	0.33	0.30	0.05
gram1-adjective-to-adverb	0.63	0.59	0.37	0.61	0.78	0.20	0.26	0.13	0.25	0.07
gram2-opposite	0.43	0.16	0.29	0.24	0.17	0.01	0.04	0.10	0.07	0.01
gram3-comparative	0.76	0.81	0.75	0.74	0.82	0.30	0.39	0.47	0.38	0.10
gram4-superlative	0.68	0.63	0.72	0.59	0.29	0.21	0.19	0.27	0.22	0.00
gram5-present-participle	0.55	0.17	0.11	0.72	0.12	0.02	0.02	0.01	0.15	0.00
gram6-nationality-adjective	0.91	0.87	0.68	0.61	0.64	0.37	0.37	0.09	0.22	0.01
gram7-past-tense	0.82	0.95	0.95	0.86	0.91	0.18	0.17	0.82	0.49	0.01
gram8-plural-noun	0.77	0.61	0.66	0.68	0.66	0.28	0.34	0.67	0.70	0.06
gram9-plural-verb	0.94	0.97	0.95	0.87	0.87	0.67	0.68	0.86	0.55	0.02
Average MRR	0.71	0.60	0.51	0.56	0.48	0.23	0.24	0.29	0.28	0.02
Overall MRR	0.77	0.58	0.46	0.46	0.45	0.24	0.23	0.22	0.22	0.02
Overall accuracy	0.71	0.49	0.39	0.38	0.37	0.18	0.18	0.18	0.17	0.01

TABLE IV
MODEL PERFORMANCE ACROSS HIDDEN SIZES FOR NER (%)

Model Name	Hidden Size						
	1	2	4	8	16	32	64
ELMo	94.81	96.11	96.44	97.01	97.39	97.54	97.62
huBERT _{x2}	93.55	95.42	96.02	96.68	97.24	97.40	97.49
RoBERTa _{x2}	94.79	95.48	96.16	96.55	97.20	97.33	97.44
huBERT _{de}	94.89	94.12	96.22	96.76	97.16	97.24	97.38
HuSpaCy	94.16	94.71	95.84	96.23	96.54	96.92	97.13
huBERT _{agg}	94.92	94.97	95.16	95.95	96.61	96.94	97.01
EFNILEX	94.48	94.86	95.49	96.09	96.46	96.71	96.80
FastText	94.59	94.89	95.62	96.32	96.56	96.68	96.78
XLM-R _{agg}	85.39	94.67	94.37	95.07	95.12	95.30	95.68
XLM-R _{de}	94.79	88.59	94.97	95.03	95.10	95.22	95.37

As shown in Table IV, most models demonstrate a consistent improvement in performance with increasing hidden size. However, exceptions to this trend include XLM-R_{agg} (between hidden sizes 2 and 4), as well as huBERT_{de} and XLM-R_{de} (between sizes 1 and 2), with the latter experiencing a particularly sharp decline. To complement the tabular data, Figure 1 offers a visual summary that aids intuitive comparison, albeit with a slight trade-off in numerical precision. For clarity, hidden sizes 1 and 2 were excluded from the plot due to their disproportionate effect on the plot scale, which hindered the visibility of differences among the larger hidden sizes.

Regarding overall performance, XLM-R_{agg} and XLM-R_{de} were the lowest-performing models, indicating their relative inefficacy for this task. In contrast—and somewhat unexpectedly—ELMo emerges as the top-performing model across all hidden sizes of two and above, despite its weaker results in the analogy task. It may come from the fact, that ELMo is originally trained using a bidirectional LSTM, similarly to the model used in the extrinsic evaluations. For larger hidden sizes, it is followed by the two X2Static models, huBERT_{x2} and XLM-R_{x2}, which maintain close performance levels but exchange rankings depending on the hidden size. Notably, at smaller hidden sizes, huBERT_{de} outperforms both X2Static models. The remaining four models—HuSpaCy, huBERT_{agg}, EFNILEX, and FastText—consistently occupy the middle range, positioned between the top four and the bottom two, with their relative rankings fluctuating based on hidden size.

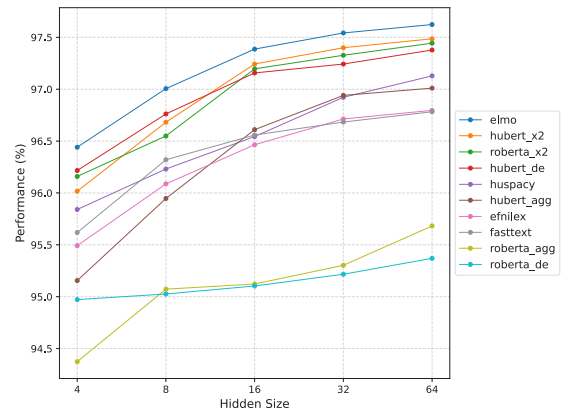


Fig. 1: Model performance across hidden sizes for NER (%)

In terms of extraction methodologies, the X2Static approach remains the most effective, consistent with the intrinsic task results. HuBERT performs better in its *decontextualized* configuration compared to its *aggregate* variant in most cases, likely due to the aggregate model incorporating the full contextual embedding rather than isolating word-level information. XLM-R exhibits an opposite trend, albeit with a less pronounced difference than in the analogy task.

These findings suggest that the top-performing four models are all derived from dynamic embeddings, highlighting their advantage for this task. Nevertheless, purely static embeddings are not far behind in performance, demonstrating competitiveness. Notably, huBERT_{agg} is the only BERT-based model to be consistently outperformed—depending on the hidden size—by static embedding models.

Table V summarizes the numerical results for the POS tagging task, while Figure 2 offers a visual depiction of model performance. As with Figure 1, hidden size 1 is excluded to preserve visual clarity and enable more meaningful comparison across the remaining configurations.

Among the evaluated models, XLM-R_{agg} and XLM-R_{de} exhibited the lowest performance, while huBERT_{agg} outperformed them but remained behind the rest. Consistently,

TABLE V
PERFORMANCE OF VARIOUS MODELS ACROSS DIFFERENT
HIDDEN SIZES IN POS TAGGING (%)

Model Name	Hidden Size						
	1	2	4	8	16	32	64
ELMo	45.69	83.94	88.30	90.56	93.00	94.00	94.58
huBERT _{x2}	60.40	73.06	84.59	91.34	92.50	93.38	93.95
RoBERTa _{x2}	40.89	72.65	87.38	90.21	92.36	93.46	93.76
HuSpaCy	51.20	69.16	83.09	89.31	91.26	92.47	93.33
huBERT _{de}	15.53	72.69	86.80	88.86	91.05	92.98	93.17
EFNILEX	38.15	67.11	78.81	85.92	87.91	89.38	90.48
FastText	46.09	69.46	80.01	86.09	88.33	89.47	89.99
huBERT _{agg}	25.87	48.99	65.64	76.76	82.40	85.60	87.39
XLM-R _{de}	32.19	45.43	54.20	60.57	67.95	71.20	78.37
XLM-R _{agg}	15.63	35.41	39.23	49.14	55.57	53.53	52.57

ELMo achieved the highest accuracy across most hidden sizes, followed closely by the two *X2Static* models, huBERT_{x2} and XLM-R_{x2}.

The remaining models can be grouped into two performance tiers. HuBERT_{de} and HuSpaCy formed the stronger pair, demonstrating similar results. EFNILEX and FastText constituted the second tier, with FastText maintaining a slight edge in configurations with less than 64 hidden sizes.

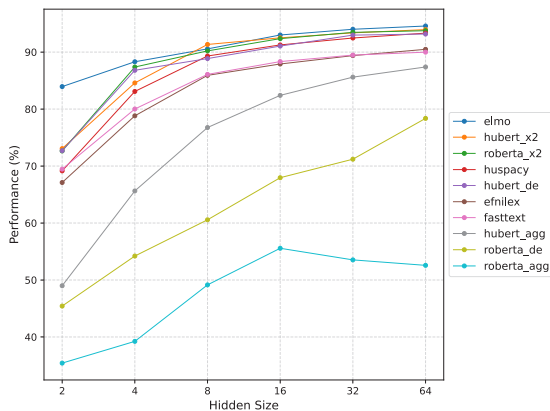


Fig. 2: Performance of various models across different hidden sizes in POS tagging (%)

In alignment with findings from other experiments, the *X2Static* method emerged as the best-performing approach. Notably, both huBERT and XLM-R achieved superior results when employing the *decontextualized* method rather than the *aggregate* one.

As observed in the Named Entity Recognition (NER) task, static embeddings derived from dynamic models consistently outperformed purely static models. Despite this trend, the gap in performance remained relatively small, reinforcing the effectiveness of static embeddings in POS tagging tasks.

VI. CONCLUSION AND FUTURE WORK

In this paper, we conducted a comprehensive analysis of various static word embeddings for Hungarian, alongside static embeddings derived from BERT-based models, evaluating them on both intrinsic and extrinsic tasks. To ensure a fair comparison, all models were restricted to a common vocabulary.

For intrinsic evaluation, we employed an analogy task, where FastText demonstrated superior performance, achieving an overall accuracy of 71% and a Mean Reciprocal Rank (MRR) score of 0.77. Among the BERT-based models, huBERT_{x2} emerged as the best performer, with an accuracy of 49% and an MRR score of 0.58. Notably, the *X2Static* method for extracting static embeddings from BERT-based models outperformed both the *decontextualized* and *aggregate* methods, even rivaling traditional static embeddings in intrinsic evaluations.

For extrinsic evaluation, we utilized a single-layer bidirectional LSTM with varying hidden sizes to assess the effectiveness of the embeddings in downstream tasks. The ELMo embeddings consistently outperformed other models in both Named Entity Recognition (NER) and Part-of-Speech (POS) tagging tasks. The *X2Static* method remained the most effective for extracting static embeddings from BERT-based models, while static embeddings derived from dynamic models outperformed purely static models in both tasks.

This piece of research paves the way for multiple avenues of future exploration. A key direction would be the development of a new intrinsic evaluation dataset for Hungarian, as the analogy question dataset remains the sole benchmark for such evaluations. Additionally, investigating the impact of dimensionality reduction on model performance could yield insights into the trade-offs between efficiency and accuracy, as the models analyzed in this study were trained with varying dimensional settings. Furthermore, the methodology employed here could be extended to evaluate other Hungarian BERT-based models, broadening the scope of comparative analyses. Another significant area of investigation is the exploration of Hungarian GPT-based models ([29], [30]), which were not included in this study due to the lack of a clear methodology for extracting static embeddings from such architectures.

In the spirit of open science, the training scripts, evaluation codes, restricted vocabulary, and the extracted huBERT_{x2} embeddings are made publicly available, facilitating further research and reproducibility in the field.

REFERENCES

- [1] Yoshua Bengio et al. "A neural probabilistic language model". In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 1137–1155. ISSN: 1532-4435. **doi:** 10.5555/944919.944966.
- [2] Ronan Collobert and Jason Weston. "A unified architecture for natural language processing: deep neural networks with multitask learning". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, 2008, pp. 160–167. ISBN: 9781605582054. **doi:** 10.1145/1390156.1390177. URL: <https://doi.org/10.1145/1390156.1390177>.
- [3] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. **doi:** 10.48550/arXiv.1301.3781. arXiv: 1301.3781 [cs.CL]. URL: <https://arxiv.org/abs/1301.3781>
- [4] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. **doi:** 10.3115/v1/D14-1162. URL: <https://aclanthology.org/D14-1162/>.

A Comparative Analysis of Static Word Embeddings for Hungarian

- [5] Piotr Bojanowski et al. "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5 (July 2016). **doi:** 10.1162/tacl_a_00051.
- [6] Ashish Vaswani et al. "Attention is all you need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17*. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964. **doi:** 10.5555/3295222.3295349.
- [7] Jacob Devlin et al. "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)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. **doi:** 10.18653/v1/N19-1423. URL: <https://aclanthology.org/N19-1423/>.
- [8] Alec Radford et al. "Improving language understanding by generative pre-training". In: (2018).
- [9] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. Oct. 2019. **doi:** 10.48550/arXiv.1910.10683.
- [10] Rishi Bommasani, Kelly Davis, and Claire Cardie. "Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 4758–4781. **doi:** 10.18653/v1/2020.acl-main.431. URL: <https://aclanthology.org/2020.acl-main.431/>.
- [11] Prakhar Gupta and Martin Jaggi. "Obtaining Better Static Word Embeddings Using Contextual Embedding Models". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong et al. Online: Association for Computational Linguistics, Aug. 2021, pp. 5241–5253. **doi:** 10.18653/v1/2021.acl-long.408. URL: <https://aclanthology.org/2021.acl-long.408/>.
- [12] Karahan Sarıtaş, Cahid Arda Öz, and Tunga Güngör. "A comprehensive analysis of static word embeddings for Turkish". In: *Expert Systems with Applications* 252 (2024). ISSN: 0957-4174. **doi:** 10.1016/j.eswa.2024.124123. URL: <http://dx.doi.org/10.1016/j.eswa.2024.124123>.
- [13] Bálint Döbrösy et al. "Investigating Sub-Word Embedding Strategies for the Morphologically Rich and Free Phrase-Order Hungarian". In: *Proceedings of the 4th Workshop on Representation Learning for NLP (ReplNLP-2019)*. Ed. by Isabelle Augenstein et al. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 187–193. **doi:** 10.18653/v1/W19-4321. URL: <https://aclanthology.org/W19-4321/>.
- [14] Judit Ács et al. *Evaluating Contextualized Language Models for Hungarian*. Feb. 2021. **doi:** 10.48550/arXiv.2102.10848.
- [15] Judit Ács, Ákos Kádár, and Andras Kornai. "Subword Pooling Makes a Difference". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty. Online: Association for Computational Linguistics, Apr. 2021, pp. 2284–2295. **doi:** 10.18653/v1/2021.eacl-main.194. URL: <https://aclanthology.org/2021.eacl-main.194/>.
- [16] Márton Makrai. "Filtering Wiktionary Triangles by Linear Mbetween Distributed Word Models". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Ed. by Nicoletta Calzolari et al. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 2766–2770. URL: <https://aclanthology.org/L16-1439/>.
- [17] Péter Halácsy et al. "Creating Open Language Resources for Hungarian". In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Ed. by Maria Teresa Lino et al. Lisbon, Portugal: European Language Resources Association (ELRA), May 2004. URL: <https://aclanthology.org/L04-1320/>.
- [18] Tamás Váradi. "The Hungarian National Corpus". In: *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*. Ed. by Manuel González Rodríguez and Carmen Paz Suarez Araujo. Las Palmas, Canary Islands - Spain: European Language Resources Association (ELRA), May 2002. URL: <https://aclanthology.org/L02-1217/>.
- [19] Matthew Honnibal et al. "spaCy: Industrial-strength Natural Language Processing in Python". In: (2020). **doi:** 10.5281/zenodo.1212303.
- [20] György Orosz et al. "Advancing Hungarian Text Processing with HuSpaCy: Efficient and Accurate NLP Pipelines". In: *Text, Speech, and Dialogue*. Ed. by Kamil "Ekstein, František Pártl, and Miloslav" Konopík. "Cham": Springer Nature Switzerland", 2023, pp. 58–69. ISBN: 978-3-031-40498-6. **doi:** 10.1007/978-3-031-40498-6_6.
- [21] Matthew E. Peters et al. "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2227–2237. **doi:** 10.18653/v1/N18-1202. URL: <https://aclanthology.org/N18-1202/>.
- [22] Dávid Márk Nemeskey. "Introducing huBERT". In: *XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021)*. Szeged, 2021, TBA.
- [23] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. **doi:** 10.18653/v1/2020.acl-main.747. URL: <https://aclanthology.org/2020.acl-main.747/>.
- [24] Márton Makrai. "Comparison of distributed language models on medium-resourced languages". In: *XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2015)*. Ed. by Attila Tanács, Viktor Varga, and Veronika Vincze. 2015. ISBN: 978-963-306-359-0.
- [25] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. **doi:** 10.48550/arXiv.1301.3781. arXiv: 1301.3781 [cs.CL]. URL: <https://arxiv.org/abs/1301.3781>.
- [26] Eszter Simon and Noémi Vadász. "Introducing NYTKNerKor, A Gold Standard Hungarian Named Entity Annotated Corpus". In: *Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings*. Ed. by Kamil Ekstein, Frantisek Pártl, and Miloslav Konopík. Vol. 12848. Lecture Notes in Computer Science. Springer, 2021, pp. 222–234. **doi:** 10.1007/978-3-030-83527-9_19.
- [27] Attila Novák and Borbála Novák. "NerKor 1.41e". In: *XVIII. Magyar Számítógépes Nyelvészeti Konferencia*. 2022.
- [28] Simon Eszter et al. "Az NYTK-NerKor több szempontú kiértékelése". In: *XVIII. Magyar Számítógépes Nyelvészeti Konferencia*. Szeged: Szegedi Tudományegyetem, 2022, pp. 403–416.
- [29] Zijian Győző Yang et al. "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.
- [30] Kristóf Szentmihályi et al. "Pretraining GPT-style models in Hungarian". In: *Infocommunications Journal* (2025).



Máté Gedeon earned his bachelor's degree in Mathematics from the Budapest University of Technology in 2023. He is currently pursuing an MSc in Mathematics at the same institution, specializing in data science. His primary research interests lie in the fields of Natural Language Processing (NLP) and Automatic Speech Recognition (ASR).