

# GliLem: Leveraging GliNER for Contextualized Lemmatization in Estonian

**Aleksei Dorkin**

Institute of Computer Science  
University of Tartu  
aleksei.dorkin@ut.ee

**Kairit Sirts**

Institute of Computer Science  
University of Tartu  
kairit.sirts@ut.ee

## Abstract

We present GliLem—a novel hybrid lemmatization system for Estonian that enhances the highly accurate rule-based morphological analyzer Vabamorf with an external disambiguation module based on GliNER—an open vocabulary NER model that is able to match text spans with text labels in natural language. We leverage the flexibility of a pre-trained GliNER model to improve the lemmatization accuracy of Vabamorf by 10% compared to its original disambiguation module and achieve an improvement over the token classification-based baseline. To measure the impact of improvements in lemmatization accuracy on the information retrieval downstream task, we first created an information retrieval dataset for Estonian by automatically translating the DBpedia-Entity dataset from English. We benchmark several token normalization approaches, including lemmatization, on the created dataset using the BM25 algorithm. We observe a substantial improvement in IR metrics when using lemmatization over simplistic stemming. The benefits of improving lemma disambiguation accuracy manifest in small but consistent improvement in the IR recall measure, especially in the setting of high  $k$ .<sup>1</sup>

## 1 Introduction

Lemmatization plays an important role in natural language processing by reducing words to their base or dictionary forms, known as lemmas. This process is especially crucial for morphologically rich languages such as Estonian, where words can

exhibit a multitude of inflected forms. Effective lemmatization enhances various downstream NLP tasks, including information retrieval based on lexical search and text analysis. Although dense vector retrieval is gaining traction in information retrieval, lexical search methods remain highly relevant, particularly in modern hybrid systems. Lexical search excels as a first-stage retriever due to its efficiency with inverted indices, and provides reliable exact term matching that dense retrievers may miss (Gao et al., 2021). Recent research demonstrates that lexical and dense retrieval are complementary, lexical matching providing a strong foundation for precise word-level matches, while dense retrieval captures semantic relationships and handles vocabulary mismatches. The complementary nature of these approaches has led to state-of-the-art hybrid systems that outperform either method alone (Lee et al., 2023).

Vabamorf (Kaalep and Vaino, 2001) is a rule-based morphological analyzer for the Estonian language. It provides one or more morphological analysis (including lemma) candidates for each token in a text, where the token can be a word or a punctuation mark. The Vabamorf’s analyzer functionality aims to generate all possible morphological analysis and lemma candidates for each word, regardless of its context. However, in order to find the appropriate analysis together with the lemma in a particular textual context, the analyzer output needs to be disambiguated. Vabamorf employs a built-in Hidden Markov Model (HMM) based disambiguator that can only look at the word’s immediate context to rank the analysis candidates by their likelihood scores. Thus, despite its high precision in generating lemma candidates, Vabamorf’s ability to disambiguate these candidates in context is limited due to its weak representational power.

Previously, Dorkin and Sirts (2023) have shown that, when evaluated on the Estonian Universal

<sup>1</sup>A demo of the system is available at <https://huggingface.co/spaces/adorkin/GliLem>

Dependencies corpus, Vabamorf’s disambiguation abilities reach to ca 89% for lemmatization. However, when evaluated in the oracle mode, where a prediction is considered correct if the true lemma appears among the candidates, it achieves an accuracy above 99%.<sup>2</sup> This significant difference highlights the limitations of the Vabamorf’s current disambiguator and underscores the need for improving its disambiguation component.

Recent methods to neural lemmatization generally follow two approaches: pattern-based token classification (Straka, 2018; Straka et al., 2019) and generative modeling (Kanerva et al., 2018, 2021). The pattern-based approach predicts for each word a transformation pattern that can be used to transform the word token into corresponding lemma. When built on top of contemporary BERT-based encoders, the pattern-based lemmatizer makes use of the contextual token representations directly to make the prediction. The generative approach uses a character-based sequence-to-sequence model to generate the lemma conditioned on the word, relying on disambiguated morphological information as context. While both of these approaches have shown good results on Estonian (Dorkin and Sirts, 2023), neither of them is well suited for developing a new disambiguator for Vabamorf. First, the pattern-based token classification approach operates with a limited pattern vocabulary extracted from a training set and cannot handle previously unseen patterns that may be output by Vabamorf. Secondly, the generative model already assumes the presence of disambiguated morphological analyses making the disambiguation problem circular.

Recently, an open vocabulary model GliNER for Named Entity Recognition (NER) was proposed by Zaratiana et al. (2024) which can be used to match arbitrary text labels with input text spans. In the lemmatizer disambiguation setting, the GliNER approach can be used to match the transformation patterns extracted from Vabamorf analysis candidates to the spans of sub-word tokens making up words in the text, making it suitable for scoring a limited number of lemma candidates for each word.

Our first aim in this paper is to investigate

---

<sup>2</sup>For instance, if Vabamorf outputs three distinct lemma candidates for a given token, the oracle considers the prediction correct if one of these candidates is correct. This approach is unusable in a practical scenario, because the predictions have to be disambiguated.

whether GliNER method can be used to disambiguate the Vabamorf’s lemma candidates. For that, we modify the GliNER implementation to predict the transformation patterns of Vabamorf’s generated lemma candidates, using the Estonian Universal Dependencies corpus (Zeman et al., 2023) for training. We find that using this approach boosts the disambiguation accuracy from the HMMs 89% to 97.7%, significantly narrowing the gap between the disambiguator and the oracle.

Our second research question examines the impact of the improved lemma disambiguation accuracy on a downstream information retrieval (IR) task. Due to the lack of suitable Estonian datasets, we first translate the English DBpedia-entity dataset (Hasibi et al., 2017) into Estonian, employing the NLLB translation model (NLLB Team et al., 2022). We compare the performance of stemming, Vabamorf HMM-disambiguated lemmatization, and Vabamorf GliNER-disambiguated lemmatization in a BM25 retrieval setup. The results indicate ca 10% improvement in retrieval metrics when using Vabamorf lemmatization over stemming, with an additional 1% gain achieved through GliNER-enhanced disambiguation.

Overall, our contributions in this paper are threefold:

1. We implement a new neural disambiguator based on an open-vocabulary span-labeling method for the Estonian rule-based morphological analyzer Vabamorf (henceforth referred to as GliLem) and show that it considerably improves the lemmatization results over the existing HMM-based disambiguator.
2. We produce and release the first IR dataset for Estonian by machine translating the English DBpedia-entity dataset.<sup>3</sup>
3. We demonstrate the efficacy of the proper lemmatization over stemming for the IR task in Estonian, showing also that improved disambiguation translates into up to 1% improvement in the IR metrics.

## 2 Vabamorf and GliNER

In this section, we first give an overview of both the Estonian morphological analyzer Vabamorf and the open-vocabulary NER model GliNER.

---

<sup>3</sup><https://huggingface.co/datasets/adorkin/dbpedia-entity-est>

## 2.1 Vabamorf

Vabamorf (Kaalep and Vaino, 2001) is a comprehensive, rule-based morphological analyzer specifically developed for the Estonian language. It leverages extensive morphological rules to generate all possible morphological analyses, including lemma candidates, for each analyzed word token. The analyzer accounts for the rich inflectional patterns of Estonian, which include numerous cases, tenses, and degrees of comparison.

Because many Estonian words can have several morphological analyses, Vabamorf includes a built-in HMM-based disambiguator, which aims to rank these candidates based on the contextual likelihood. However, under the HMM formulation, the disambiguation context is very limited, with the analysis of the current word only being dependent on the analysis of the previous word. Therefore, the performance of the HMM-based disambiguator is more than 10% lower than the oracle accuracy that can be obtained on the Estonian UD dataset (Dorkin and Sirts, 2023). We used Vabamorf via EstNLTK, which is a library that provides an API to various Estonian language technology tools (Orasmaa et al., 2016).

## 2.2 GliNER

GliNER (Zaratiana et al., 2024) is an open-vocabulary Named Entity Recognition (NER) model that extends traditional NER capabilities by allowing the labels to be specified in natural language (as opposed to nominal labels represented as integer indices in traditional classification models). Unlike conventional NER models that rely on a fixed set of entity types, GliNER can handle an arbitrary number of labels, making it highly adaptable for tasks requiring flexible label sets.

GliNER is based on an encoder-only BERT-like architecture, which is expanded with span representation and entity representation modules (see Figure 1). The modules are used to produce span and entity embeddings, accordingly. Span and entity embeddings are then used to measure pairwise similarity to identify entities in the input text. Entity types are expressed in natural language and separated from each other with the special [ENT] token. The entity types and input text are separated from each other with a [SEP] token, and they are processed in the model simultaneously in a cross-encoder fashion.

To implement GliNER, Zaratiana et al. (2024)

take an existing pre-trained encoder model as a basis for both the span and entity representation modules, and add two blocks of feed-forward layers on top of the encoder to process the spans and entities separately. Finally, entities are assigned to spans by scoring the similarities between the output representations from both the span and entity modules.

GliNER was pretrained on Pile-NER<sup>4</sup> (Zhou et al., 2023), which is a synthetically annotated large scale NER dataset derived from the Pile corpus (Gao et al., 2020) that has ca 13K distinct entity types. Such pretraining is expected to give the GliNER model an ability to generalize to very different types of labels.

## 3 Adapting GliNER for Vabamorf Lemma Disambiguator

We observe that the GliNER architecture is flexible enough to be used for essentially any kind of token classification task, including part-of-speech tagging and morphological analysis. To be applicable for lemmatization, we adopt the approach proposed by Straka (2018) that expresses each example of *form*  $\rightarrow$  *lemma* as a transformation rule. Each transformation rule comprises the minimal sequence of character-level edits—commonly referred to as a shortest edit script—such as adding, removing, or replacing characters, required to transform a given *form* into its *lemma*. The transformation rules are represented simply as string labels, which are then used in token classification. For specific examples of transformation rules refer to Table 1.

While in theory it would be possible to use lemmas directly as “entities” to be scored instead of transformation rules, that would inflate the number of “entity types” to be learned considerably. Effectively, each token type would have to have its own lemmatization label. Meanwhile, the transformation rules proposed by Straka (2018) are abstract enough to allow for compact representation of similar transformations, and, according to Toporkov and Agerri (2024a), they offer stronger generalization than alternative approaches to shortest edit script generation. For instance, some common rules, such as “do nothing” and “upper case the first character”, are easily applicable to any surface form.

<sup>4</sup><https://huggingface.co/datasets/Universal-NER/Pile-NER-type>

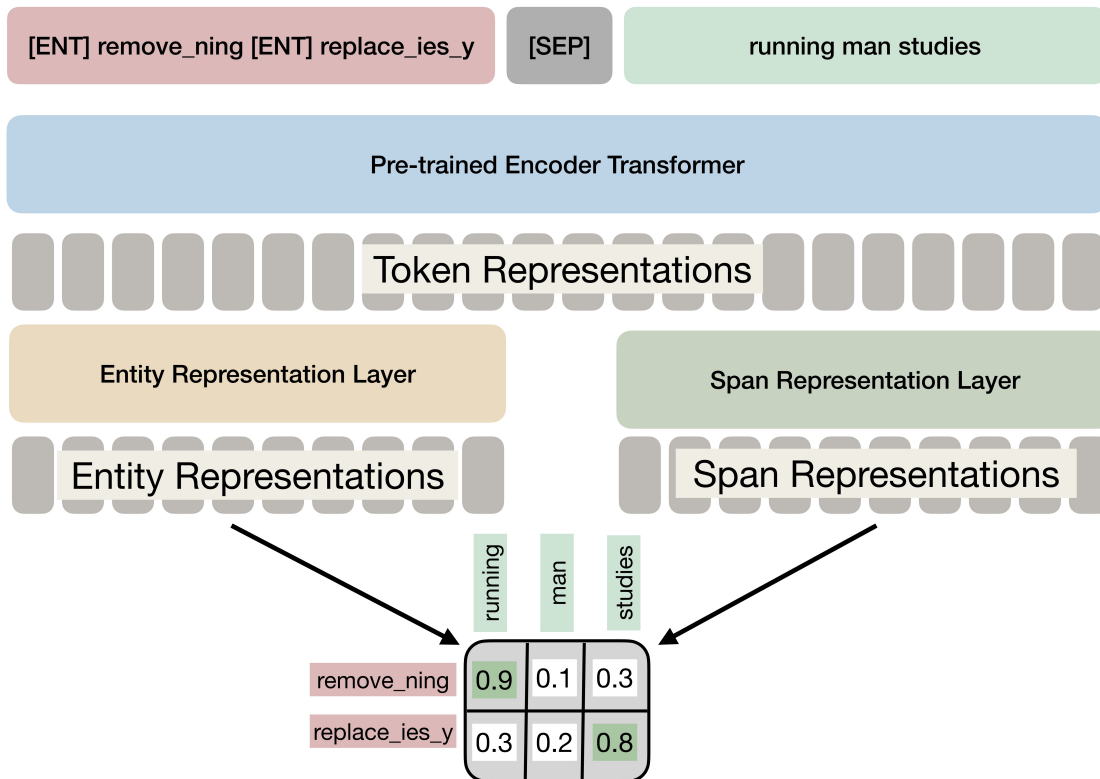


Figure 1: Schematic representation of the GliNER architecture applied to lemmatization.

The total number of unique transformation rules relevant for Estonian is too large to be used as input to GliLem. However, we only aim to score and rank the lemma candidates that the rule-based Vabamorf outputs for each token in the text. This limits the total number of possible “entities” to be scored by the number of tokens in the text, but usually it is much lower than that, mainly because the “do nothing” rule is the most common by far even in morphologically rich Estonian (see Table 1). According to Toporkov and Agerri (2024b), the “do nothing” rule is also the most common rule in diverse languages such as Basque, English, Russian, and Spanish. In Estonian, major contributors to the frequency of this rule, in addition to punctuation marks, include conjunctions, adverbs, some types of adjectives, postpositions, and inflected forms homonymous with the base form.

For each input token Vabamorf outputs at least one morphological analysis (together with lemma). Accordingly, to prepare Vabamorf outputs for disambiguation, a transformation rule is found for each token and all of its lemma candidates. The set of strings representing the obtained unique transformation rules are given as entities in

the GliLem input. GliLem outputs a list of spans, each accompanied by a proposed matching transformation rule and its score. The obtained rules are then applied to the respective spans to get the lemmas. The overall GliLem architecture, i.e., the GliNER architecture applied to lemmatization disambiguation, is schematically represented in Figure 1.

#### 4 Enhancing Disambiguation with GliLem

We implement the GliLem for disambiguating lemma candidates generated by Vabamorf. To assess the effectiveness of the GliLem approach we evaluate the following approaches:

1. Vabamorf lemmatization using the built-in HMM-based disambiguator;
2. Vabamorf lemmatization in the Oracle mode (the prediction is considered correct if the correct lemma is in the proposed non-disambiguated candidates);
3. Pattern-based token classification model for lemmatization;

| %    | Rule        | Description                            |
|------|-------------|--|
| 49.6 | ↓0; d↓      | Do nothing                             |
| 7.0  | ↓0; d↓-     | Remove the last letter                 |
| 4.8  | ↓0; d↓--    | Remove two last letters                |
| 3.7  | ↑0↓↑1; d↓   | Upper case the first letter            |
| 3.3  | ↓0; d↓-+m+a | Replace the last letter with <i>ma</i> |
| 3.2  | ↓0; d↓---   | Remove three last letters              |

Table 1: Top 6 most common transformation rules present in the train split of the EDT dataset.

4. Vabamorf lemma candidates disambiguated with GliLem.

#### 4.1 GliLem training

We conducted experiments using the Estonian Universal Dependencies EDT corpus version 2.14, using the pre-defined splits. During training we do not make use of Vabamorf. Instead, we convert the token/lemma pairs provided in the corpus into respective lemmatization labels (transformation rules) and format the data according to the GliNER convention.

GliNER annotation schema differs from the BIO scheme typically used in the NER task. In GliNER, entire token spans with corresponding labels are used as inputs, and more importantly for our case, non-entities, i.e., the most common “default” class, are not labeled. Correspondingly, we do not use the “do nothing” rule as a label, and instead consider it the default state of the token. That means, we only score cases where the lemma is different from the surface form.

For training the GliLem, we use the GliNER training script provided by the authors<sup>5</sup> using the default parameters and our lemmatization data to train the multilingual version of the pretrained model. The reason for using this model as a base model instead of initializing span and entity modules from scratch is that we expect to benefit from multilingualism of the backbone encoder, and also from the learned span representations of the NER model itself.

#### 4.2 Token Classification Baseline

To contextualize the effect of Vabamorf disambiguation with GliLem, we reproduce the experiments by Dorkin and Sirts (2023) with some differences. We reduce the amount of preprocessing applied to the dataset: we do not lowercase the

<sup>5</sup><https://github.com/urchade/GLiNER/blob/main/train.py>

data and do not remove the derivational symbols present in some lemmas. We also use the more recent UD version (2.10 → 2.14).

The token classification model is a simple, efficient, and computationally cheap baseline to offset the complexity of the GliNER-based approach. For that reason we do not directly reproduce the token classification approach of Dorkin and Sirts (2023), but rather use the adapter-based parameter efficient fine-tuning (Houlsby et al., 2019), which reduces the training time down to minutes.

We do not reproduce the results of the generative character-level transformer model (Wu et al., 2021) that Dorkin and Sirts (2023) reported as the highest scoring approach, because it requires additional morphological annotation as input. Essentially, it needs the data to be disambiguated first which is contradictory to our goals in this work.

#### 4.3 Results

The lemmatization results are shown in Table 2. The GliLem model achieves the lemmatization accuracy of 97.7% on the test, which significantly outperforms Vabamorf’s disambiguator that scores only 89.2% on the same set, demonstrating the efficiency of a more advanced disambiguation approach.

The pattern-based token classification model that does not utilize Vabamorf’s candidates reached 96.2% accuracy. While the difference with the GliLem disambiguation is modest (only ca 1.2% in absolute), it suggests that leveraging Vabamorf’s morphological analysis combined with GliLem’s disambiguation capabilities provides a performance advantage. Moreover, lemmatization accuracy scores are skewed towards higher values due to the majority of corpus tokens requiring no changes to transform the initial word form into the lemma, and that is generally not very difficult for any model to learn and predict.

In the Oracle mode, Vabamorf achieves an accuracy over 99%, showing that the disambiguator module has still room for improvement. However, the gap with the GliLem is less than 2% in absolute that can be hard to close.

### 5 Impact on Information Retrieval

The problem of lemmatization is usually evaluated in isolation, separately from an actual application. While lemmatization can be a useful step in some

| Method                             | Dev                         | Test                        |
|------------------------------------|-----------------------------|-----------------------------|
| Vabamorf                           | 0.878 [0.877, 0.883]        | 0.892 [0.889, 0.895]        |
| Oracle Vabamorf                    | 0.992 [0.992, 0.993]        | 0.993 [0.992, 0.994]        |
| Pattern-based Token Classification | 0.962 [0.960, 0.964]        | 0.966 [0.964, 0.968]        |
| GliLem                             | <b>0.974 [0.973, 0.976]</b> | <b>0.977 [0.975, 0.978]</b> |

Table 2: Bootstrap estimates of the lemmatization accuracy on the Estonian UD EDT dev and test sets with 95% confidence intervals. Oracle Vabamorf considers the prediction correct if the correct lemma appears in non-disambiguated Vabamorf predictions, thus making it unusable in a practical scenario where no labels are available.

realistic scenarios, the impact of the improvement in the lemmatization accuracy on the improvement of the downstream task can be difficult to estimate. To emulate the realistic scenario, we evaluate both the original Vabamorf disambiguator and the GliLem disambiguator in an information retrieval (IR) task. While the IR task is nowadays often addressed with dense vector retrieval, hybrid methods that, as a first step, adopt lexical search methods are still highly relevant. Input normalization via lemmatization is also more important in morphologically complex languages that typically have less resources than English. In particular, there is currently no IR benchmark dataset available in Estonian that would allow to evaluate the effect of different text normalization methods to the IR task. The only previous work in Estonian related to information retrieval that we are aware of is by Dorkin and Sirts (2024). However, this work addressed the problem of retrieving dictionary words based on their definitions using dense IR methods and did not deploy hybrid methods necessitating lexical normalization in the first steps. For this reason, we first translate an existing English information retrieval dataset to the Estonian language.

### 5.1 Dataset Preparation and Translation

DBpedia-Entity v2 (Hasibi et al., 2017) is a test collection for entity search evaluation, consisting of 467 queries with graded relevance judgments for entities from the DBpedia 2015-10 dump. In this work, we refer to the test collection together with the DBpedia dump as DBpedia-Entity. The collection comprises several distinct types of queries:

1. Short, ambiguous queries searching for one particular entity;

2. Information retrieval-style keyword queries;
3. Queries seeking a list of entities;
4. Natural language questions answerable by DBpedia entities.

For each query there is a list of a variable number of documents and their relevance judgments: highly relevant (2), relevant (1), irrelevant (0). Each document in the corpus represents an entity which has an ID, a title in natural language, and a variable length description. The dataset corpus—DBpedia 2015-10 dump—comprises approximately 4.5 million documents. We chose this dataset due to its general domain, the variety of query types it contains, and its focus on retrieving information from a very large collection of documents.

To evaluate the effect of lemmatization accuracy on information retrieval quality in Estonian, we translated the DBpedia-Entity dataset into Estonian using the NLLB (NLLB Team et al., 2022) translation model. We translated both documents and queries using the NLLB 3B,<sup>6</sup> which is the largest available dense version of the NLLB. We adopted the CTranslate2<sup>7</sup> library for efficient translation at large scale. Translating the entire dataset took approximately two days on a single A100 GPU on the University’s High Performance Cluster (University of Tartu, 2018).

At this time, we did not perform any quantitative quality evaluation of the resulting translations. Based on the manual examination of a small sample of examples, we note that while the translation quality is far from perfect, it generally pre-

<sup>6</sup><https://huggingface.co/facebook/nllb-200-3.3B>

<sup>7</sup><https://github.com/OpenNMT/CTranslate2>

serves the meaning well enough to be useful for our benchmark.

## 5.2 Retrieval Experiments

The BM25 algorithm (Robertson et al., 1995) is considered a strong information retrieval baseline to this day even when compared to modern dense retrieval models (Karpukhin et al., 2020; Thakur et al., 2021). BM25 relies on sparse lexical representations of documents and queries, with word-level tokens most commonly used for these representations. The tokens usually undergo additional preprocessing to account for surface form variation. For example, in sparse lexical representation, the present simple and the present participle forms of the word “run” (“run” and “running”) are considered entirely unrelated. That makes it difficult for the user to formulate queries because they have to guess in what form the desired term appears in the documents. For English, applying a stemming algorithm such as PorterStemmer is generally sufficient to deal with this problem.

Meanwhile, stemming algorithms do not perform well for morphologically rich languages like Estonian due to significant variation in stem surface forms in many words. This scenario highlights a practical application of lemmatization—improving the quality of lexical search in such languages. While it is intuitive to expect that lemmatization can help, there are no previous works showing that for the Estonian language. Moreover, it needs to be clarified what effect the additional lemmatization accuracy obtained from better disambiguation of Vabamorf outputs has on information retrieval.

For our experiments we used the recent BM25s library<sup>8</sup> (Lù, 2024) that provides a fast implementation of BM25. For indexing, we used the default parameters and omitted the preprocessing done by the library—we input the corpus preprocessed by us directly.

We preprocessed the Estonian translation of the DBpedia-Entity corpus by applying the following four preprocessing approaches to the dataset documents:<sup>9</sup>

1. Identity (only tokenization is applied);

<sup>8</sup><https://github.com/xhluca/bm25s>

<sup>9</sup>We exclude the token-classification baseline because we are interested in gauging the effect of improved lemmatization disambiguation on IR specifically.

2. Stemming using the Estonian Stemmer available in Apache Lucene;<sup>10</sup>
3. Vabamorf lemmatization with the built-in HMM disambiguation;
4. Vabamorf lemmatization with the GliLem disambiguation.

The output from each preprocessing resulted in each document being represented as a list of tokens, which were then concatenated with whitespace, the expected input format for BM25. The entire corpus was then passed to the indexer implementation. The indexing process took about three minutes, regardless of the preprocessing type.

Finally, we applied the same preprocessing options to the translated queries and, for each query, retrieved **100** most relevant documents from the corpus. Then, we employed relevance judgments from the original DBpedia-entity dataset to obtain the ground truth documents for each query (we selected only the documents deemed relevant or highly relevant) to calculate several retrieval metrics explained in the next section.

## 5.3 Evaluation Metrics

**Success@k** measures whether a user’s information need is satisfied by at least one result in the top **k** retrieved items (Karpukhin et al., 2020; Khatib et al., 2021). It is a coarse-grained metric that does not distinguish how well the user’s information need was satisfied.

**Recall@k** measures what percentage of all relevant items for a query appear within the top **k** retrieved results (Buttcher et al., 2016). The metric is suitable for our case, because only a small proportion of the total number of documents is annotated with relevance judgments and therefore the Recall will be upper bounded only with very small **k** values.<sup>11</sup>

**Mean Average Precision (MAP)@k** measures both the precision and ranking quality of the results up to position **k**, averaged across all queries.

<sup>10</sup>[https://lucene.apache.org/core/8\\_11\\_0/analyzers-common/org/apache/lucene/analysis/et/EstonianAnalyzer.html](https://lucene.apache.org/core/8_11_0/analyzers-common/org/apache/lucene/analysis/et/EstonianAnalyzer.html)

<sup>11</sup>Consider for instance the case where there are 1000 relevant documents per query. In this case, for instance with **k** of 100, the Recall will be upper bounded by 0.1.

| Metric      | Baseline | Stemming | Vabamorf      | GliLem        |
|-------------|----------|----------|---------------|---------------|
| Recall@1    | 0.0269   | 0.0260   | 0.0218        | <b>0.0278</b> |
| Recall@5    | 0.0633   | 0.0627   | 0.0702        | <b>0.0734</b> |
| Recall@100  | 0.2212   | 0.2167   | 0.2831        | <b>0.2935</b> |
| MAP@1       | 0.2077   | 0.2120   | 0.2527        | <b>0.2591</b> |
| MAP@5       | 0.1201   | 0.1312   | <b>0.1596</b> | 0.1577        |
| MAP@100     | 0.0874   | 0.0856   | 0.1057        | <b>0.1115</b> |
| Success@1   | 0.2077   | 0.2120   | 0.2527        | <b>0.2591</b> |
| Success@5   | 0.3704   | 0.4004   | <b>0.4925</b> | 0.4797        |
| Success@100 | 0.6681   | 0.6767   | <b>0.7901</b> | 0.7837        |

Table 3: Retrieval metrics for the proposed token normalization approaches on the translated DBpedia-Entity dataset.

It captures not just whether relevant items were retrieved, but also how high they were ranked, giving more weight to relevant items appearing higher in the results (Buttcher et al., 2016). This metric prioritizes results that group relevant results closer to the top.

#### 5.4 Results and Discussion

The IR performance measures at several  $k$ -s are shown in Table 3. First, we observe that the baseline of using word forms is on the same level with stemming on all measures, which is due to the Apache Lucene stemmer, although Estonian-specific, being very weak for Estonian.

When looking at the setting with  $k$  equal to 1, the Recall does not change considerably, but both MAP and Success rate (that are by definition equal in this setting) improve more than 4% when using lemmatization over stemming, although enhanced disambiguation with GliLem gives only a minor improvement over the default Vabamorf disambiguation.

In the  $k$  equal to 5 setting, Recall improves about 1%, MAP about 3%, and the Success rate, which is the most lenient measure, improves about 9%, when comparing Stemming to lemmatization with Vabamorf. In this setting, only the Recall measure shows a small positive impact of the more complex disambiguation with GliLem over the default Vabamorf disambiguation, while for the MAP and Success rate, the baseline Vabamorf gives better results.

Finally, in the  $k$  equal to 100 setting, when comparing lemmatization to stemming, Recall improves ca 7%, MAP about 2% and Success rate improves about 11%, with GliLem disambigua-

tion showing an additional improvement of ca 1% in both Recall and MAP over the Vabamorf default disambiguation.

We conclude that proper lemmatization can considerably improve IR results compared to stemming. At the same time, even large improvements in lemmatization accuracy, obtained by replacing the simple HMM-based disambiguation component with a more complex GliNER-based disambiguation do not easily translate into significantly better IR results. However, when comparing the baseline Vabamorf with the GliLem disambiguation, we observe a small but consistent improvements in Recall for all values of  $k$ , with the improvement being the most pronounced in the highest  $k$  setting. Using a high  $k$  is typical in hybrid IR systems, where the lexical retrieval is the first step to reduce the number of potentially relevant documents. Thus, the relatively small lemmatization improvement can have a positive effect in the downstream IR task.

Upon manual inspection of the original DBpedia-Entity corpus, we observed that it is somewhat noisy. Some entries have little to no content, while others are comprised of large listings. Many entries have characters from diverse writing systems. This results in additional noise introduced during the imperfect translation process. Moreover, there are translation errors in the translated queries (such as the presence of non-existent words), as well. We believe that the positive effect of the improved lemmatization being somewhat small can be at least partially attributed to these issues. Accordingly, some future work could be dedicated to improving the translated dataset, e.g., manually correcting the



query translations, performing translation quality estimation to redo or filter out low quality document translations, and filtering out entries with no useful content. Consequently, we would expect a larger positive effect of improved lemmatization on a corrected dataset. However, we believe that the noisiness of the dataset affects each approach similarly, and thus the relative ranking between the preprocessing methods remains stable.

We also note that both disambiguation approaches are somewhat computationally intensive. In the current implementation of GliNER, the batch processing does not allow different sets of labels for each example, and thus each example must be processed separately, which makes it difficult to make use of GPU acceleration during inference. The Vabamorf disambiguator, on the other hand, cannot be accelerated at all. Applying both disambiguation approaches to the large corpus of 4.5M documents took over 50 hours for each, using parallelization with approximately 30 concurrent processes on CPU hardware.

## 6 Conclusion

This study demonstrates that integrating an external disambiguation model like GliLem with a rule-based morphological analyzer can substantially improve the accuracy of lemmatization in Estonian. The enhanced lemmatization bridges the accuracy gap caused by the limitations of Vabamorf’s built-in disambiguator. This proves our initial hypothesis that the main weakness of Vabamorf is in fact its inability to correctly select the lemma candidate in context.

Additionally, we estimated the effect of improved lemmatization accuracy on an information retrieval downstream task. Although the precise effect is difficult to estimate due to the noisiness of the original data and additional noise introduced by imperfect machine translation, we observed small consistent improvements in Recall, and especially in the setting with a high  $k$ , suggesting that improved lemmatization might translate into actual improvements in a hybrid information retrieval setting.

## Acknowledgments

This research was supported by the Estonian Research Council Grant PSG721.

## References

- Stefan Buttcher, Charles LA Clarke, and Gordon V Cormack. 2016. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press.
- Aleksei Dorkin and Kairit Sirts. 2023. Comparison of Current Approaches to Lemmatization: A Case Study in Estonian. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 280–285, Tórshavn, Faroe Islands. University of Tartu Library.
- Aleksei Dorkin and Kairit Sirts. 2024. Sõnajaht: Definition Embeddings and Semantic Search for Reverse Dictionary Creation. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (\*SEM 2024)*, pages 410–420, Mexico City, Mexico. Association for Computational Linguistics.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *ArXiv*, abs/2101.00027.
- Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. Complementing Lexical Retrieval with Semantic Residual Embedding. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*, pages 146–160. Springer.
- Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. DBpedia-Entity V2: A Test Collection for Entity Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’17*, pages 1265–1268. ACM.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *International conference on machine learning*, pages 2790–2799. PMLR.
- Heiki-Jaan Kaalep and Tarmo Vaino. 2001. Complete Morphological Analysis in the Linguist’s Toolbox. *Congressus Nonus Internationalis Fenno-Ugristarum Pars V*, pages 9–16.
- Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku Neural Parser Pipeline: An End-to-End System for the CoNLL 2018 Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 133–142.
- Jenna Kanerva, Filip Ginter, and Tapio Salakoski. 2021. Universal Lemmatizer: A sequence-to-sequence model for lemmatizing Universal Depen-

- dencies treebanks. *Natural Language Engineering*, 27(5):545–574.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.
- Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided Supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9.
- Dohyeon Lee, Seung-won Hwang, Kyungjae Lee, Seungtaek Choi, and Sunghyun Park. 2023. On Complementarity Objectives for Hybrid Retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13357–13368.
- Xing Han Lù. 2024. BM25S: Orders of Magnitude Faster Lexical Search via Eager Sparse Scoring. *arXiv preprint arXiv:2407.03618*.
- Siim Orasmaa, Timo Petmanson, Alexander Tkachenko, Sven Laur, and Heiki-Jaan Kaalep. 2016. EstNLTk - NLP Toolkit for Estonian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp*, 109:109.
- Milan Straka. 2018. UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.
- Milan Straka, Jana Straková, and Jan Hajic. 2019. UDPipe at SIGMORPHON 2019: Contextualized Embeddings, Regularization with Morphological Categories, Corpora Merging. *SIGMORPHON 2019*, page 95.
- NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Searley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation. *arXiv preprint arXiv:2207.04672*.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.
- Olia Toporkov and Rodrigo Agerri. 2024a. Evaluating Shortest Edit Script Methods for Contextual Lemmatization. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6451–6463.
- Olia Toporkov and Rodrigo Agerri. 2024b. On the Role of Morphological Information for Contextual Lemmatization. *Computational Linguistics*, 50(1).
- University of Tartu. 2018. UT Rocket.
- Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the Transformer to Character-level Transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.
- Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Mexico City, Mexico. Association for Computational Linguistics.
- Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackeremann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Salih Furkan Akkurt, Gabrielè Aleksandravičiūtė, Ika Alfina, Avner Algom, Khalid Alnajjar, Chiara Alzetta, Erik Andersen, Lene Antonsen, Tatsuya Aoyama, Katya Aplonova, Angelina Aquino, Carolina Aragon, Glyd Aranes, Maria Jesus Aranzabe, Bilge Nas Arıcan, Hórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Katla Ásgeirsdóttir, Deniz Baran Aslan, Cengiz Asmazoğlu, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Mariana Avelãs, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Shabnam Behzad, Kepa Bengoetxea, Ibrahim Benli, Yifat Ben Moshe, Gözde Berk, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė,

Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, António Branco, Kristina Brokaitė, Aljoscha Burchardt, Marisa Campos, Marie Candito, Bernard Caron, Gauthier Caron, Catarina Carvalheiro, Rita Carvalho, Lauren Cassidy, Maria Clara Castro, Sérgio Castro, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Liyanage Chamila, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Daniela Corbetta, Francisco Costa, Marine Courtin, Mihaela Cristescu, Ingerid Løyning Dale, Philemon Daniel, Elizabeth Davidson, Leonel Figueiredo de Alencar, Mathieu Dehouck, Martina de Laurentiis, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Adrian Doyle, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Christian Ebert, Hanne Eckhoff, Masaki Eguchi, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaz Erjavec, Farah Essaidi, Aline Etienne, Wograinne Evelyn, Sidney Facundes, Richárd Farkas, Federica Favero, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Amal Fethi, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Federica Gamba, Marcos Garcia, Moa Gårdenfors, Fabrício Ferraz Gerardi, Kim Gerdes, Luke Gessler, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökürmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Takahiro Harada, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Henning, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Marivel Huerta Mendez, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Olajidé Ishola, Artan Islamaj, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Katharine Jiang, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Ritván Karahóga, Andre Kåsen, Tolga Kayadelen, Sarveswaran Kengatharaiyer, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Alexey Koshevoy, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krish-

namurthy, Sandra Kübler, Adrian Kuqi, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Kris Kyle, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Lauren Levine, Cheuk Ying Li, Josie Li, Keying Li, Yixuan Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Yi-Ju Jessica Lin, Krister Lindén, Yang Janet Liu, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Ilya Makarchuk, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Măranduc, David Mareček, Katrin Marheinecke, Stella Markantonatou, Héctor Martínez Alonso, Lorena Martín Rodríguez, André Martins, Cláudia Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Aaron Miller, Karina Mischenkova, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, AmirHossein Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskiy, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horriacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Hulda Óladóttir, Adedayo Olúokun, Mai Omura, Emeka Onwuegbuzia, Noam Ordan, Petya Osenova, Robert Östling, Lilja Øvreid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Teresa Pacosi, Alessio Palmero Aprosio, Anastasia Panova, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Giulia Pedonese, Angelika Peljak-Łapińska, Siyao Peng, Siyao Logan Peng, Rita Pereira, Sílvia Pereira, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Andrea Peverelli, Jason Phelan, Jussi Piitulainen, Yuval Pinter, Clara Pinto, Tommi A Pirinen, Emily Pitler, Magdalena Plamada, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Robert Pugh, Tina Puolakainen, Sampo Pyysalo, Peng Qi, Andreia Querido, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Joana Ramos, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Arij Riabi, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvalds-

son, Ivan Roksandic, Mykhailo Romanenko, Rudolf Rosa, Valentin Roşca, Davide Rovati, Ben Ronzoyer, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Aleksí Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Marta Sartor, Mitsuya Sasaki, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Maria Shvedova, Janine Siewert, Einar Freyr Sigurðsson, João Silva, Aline Silveira, Natalia Silveira, Sara Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Haukur Barri Símonarson, Kiril Simov, Dmitri Sitchinava, Ted Sither, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Per Erik Solberg, Barbara Sonnenhauser, Shafi Sourov, Rachele Sprugnoli, Vivian Stamou, Steinhór Steingrímsson, Antonio Stella, Abishek Stephen, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Daniel Swanson, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Mirko Tavoni, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Sara Tonelli, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sveinbjörn Hórrðarson, Vilhjálmur Hörsteinsson, Sumire Uematsu, Roman Untilov, Zdeňka Uřešová, Larraitz Uriá, Hans Uszkoreit, Andrius Utka, Elena Vagnoni, Sowmya Vajjala, Socrates Vak, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Uliana Vedenina, Giulia Venturi, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jonathan North Washington, Maximilian Wendt, Paul Widmer, Shira Wigderson, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Yilun Zhu, Anna Zhuravleva, and Rayan Ziane. 2023. Universal dependencies 2.12. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023. UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition. *arXiv preprint arXiv:2308.03279*.