ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 2 (32)

UDC 81'322.3 DOI: https://doi.org/10.30837/2522-9818.2025.2.102

K. SYROTKIN

ACCURACY EVALUATION AND ERROR ANALYSIS OF DEPENDENCY PARSING FOR TEXTS IN UKRAINIAN

The **subject** of our research is the dependency parsing of sentences in the Ukrainian language using the Universal Dependencies framework. The **goal** of the work is to evaluate the accuracy of existing transition-based and graph-based parsing architectures with and without deep word embeddings on the Ukrainian dataset, and to analyze the error profiles of such parsers. The article addresses two **tasks**. One is to evaluate the accuracy of several modern dependency parsing approaches applied to a hand-annotated gold standard dataset, using labeled and unlabeled attachment scores as the metric to evaluate the parsing accuracy. The other task is to analyze and categorize the errors made by standard parsers. Resolving these errors could potentially allow us to build a more accurate parser in the future. Error rate for different categories is compared to the baseline error rate, and statistical significance of such comparison is validated using the chi-square **method**. The key **results** are as follows. For the Ukrainian language, parsing accuracy is greatly increased with the use of deep word embeddings. Transition-based parser with deep word embeddings provides the highest labeled attachment score of 84.66% for the test dataset. For the same parser, higher error rates are associated with non-projectivity of dependencies, higher sentence length and higher distance to head. Also, for pronouns and numerals the error rate for labeled attachment is significantly higher than the baseline, while the unlabeled error rate is at the baseline. **Conclusions:** parsing accuracy for the Ukrainian dataset is sub-par in comparison with other languages, but the overall trend of accuracy improvement with the use of deep word embeddings is consistent with existing research. To improve overall parsing accuracy, we must focus on such problem areas as non-projective dependencies, longer sentences, and greater distance between the head and the dependent. In future work we intend to explore ways to improve parsing accuracy by supplementing neural parsing with other approaches, like formal rules or pre- and post-processing.

Keywords: natural language processing; syntactic parsing; dependency parsing; Universal Dependencies.

1. Introduction

1.1 Relevance of dependency parsing

Dependency parsing is a method of analysis in natural language processing. It involves analyzing the grammatical structure of a sentence by identifying the relationships between words and representing these relationships as a directed graph.

Syntactic analysis is usually considered an important building block used to improve the performance of subsequent tasks.

For example, since named entities are often expressed as noun phrases, the analyzed input can greatly benefit named entity recognition (NER). Part-of-speech tagging (POS) is closely related to dependency parsing. Correct identification of the subject and object of a sentence is extremely useful for dialogue systems, sentiment analysis, and machine translation. Last but not least, syntactic analysis is often used in advanced writing systems and educational systems to check written text for compliance with a set of formal rules.

Given the importance of parsing dependencies, our focus in this work is on the accuracy of parsing.

1.2 Problems in analyzing dependencies in Ukrainian texts

1.2.1 Morphological richness and free word order

One of the main problems inherent in the Ukrainian language is its morphological richness. Morphologically rich languages are those in which grammatical information about the organization of words into syntactic units and the characteristics of syntactic relations are expressed at the word level. Since information about the relationships between syntactic elements is indicated in the form of words, these words can freely change their positions in a sentence. This is called free word order. Information about the grouping of elements can be expressed by reference to their morphological form. Such logical groups of non-contiguous elements are often called discontinuous constituents. In dependency structures, such breaks lead to non-projectivity [1].

As a language with a fairly flexible word order, Ukrainian has a higher proportion of non-projective sentences than, for example, English. The proportion of non-projective sentences in the gold standard corpus of universal dependencies for the Ukrainian language [2] used in this evaluation is 7.6%. For comparison, the Gold Standard Dependency Corpus for English [3] contains

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 2 (32)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

only 2.0% non-projective sentences. An example of non-projective dependency analysis is shown in Figure 1. The arc from *сльози* to *мої* intersects the arc between *пролились* and *там*.

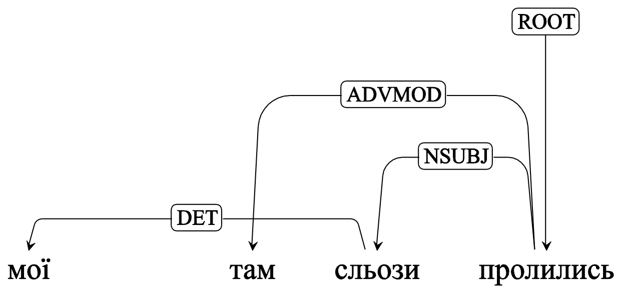

Fig. 1. Example of a non-projective dependency tree

The arc from the head word to the dependent word is called projective if there is a path from the head word to each of the dependent words that are located between the head word and the dependent word in the sentence. A dependency tree is called projective if all the arcs that form it are projective. However, there are many correct constructions that result in non-projective trees, especially in languages with relatively flexible word order [4].

Since basic transition-based parsers do not inherently produce non-projective trees, special measures must be taken to solve this problem. The parser we chose for our evaluation uses a special SWAP transition to eliminate non-projectivity.

1.2.2 Lack of resources

In terms of available tree banks and corpses, Ukrainian is a rather low-resource language. At the time of writing, there is only one available "silver standard" corpse with manually annotated dependencies. Other corpses are available, but they are either specialized, lack dependency annotations, or have machine-annotated dependencies ("silver standard"). The smaller size of the dataset may explain why the parser's accuracy for Ukrainian texts is lower than for other languages (see Table 2).

1.3 Research objectives

1.3.1 Accuracy assessment

In this study, the first objective is to evaluate the accuracy of several modern dependency analysis approaches applied to a manually annotated gold standard dataset. This allows us to establish a baseline for future work, as the ultimate goal of our research is to develop an improved dependency analyzer. To evaluate the accuracy of syntactic analysis, we use the accuracy measures for labeled and unlabeled link analysis (LAS and UAS).

1.3.2 Error analysis

Our second goal is to analyze and classify errors made by standard syntax analyzers, as we believe that tuning the parser to handle language-specific errors is a viable way to improve the accuracy of syntax analysis. Errors that consistently appear in unrelated parser models may indicate a typical error specific to a particular parser architecture or language. Eliminating these errors could potentially allow us to create a more accurate parser in the future.

2. Other works on the topic

Most existing dependency analysis studies are not specific to the Ukrainian language. Instead, universal approaches to syntactic analysis are being developed, studied, and tested in many languages, including Ukrainian.

A brief overview is provided below, but we believe that the arc-hybrid transition-based approach, which uses deep word embeddings for encoding, is the most promising, and we should focus on improving it.

2.1 Brief overview of dependency analysis approaches

Two common approaches to dependency analysis are transition-based and graph-based architectures

The classic transition-based syntax analysis architecture is based on shift-reduce syntax analysis, a paradigm originally developed for analyzing programming languages. In transition-based parsing, we have a stack on which we build the parsing tree, a token buffer for parsing, and a parser that performs actions through a predictor called an oracle. Such parsers are generally very efficient, often having linear time complexity, but their accuracy can suffer due to the accumulation of search errors.

Graph-based dependency parsers search the space of possible trees for a given sentence for a tree (or trees) that maximizes a certain score. These methods encode the search space as directed graphs and use methods from graph theory to search the space for optimal solutions. These analyzers do not suffer from search error accumulation, but their analysis algorithms are typically more complex [4].

Although both architectures can be implemented using formal algorithms, the modern neural approach is

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 2 (32)

now almost exclusively used. Regardless of whether a neural classifier is used for the oracle in transition-based parsing, or a neural algorithm is used for scoring in graph-based parsing, the representation of tokens is a significant factor in the accuracy of parsing. The latest systems use an encoder, typically in the form of a BiLSTM, which provides contextualized representations of input words as input to transition estimation – in transition-based analyzers – or dependency arcs – in graph-based analyzers.

Neural syntactic analyzers rely on vector representations of words as their main input, often in the form of pre-trained embeddings such as word2vec [5]. These methods assign a single static representation to each word and therefore cannot capture context-dependent changes in meaning and syntactic behavior.

In contrast, deep contextualized word representations encode words with respect to the context of the sentence in which they appear. Similar to word embeddings, such models are typically trained for language modeling but produce sentence-level tensors as representations instead of individual vectors. The advantage of such models is that they not only create contextualized representations, but also do so at multiple levels of abstraction, as captured by different layers of the model, and are pre-trained on corpora much larger than typical tree banks. The use of embeddings with information about the global structure of the sentence in the representation of local functions makes transition-based and graph-based parsers practically equivalent in terms of both accuracy and error profile [6].

In addition to better computational complexity, transition-based analyzers offer another significant advantage. Both dependency analysis and sentence segmentation can be performed simultaneously using the same oracle by including an additional "end of sentence" transition. This approach by Honiball and Johnson [7] is used in the spaCy NLP library for Python.

3. Methodology

Our work on evaluating syntactic analyzers is carried out within the Universal Dependencies framework. Universal Dependencies (UD) is a project that develops a cross-linguistically aligned annotation of a tree bank for many languages with the aim of facilitating the development of a multilingual parser, cross-lingual learning, and typological analysis research [8]. UD defines a data format for the analyzed sentences and provides a variety of annotated datasets for different languages.

3.1 Dataset selection

To train and evaluate different parsers, we need a dataset that meets certain requirements. The data should come from various sources, such as news articles, fiction, or social media posts. The data should meet the "gold standard", i.e., be annotated manually by experts. The size of the dataset should be of the same order as the data used by Kulmiziev et al. [6], as we intend to compare our findings with the results of their work.

We have researched the available resources and found several notable Ukrainian corpora that can be downloaded and used for NLP research:

- Uber Text 2.0 [9];
- Ukrainian Brown Corpus, or BRUK [10];
- Ukrainian parliamentary stenograms annotated in Universal Dependencies [11];
- Universal Dependencies gold standard corpus for the Ukrainian language [2].

Although UberText and BRUK have a lot of diverse and balanced data, they lack the dependency annotations we need. Both the parliamentary transcripts and the gold standard corpus have dependency annotations, but the parliamentary transcripts do not have enough diverse data for our use case. The UD gold standard corpus, however, contains 122,000 tokens in 7,000 sentences from fiction, news, articles, Wikipedia, legal documents, letters, publications, and comments.

Thus, model training and evaluation should be performed on the UD gold standard corpus for the Ukrainian language.

3.2 Parser implementation

In this work, we conduct experiments described by Kulmiziev et al. [6] on our Ukrainian dataset. Therefore, we use the same system, UUParser, originally developed by de Lhoneux, Stymne, and Nivre [12]. The UUParser project includes a transition-based and a graph-based parser, both of which can be trained with or without deep word embeddings.

For an input sentence $S = w_{1,\dots,} w_N$, the parser creates a sequence of vectors $W_{1:N}$, where a vector $W_k = x_k \circ BiLSTM\left(c_{1:M}\right)$ represents an input word w_k as a concatenation of a pre-trained word x_k embedding and a character $BiLSTM\left(c_{1:M}\right)$ embedding obtained

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 2 (32) ISSN 2524-2296 (online)

by processing the BiLSTM sequence of the vector's w_k characters $c_{1:M}$. Next, each input element is represented as a BiLSTM vector, $h_k = BiLSTM\left(W_{1:N,k}\right)$.

During transition-based syntactic analysis, BiLSTM vectors are fed into a multilayer perceptron (MLP) to count transitions using an arc-hybrid transition system extended with SWAP transitions to allow the construction of non-projective dependency trees. The evaluation is based on the top three words in the stack and the first word in the buffer, and the input to the MLP includes BiLSTM vectors for these words as well as their leftmost and rightmost dependencies (up to 12 words in total).

In graph-based analysis, BiLSTM vectors are fed into the MLP to evaluate all possible dependency relationships in the arc model, meaning that only vectors corresponding to the header and dependent are part of the input (2 words in total). Then, the syntactic analyzer extracts the maximum spanning tree over the feature matrix using the Chu-Liu-Edmonds algorithm (CLE), which allows building non-projective trees [6].

To evaluate the effectiveness of the parser using deep word embeddings, we use ELMo representations. These differ from traditional word embeddings in that each lexeme is assigned a representation that is a function of the entire input sentence. ELMo uses vectors obtained from a bidirectional LSTM trained with an object-related language model (LM) on a large text corpus. ELMo representations are deep in the sense that they are a function of all internal layers of biLM [13].

These embeddings were chosen because of the availability of a pre-trained ELMo model for the Ukrainian language [14]. More importantly, using this model allows us to make a direct comparison with the data from [6], as it includes accuracy values for both transition-based parsers and graph-based parsers with ELMo embeddings.

In this work, we conduct the experiments described by Kulmiziev et al. [6] on our Ukrainian dataset. Therefore, we use the same system, UUParser, originally developed by de Lhoneux, Stymne, and Nivre [12]. The UUParser project includes a transition-based parser and a graph-based parser, both of which can be trained with or without deep word embeddings.

For an input sentence, the parser creates a sequence of vectors, where a vector represents an input word as a concatenation of a pre-trained word embedding and a character embedding obtained by processing the BiLSTM sequence of the vector's characters. Next, each input element is represented as a BiLSTM vector.

During transition-based syntactic analysis, BiLSTM vectors are fed into a multilayer perceptron (MLP) to count transitions using an arc-hybrid transition system extended with SWAP transitions to allow the construction of non-projective dependency trees. The evaluation is based on the top three words in the stack and the first word in the buffer, and the input to the MLP includes BiLSTM vectors for these words as well as their leftmost and rightmost dependencies (up to 12 words in total).

In graph-based analysis, BiLSTM vectors are fed into the MLP to evaluate all possible dependency relationships in the arc model, meaning that only vectors corresponding to the header and dependent are part of the input (2 words in total). Then, the syntactic analyzer extracts the maximum spanning tree over the feature matrix using the Chu-Liu-Edmonds algorithm (CLE), which allows building non-projective trees [6].

To evaluate the effectiveness of the parser using deep word embeddings, we use ELMo representations. These differ from traditional word embeddings in that each lexeme is assigned a representation that is a function of the entire input sentence. ELMo uses vectors obtained from a bidirectional LSTM trained with an object-related language model (LM) on a large text corpus. ELMo representations are deep in the sense that they are a function of all internal layers of biLM [13].

These embeddings were chosen because of the availability of a pre-trained ELMo model for the Ukrainian language [14]. More importantly, using this model allows us to make a direct comparison with the data from [6], as it includes accuracy scores for both transition-based parsers and graph-based parsers with ELMo embeddings.

3.3 Assessment metrics

To evaluate a dependency analysis system, it is necessary to measure how well it performs on a test set. The simplest approach would be to evaluate how many of the analyzed sentences exactly match the ground truth, but such a metric is too coarse to guide the development process. More detailed metrics are preferable, namely the unlabeled association score (UAS), as described by Eisner [15], and the labeled association score (LAS), the main dependency analysis metric introduced by Neuville, Hall, and Nilsson [16].

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 2 (32)

For each word, the construction of a labeled relation means the identification of the head word and the relation type, for example, NSUBJ or AMOD. Then, the accuracy of labeled relation construction, LAS, is the percentage of tokens for which both the head word and the relation type are correctly identified. The accuracy of unlabeled relations, UAS, is the percentage of tokens for which at least the headword has been correctly identified (see Fig. 2). Of the five relations, the parser incorrectly labeled *Дніпр* as NSUBJ for the word *стогне* and incorrectly identified the relation type between *широкий* and *Дніпр*. Thus, UAS is 4/5=80%, and LAS is 3/5=60%.

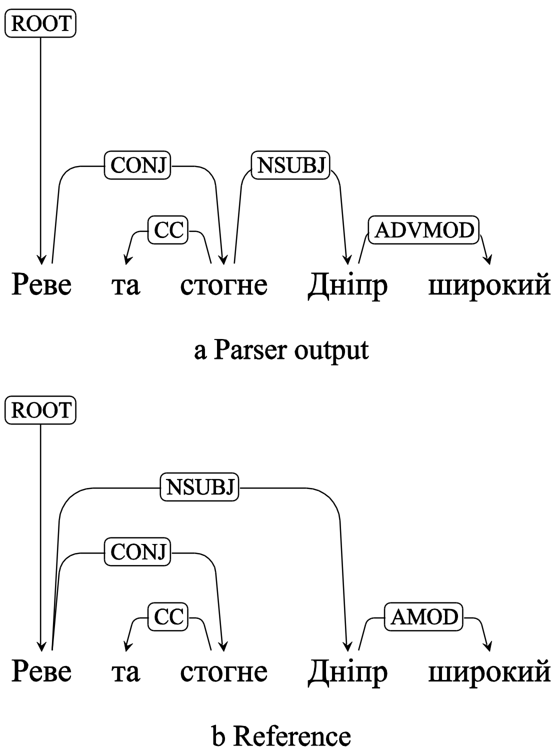

Fig. 2. *Example of LAS and UAS calculation*

There are other metrics that aim to achieve success where LAS falls short, namely for comparing the performance of syntactic analyzers in different languages [17]. However, since Ukrainian is the focus of our work, we will use LAS as our main metric.

3.4 Experiment plan

The dataset is divided into three parts: training (5521 sentences / 93161 dependencies), development (673 sentences / 12622 dependencies), and testing (898 sentences / 17249 dependencies). We train several models using the same dataset but different random initial weights and evaluate their performance.

3.4.1 Accuracy evaluation

We evaluate four different parser architectures: transition-based, graph-based, transition-based with deep embeddings, and graph-based with deep embeddings. For each architecture, the parser is trained on the training dataset for 30 epochs with default parser settings. After each epoch, the model is applied to the development dataset to determine the LAS. The epoch model with the highest LAS value is selected. The training process is repeated three times with different random initial weights, creating three models. Each model is applied to the test dataset to determine the LAS, and the average LAS is calculated.

3.4.2 Error analysis

To analyze the error pattern, we train 100 models based on transitions with ELMo embeddings with different random initial weights.

Error analysis is performed by comparing the error frequency for dependencies grouped by certain features with the baseline error frequency (for the entire test dataset). This analysis takes into account the error frequency in both labeled and unlabeled connections. We investigate the following features: projectivity of the relation, sentence length, distance between the head and dependent words, and part of speech of the dependent word.

For each group, the difference in error rates from the baseline is evaluated for statistical significance using the chi-square test $\left(p < 0.05\right)$.

4. Experiment results

4.1 Overall accuracy and comparison with other languages

The accuracy assessment results are summarized in **Tables 1 and 2**. The assessments for languages other than Ukrainian were obtained by Kulmiziev et al. [6].

Table 1. *LAS by architecture (individual experiments and average)*

Architecture	#1	#2	#3	Avg.
Transition	77.57	78.03	77.80	77.80
Graph	79.34	78.81	78.97	79.04
Transition + ELMo	84.71	84.79	84.47	84.66
Graph + ELMo	83.84	83.37	83.78	83.66

Four parsing systems were analyzed: transition-based (Transition), graph-based (Graph), transition-based

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 2 (32) ISSN 2524-2296 (online)

with deep ELMo embeddings (Transition + ELMo), and graph-based with deep ELMo embeddings (Graph + ELMo). The LAS value was obtained for three independently trained models with different random initial weights (#1, #2, and #3), as well as the average LAS value (Avg.).

Table 2. *Size of the training dataset (sentences) and average LAS for Ukrainian and other languages*

Language	Train	TR	GR	TR+E	GR+E
Ukrainian	**5.5k**	**77.8**	**79.0**	**84.7**	**83.7**
English	12.5k	82.7	83.3	87.0	86.5
Hindi	13.3k	88.4	89.6	91.0	91.2
Italian	13.1k	88.0	88.2	90.9	90.6
Russian	48.8k	88.3	88.0	90.7	90.6
Swedish	4.3k	80.5	81.6	86.9	86.2

For each language, the number of sentences in the training dataset (Train) is shown, as well as the average LAS score for each of the systems: transition-based (TR), graph-based (GR), transition-based with deep ELMo embeddings (TR+E), and graph-based with deep ELMo embeddings (GR+E).

Our results show that without deep embeddings, the graph-based parser offers better accuracy than the transition-based parser. We believe this is due to the high frequency of non-projective parse trees in the Ukrainian language.

The use of ELMo embeddings significantly improves the accuracy of the syntactic analyzer in both transition-based parsers and graph-based analyzers. The accuracy of transition-based and graph-based parsers on the Ukrainian dataset is similar.

Our findings confirm the general trend described in [6]: the use of deep word embeddings significantly improves analysis accuracy and eliminates the gap between graph-based and transition-based parsers.

It should be noted that the accuracy of Ukrainian sentence parsing is low compared to other Indo-European languages, as shown in Table 2.

This is probably due to the fact that the dataset for Ukrainian is relatively small compared to other languages. Nevertheless, the general trend of improvement from the use of embeddings is consistent with other languages.

4.2 Error profile analysis for a transition-based parser with deep word embeddings

4.2.1 Non-projectivity

Conclusions regarding the frequency of errors grouped by projectivity are presented in Table 3. Error levels above the baseline are highlighted in bold. Coefficients with statistically insignificant differences from the baseline are indicated in square brackets.

Table 3. *Error rates for labeled and unlabeled relations grouped by projectivity at the level of a single dependency and at the sentence level*

	Prevalence of feature, %	Unlabeled errors, %	Labeled errors, %
Basic level	–	12.29	15.47
Non-projective dependencies	2.57	**29.88**	**33.71**
Projective dependencies	97.43	11.83	14.99
Projective dependencies in non-representative sentences	9.76	**[12.31]**	[15.44]
Projective dependencies in fully projective sentences	87.67	11.77	14.94

There is a strong correlation between the projectivity of dependencies and the accuracy of parsing. The error rate among the analyzed non-projective dependencies is significantly higher than among projective dependencies: 29.88% vs. 11.83 error rates for unlabeled relations and 33.71% vs. 14.99 for labeled relations.

If a sentence contains a non-projective relation, other projective relations are not associated with higher error rates than relations in fully projective sentences. The error rates for both labeled and unlabeled relations do not differ significantly from the baseline level.

4.2.2 Sentence length

The results for error frequencies grouped by sentence length are shown in **Table 4**. For this analysis, we grouped sentences by length into 8 quantiles. Error levels above the baseline level are highlighted in bold. Coefficients with a statistically insignificant difference from the baseline level are indicated in square brackets.

Table 4. *Error rate for labeled and unlabeled relations grouped by sentence length*

	Prevalence of feature, %	Unlabeled errors, %	Labeled errors, %
Basic level	–	12.29	15.47
2 – 11 tokens	12.51	7.86	11.44
12 – 17 tokens	14.63	10.46	14.37
18 – 21 tokens	11.61	11.99	14.78
22 – 27 tokens	13.74	11.87	[15.36]
28 – 33 tokens	10.89	**13.39**	**16.51**
34 – 41 tokens	11.69	**14.34**	**17.31**
42 – 58 tokens	12.52	**14.18**	**17.34**
59 – 140 tokens	12.41	**14.86**	**17.10**

Longer sentences are clearly associated with higher error rates. The error rate for both labeled and unlabeled

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2025. No. 2 (32)

connections exceeds the baseline level for sentences with a length of 28 tokens or more.

From a linguistic point of view, these longer sentences are often either "sentences within sentences" (e.g., direct speech) or several different sentence structures joined together. Since the syntactic analyzer works with greater accuracy on shorter sentences, we must explore different segmentation methods that would allow us to split sentences, analyze the parts, and then merge them into a single tree.

4.2.3 Distance to the head word

The conclusions for the frequency of errors grouped by the distance between the dependent word and the head word are shown in **Table 5**. In this grouping, a negative distance means that the head word precedes the dependent word, and a positive distance means that the head word follows the dependent word. Root nodes are highlighted in a separate group. Error rates above the baseline are highlighted in bold. All differences from the baseline are statistically significant.

Table 5. *Error rate for labeled and unlabeled relations grouped by distance between dependent and head words*

	Prevalence of feature, %	Unlabeled errors, %	Labeled errors, %
Basic level	–	12.29	15.47
– 4 and further	15.09	**33.00**	**37.82**
– 3	5.50	**19.51**	**23.58**
– 2	10.45	9.99	13.65
– 1	12.60	7.38	11.19
0 (root node)	6.44	8.07	8.62
1	28.25	3.42	5.54
2	10.05	8.16	10.81
3	4.53	11.23	14.84
4 and further	7.09	**20.43**	**24.76**

The error rate increases with greater distances between the dependent and main words. It should be noted that accuracy in the "forward" direction is better than in the “backward” direction, i.e., for the same absolute distance, the error rate is lower when the dependent word precedes the corresponding main word.

The base error rate is exceeded when the main word is 4 or more words away from the dependent word in the forward direction and 3 or more words away in the reverse direction.

4.2.4 Part of speech of the dependent word

Conclusions regarding the frequency of errors grouped by the part of speech of the dependent word are given in Table 6. Error frequencies above the baseline are highlighted in bold. Coefficients with a statistically insignificant difference from the baseline are indicated in square brackets.

Table 6. *Error rate for labeled and unlabeled links grouped by part of speech of the dependent word*

	Prevalence of feature, %	Unlabeled errors, %	Labeled errors, %
Basic level	–	12.29	15.47
NOUN	26.30	**15.95**	**19.78**
PUNCT	18.15	**12.47**	12.47
ADJ	11.46	7.34	8.72
VERB	9.21	**15.22**	**20.09**
ADP	9.21	2.38	2.56
ADV	4.16	**17.79**	**24.44**
PROPN	3.67	**15.96**	**20.70**
DET	3.65	8.88	13.37
CCONJ	3.65	8.91	10.85
PRON	3.01	10.74	**18.25**
NUM	2.26	7.62	**19.40**
PART	2.17	**19.35**	**26.95**
SCONJ	1.51	6.17	10.10
X	0.72	**37.33**	**52.58**
AUX	0.70	**13.38**	**[15.68]**
SYM	0.10	**22.35**	**43.06**
INTJ	0.06	1.40	6.80

It should be noted that for punctuation (PUNC), the frequency of marked and unmarked errors is the same, since all punctuation tokens have the "punct" relation type.

The frequency of errors in pronouns (PRON) and numerals (NUM) deserves special attention. For these categories, the frequency of unlabeled errors is lower than the baseline, while the frequency of labeled errors exceeds it. This means that the analyzer often correctly identifies the head word but not the relation type.

Methods could be developed to improve recognition accuracy in this case. We believe that formal rules could be applied to check and possibly correct the labeling.

5. Conclusions

We trained several models for syntactic analysis with different architectures and evaluated their performance on a manually annotated Ukrainian dataset.

We found that the parsing accuracy for the Ukrainian dataset is lower than for many other Indo-European languages, but the overall trend of accuracy improvement with deep word embeddings is consistent with existing research.

We analyzed the error profile of the syntactic analyzer based on transitions with deep word embeddings and demonstrated that the decrease in syntactic analysis accuracy is related to non-projectivity of dependencies,

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 2 (32) ISSN 2524-2296 (online)

longer sentences, and greater distance between the head word and the dependent.

We found that pronouns and numerals meet the baseline metrics for unmarked relations but have significantly higher error rates for marked relations.

In future work, we intend to explore ways to improve syntactic analysis accuracy by supplementing neural analysis with other approaches, such as formal rules or pre- and post-processing.

References

1. Tsarfaty, R.; Seddah, D.; Goldberg, Y.; Kuebler, S.; Versley, Y.; Candito, M.; Foster, J.; Rehbein, I.; Tounsi, L. (2010), "Statistical Parsing of Morphologically Rich Languages (SPMRL) What, How and Whither". Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages. P. 1–12. URL: https://aclanthology.org/W10-1401/
2. Kotsyba, N.; Moskalevskyi, B.; Romanenko, M.; Samoridna, H.; Kosovska, I.; Lytvyn, O.; Orlenko, O.; Brovko, H.; Matushko, B.; Onyshchuk, N.; Pareviazko, V.; Rychyk, Y.; Stetsenko, A.; Umanets, S.; Masenko, L. (2021), "Gold standard Universal Dependencies corpus for Ukrainian (UD_Ukrainian-IU) v2.8". URL: https://github.com/UniversalDependencies/UD_Ukrainian-IU.
3. Silveira, N.; Dozat, T., de Marneffe M.-C.; Bowman, S.; Connor M.; Bauer, J.; Manning C. (2014), "A Gold Standard Dependency Corpus for English". Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014). P. 2897–2904. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1089_Paper.pdf
4. Jurafsky D., Martin J. "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd ed". USA. 2025. 599 p. URL: https://web.stanford.edu/~jurafsky/slp3/
5. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. (2013), "Efficient Estimation of Word Representations in Vector Space". arXiv: 1301.3781 [cs.CL]. DOI: 10.48550/arXiv.1301.3781
6. Kulmizev, A.; de Lhoneux, M.; Gontrum, J.; Fano, E.; Nivre, J. (2019), "Deep Contextualized Word Embeddings in Transition-Based and Graph-Based Dependency Parsing - A Tale of Two Parsers Revisited". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).* P. 2755–2768. DOI: 10.18653/v1/D19-1277
7. Honnibal M.; Johnson M. (2015), "An Improved Non-monotonic Transition System for Dependency Parsing". Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. P. 1373–1378. DOI: 10.18653/v1/D15-1162
8. De Marneffe, M.-C.; Manning, C.; Nivre, J.; Zeman, D. (2021), "Universal Dependencies". *Computational Linguistics.* Vol. 47. No. 2. P. 255–308. DOI: 10.1162/coli_a_00402
9. Chaplynskyi D. (2023), "Introducing UberText 2.0: A Corpus of Modern Ukrainian at Scale". *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP).* P. 1–10. DOI: 10.18653/v1/2023.unlp-1.1
10. Starko V.; Rysin A. (2023), "Creating a POS Gold Standard Corpus of Modern Ukrainian". *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP).* P. 91–95. DOI: 10.18653/v1/2023.unlp-1.11.
11. Shvedova M.; Lukashevskyi A. (2024), "UD_Ukrainian-ParlaMint". URL: https://github.com/UniversalDependencies/UD_Ukrainian-ParlaMint.
12. De Lhoneux, M.; Stymne S.; Nivre J. (2017), "Arc-Hybrid Non-Projective Dependency Parsing with a Static-Dynamic Oracle". Proceedings of the 15th International Conference on Parsing Technologies. P. 99–104. URL: https://aclanthology.org/W17-6314/
13. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. (2018), "Deep Contextualized Word Representations". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* P. 2227–2237. DOI: 10.18653/v1/N18-1202
14. Che, W.; Liu, Y.; Wang, Y.; Zheng, B.; Liu, T. (2018), "Towards Better UD Parsing: Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation". *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies.* P. 55–64. DOI: 10.18653/v1/K18-2005
15. Eisner J. (1996), "Three New Probabilistic Models for Dependency Parsing: An Exploration". COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics. P. 340–345. URL: https://aclanthology.org/C96-1058/
16. Nivre J.; Hall J.; Nilsson J. (2004), "Memory-Based Dependency Parsing". Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004. P. 49–56. URL: https://aclanthology.org/W04-2407/
17. Nivre J.; Fang C.-T. (2017), "Universal Dependency Evaluation". Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017). P. 86–95. URL: https://aclanthology.org/W17-0411/

Надійшла (Received) 19.03.2025

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) *Innovative technologies and scientific solutions for industries. 2025. No. 2 (32)*

Відомості про авторів / About the Authors

Syrotkin Kostiantyn – National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Master, Kyiv, Ukraine; e-mail: k.syrotkin@kpi.ua; ORCID ID: https://orcid.org/0009-0008-4150-8325

Сироткін Костянтин Вікторович – Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", магістр, Київ, Україна.

ОЦІНЮВАННЯ ТОЧНОСТІ ТА АНАЛІЗ ПОМИЛОК РОЗБОРУ ЗАЛЕЖНОСТЕЙ ДЛЯ ТЕКСТІВ УКРАЇНСЬКОЮ МОВОЮ

Предметом дослідження є розбір залежностей в межах фреймворку *Universal Dependencies* для речень українською мовою. **Метою роботи** є оцінювання і порівняння точності розбору залежностей, яка досягається декількома сучасними системами на стандартному наборі даних українською мовою, а також аналіз профілю помилок таких систем. У статті виконано два **завдання**. Перше – оцінювання точності декількох сучасних систем розбору залежностей із використанням анотованого вручну стандартного набору даних. Метрикою точності обрано відсоток правильно побудованих позначених та непозначених зв’язків. Друге завдання – аналіз та категоризація помилок, яких припускаються стандартні системи. Виявлення та усунення цих помилок потенційно дасть змогу в подальшому створити більш точну систему розбору. У досліджуваних системах використовуються **методи** машинного навчання і нейронних мереж разом з **методами** теорії автоматів і теорії графів, а також глибокі векторні подання слів. **Основні результати**. Для української мови точність синтаксичного розбору значно підвищується з використанням глибокого векторного подання слів. Система на основі переходів із глибоким векторним поданням слів забезпечує найвищий показник точності побудови позначених зв’язків на рівні 84,66% для тестового набору даних. Для цієї самої системи підвищення рівня помилок спостерігається для непроєктивних залежностей, довших речень і більшої відстані до основного слова. Крім того, для займенників і числівників рівень помилок для позначених зв’язків значно перевищує базовий рівень, тоді як помилки для непозначених зв’язків залишаються на базовому рівні. **Висновки:** точність розбору залежностей для української мови поступається аналогічним показникам для інших мов, однак загальна тенденція підвищення точності з використанням глибокого векторного подання слів узгоджується з результатами попередніх досліджень. Для покращення загальної точності розбору важливо зосередитися на таких проблемних аспектах, як непроєктивні залежності, довгі речення та велика відстань між основним і залежним словами. У подальших дослідженнях заплановано дослідити можливості підвищення точності синтаксичного розбору способом доповнення підходу на основі машинного навчання іншими методами, зокрема використанням формальних правил або етапів попереднього та подальшого оброблення.

Ключові слова: оброблення природної мови; синтаксичний розбір; розбір залежностей; *Universal Dependencies*.

Бібліографічні описи / Bibliographic descriptions

Сироткін К. В. Оцінювання точності та аналіз помилок розбору залежностей для текстів українською мовою. *Сучасний стан наукових досліджень та технологій в промисловості*. 2025. № 2 (32). С. 102–110. DOI: https://doi.org/10.30837/2522-9818.2025.2.102

Syrotkin, K. (2025), "Accuracy evaluation and error analysis of dependency parsing of texts in Ukrainian", *Innovative Technologies and Scientific Solutions for Industries*, No. 2 (32), P. 102–110. DOI: https://doi.org/10.30837/2522-9818.2025.2.102