WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Developing the Kyrgyz Treebank: A Framework for Linguistic Analysis

İbrahim Benli [a*], ***Bakyt Sharshembaev*** [b]

[a] *Department of Computer Engineering, Faculty of Engineering, Kyrgyzstan-Turkey Manas University, Bishkek, Kyrgyzstan*
[b] *Department of Computer Engineering, Faculty of Engineering, Kyrgyzstan-Turkey Manas University, Bishkek, Kyrgyzstan*

A B S T R A C T

Kyrgyz Treebank represents dependency parsing-based treebank for the Kyrgyz language, developed as part of the Universal Dependencies project. This study focuses on establishing a comprehensive natural language processing (NLP) framework for Kyrgyz, a language with a complex morphological structure and free word order. Dependency parsing, which maps syntactic relationships between words, plays a key role in this effort.

Using the UDPipe tool for part-of-speech tagging, lemmatization, and dependency parsing, we trained a Kyrgyz language model on CoNLL-U formatted data. Evaluation across three methods showed Unlabeled Attachment Score (UAS) and Labeled Attachment Score (LAS) results consistent with other recently developed treebanks of similar size.

This work addresses critical gaps in Kyrgyz language resources, offering a valuable contribution to the computational linguistics community and enabling future exploration of this underrepresented language.

Keywords: Dependency parsing, Kyrgyz language, natural language processing, treebanks, universal dependencies.

1. Introduction

The Kyrgyz Treebank is dependency-parsing resource for Kyrgyz, developed as part of the Universal Dependencies (UD) initiative. This research establishes a foundation for building an NLP framework tailored to Kyrgyz linguistic structures. UD, a community-driven project, standardizes grammatical analysis across languages and supports multilingual parsing, cross-lingual learning, and typological research, with contributions spanning over 300 researchers and nearly 200 treebanks in 100+ languages.

The annotation scheme is derived from the advancements in the Universal Stanford Dependencies (M.-C. De Marneffe et al., 2021), the Google Universal Part-of-Speech tag set (Petrov et al., 2012), and studies on syntactic tag set formatting (Zeman, 2008).

Dependency parsing is a computational process that determines the syntactic relationships between words in a sentence, enabling the extraction of its grammatical structure. Figure 1 presents an example of a sentence analyzed through dependency parsing in English.

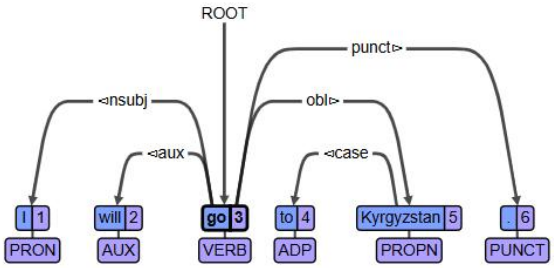

Figure 1 - An example sentence definition based on dependency parsing in English

The Kyrgyz language, the focus of our Dependency Parsing study, belongs to the Turkic language family, which is spoken by over 170 million people worldwide (Menz, 2017). While it shares syntactic and morphological similarities with other Turkic languages such as Kazakh, Uzbek, Tatar, and Turkish (Toktonaliev, 2018), most Natural Language Processing (NLP) studies in this linguistic family have predominantly focused on Turkish (Kuriyozov et al., 2020).

The Kyrgyz language exhibits a relatively free word order in its syntactic structure, influenced by elements of discourse and knowledge frameworks, similar to Turkish (Maviş et al., 2020). This flexibility in word order is a significant factor to consider when modeling speech in the Kyrgyz native language.

Example Sentence in Kyrgyz: " Чыңгыз мектепке барат." (Cengiz is going to school.)

Sentence Order and Dependencies:

SOV Order:

Kyrgyz: "Чыңгыз мектепке барат." English: "Cengiz is going to school."

Dependencies:

- nsubj(барат, Чыңгыз) (subject: Cengiz)
- obl(барат, мектепке) (oblique: to school)
- root(ROOT, барат) (root: is going)

OSV Order:

Kyrgyz: "Мектепке Чыңгыз барат." English: "To school, Cengiz is going."

Dependencies:

- obl(барат, мектепке) (oblique: to school)
- nsubj(барат, Чыңгыз) (subject: Cengiz)
- root(ROOT, барат) (root: is going)

VOS Order:

Kyrgyz: "Барат мектепке Чыңгыз." English: "Is going to school, Cengiz."

Dependencies:

- root(ROOT, барат) (root: is going)
- obl(барат, мектепке) (oblique: to school)
- nsubj(барат, Чыңгыз) (subject: Cengiz)

In Kyrgyz, disambiguating dependencies based solely on word order presents a significant challenge due to its free word order. Dependency parsers, primarily trained on languages with fixed word order (such as English), face difficulty in accurately parsing sentences with flexible structures due to heightened ambiguity. As illustrated in the examples above, the roles of subject, object, and verb shift depending on their sentence positions. To ensure accurate annotation and parsing, annotators must incorporate additional linguistic cues, such as case markings, agreement patterns, and semantic context.

The suitability of different syntactic representations for various language types is a critical consideration. Constituency-based representations, predominantly shaped by data from the Penn Treebank (Marcus et al., 1993), have been widely employed in most English linguistic studies.

Accurate and consistent tagging of words in training data is essential for effective model creation. This study employed data in CoNLL-U format, where word dependencies within sentences are specified under the "DEPREL" (Dependency Relations) label. The CoNLL-U format facilitates the representation of sentence structures, including word order, qualifying relationships, and interdependencies.

Hierarchical trees provide a visual representation of these dependencies. To ensure the correctness of dependencies, we utilized hierarchical trees as verification tools. Figure 2 demonstrates examples of hierarchical tree structures in both Kyrgyz and English.

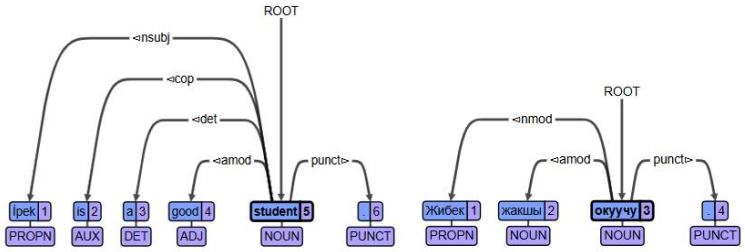

Figure 2 - Hierarchical Tree Structure Example in Kyrgyz and English

In these examples, the graphical tree structures were created using a manual dependency annotation tool called **"UD Annotatrix."** This tool provides a user-friendly drag-and-drop interface, allowing for the editing of POS tags and dependency relations efficiently (Tyers et al., 2017).

2. Related Works

The Universal Dependencies (UD) project provides dependency-parsing-based treebanks for over 160 languages on its official website. While UD v2.11 includes several Turkic treebanks, no treebank currently exists for the Kyrgyz language. For other Turkic languages, resources include a Kazakh treebank with 10K words (Tyers & Washington, 2015), a Uyghur treebank with 40K words (Eli et al., 2016), and a Tatar treebank with 2K words (Taguchi et al., 2022). Additionally, there is a treebank named UD Old Turkish Tonqq, comprising 20 sentences and 158 words (Derin & Harada, 2021). Turkish, with nine different UD treebanks totaling 736K words, has the most extensive resources among Turkic languages. However, apart from Turkish, resources for Turkic languages remain limited in UD v2.12.

Sulubacak et al. analyzed the Turkish-IMST Treebank, consisting of 57K words, achieving a labeled attachment score (LAS) of 75.3% and an unlabeled attachment score (UAS) of 83.7% (Sulubacak et al., 2016). For smaller treebanks (fewer than 20K words), Lynn et al. achieved LAS 63.3% and UAS 73.3% with the Irish Treebank (Lynn et al., 2012), Tyers et al. obtained LAS 64.9% and UAS 77.0% with the Kazakh-KTB Treebank (Tyers & Washington, 2015), and Ishola et al. reached LAS 64.9% and UAS 71.8% with the Yoruba Treebank (Ishola & Zeman, 2020).

Arnardóttir et al., working with Icelandic data using UDPipe, reported LAS and UAS scores of 55.29% and 63.03%, respectively (Arnardóttir et al., 2023). In contrast, Özateş et al. achieved LAS 77.65% and UAS 82.58% using the BERT model on an Ottoman Turkish treebank with 100 sentences (Özateş et al., 2024). Additionally, Blaschke et al. analyzed the Multi-Dialectal Bavarian UD Treebank (GSD) with techniques such as Stanza, BERT, and UDPipe, where UDPipe-trained GSD models yielded the highest scores, with LAS 65.79% and UAS 79.60% (Blaschke et al., 2024).

Table 1 provides a comparative overview of LAS and UAS results for selected languages from the CoNLL 2017 Shared Task (Straka & Straková, 2017).

Table 1- Results of CoNLL 2017 Shared Task

Language	UAS	LAS
Bulgarian	91.86	87.56
English-LinES	83.36	80.51
Finnish-TDT	89.45	85.31
Italian	90.72	87.21
French-Sequoia	88.11	86.66
Korean	68.10	62.06
Russian	83.73	80.84
Swedish	86.45	83.52
Vietnamese	69.63	66.22

3. Method

3.1 Data Set

A total of 1100 sentences were selected to train and develop the Kyrgyz UDPipe model, ensuring diversity by incorporating multiple sources. Of these, 900 sentences were obtained from Kyrgyz news websites, while 200 were extracted from Kyrgyz stories and novels. The dataset consists of an average sentence length of 8 words, amounting to 9500 words and 10522 tokens. Data extraction from news websites was performed using the Python BeautifulSoup library(Hajba, 2018). The resulting Kyrgyz Treebank is included in UD v2.13, and all data utilized in this study are publicly available.

3.2 Related NLP Techniques

UDPipe[1] is a versatile C++ tool integrating POS tagging, lemmatization, and dependency parsing, released under the Mozilla Public License (MPL). Supporting approximately 100 languages, it processes text to the dependency syntax level, allowing customization of input text, tokenization,

[1] http://ufal.mff.cuni.cz/udpipe

segmentation, and output formats. UDPipe enables training on any language with an accessible CoNLL-U treebank, including all UD corpora, storing trained models in a single file. Libraries for Java, Python, Perl, C#, and R are also available (Straka et al., 2016). In this study, model training was conducted in R using the CRAN udpipe library[2].

BERT (Bidirectional Encoder Representations from Transformers), introduced by Google in 2019 (Devlin et al., 2019), is a 12-layer deep neural network designed for self-supervised language training. Unlike word2vec, BERT is a language representation model that generates contextual word embeddings and explicit sentence representations, offering a more nuanced approach to text representation in NLP. Unlike word2vec, which primarily maps words to vectors, a pre-trained BERT model enables tasks at both the token and sentence levels.

OntoLex, or OntoLex-Lemon, is a standard ontology providing a formal model for lexical entries, word senses, and related linguistic information using RDF and OWL semantics (McCrae et al., 2017).A key feature of OntoLex is its alignment with other semantic standards, enabling interoperability across computational systems (Buitelaar et al., 2011). Designed for the Semantic Web framework, OntoLex represents lexical resources and linguistic knowledge (Cimiano et al., 2010).Using OntoLex, Senuma and Aizawa developed a dependency parsing model for Ainu, encompassing 10,000 words (Senuma & Aizawa, 2019).

ELMo (Embeddings from Language Models) is a deep contextualized word representation model that adapts embeddings based on sentence context, unlike traditional embeddings with fixed vectors. By leveraging LSTM networks, ELMo captures complex linguistic features and has significantly enhanced NLP tasks like sentiment analysis, named entity recognition, and question answering through its ability to capture both syntactic and semantic nuances (Peters et al., 2018).

Our research primarily aimed to create a comprehensive dependency parsing-based treebank for the Kyrgyz language within the Universal Dependencies (UD) framework. Focusing on facilitating morphosyntactic annotation and syntactic analysis, we emphasized tasks essential for NLP specific to Kyrgyz. UDPipe was chosen for its capability to handle part-of-speech tagging, lemmatization, and syntactic parsing, offering pre-trained models tailored for morphosyntactic annotation. These features made it an ideal tool for developing the Kyrgyz treebank.

3.2.1. UDPipe Tokenizer

Text in UD and CoNLL-U files is organized into multiple levels, where documents contain paragraphs, which in turn include sentences made up of token sequences. This structure enables the original text to be reconstructed as a sequence of tokens separated by appropriate spaces rather than as a simple sequence of words.

A single-layer, bidirectional GRU (Gated Recurrent Units) network performs sentence segmentation and tokenization simultaneously. The network predicts, for each character, whether it marks the end of a token, the end of a sentence, or neither. By excluding spaces from tokens, the network ensures precise predictions of sentence and token boundaries (Straka & Straková, 2017).

To support accurate reconstruction of pre-tokenized text, the CoNLL-U format used in UD treebanks employs the **SpaceAfter=No** feature, indicating that a token was not followed by a space in the original text (Straka et al., 2016).

3.2.2. UDPipe Tagger

For each word, UDPipe generates several triples (UPOS, XPOS, FEATS) based on the last four characters, and an average perceptron tagger with a fixed set of features resolves the ambiguity of these tags. To produce (lemma rule, UPOS) pairs, UDPipe applies a lemma rule that modifies a word by removing specific prefixes and suffixes and adding new ones.

The correct lemma rules are generated not only by considering the last four characters of a word but also by incorporating the word's prefix. Disambiguation in this process is also handled by an average perceptron tagger (Straka & Straková, 2017).

3.2.3. UDPipe Dependency Parsing

The parser utilizes the embeddings FORM, UPOS, FEATS, and DEPREL, all of which are randomly initialized and updated during training. For form embeddings, precomputed values generated with word2vec using the training data are applied.

To optimize efficiency, UDPipe precomputes as many network operations as possible for input embeddings but limits this to the 1000 most frequently used placements of each type to maintain reasonable memory requirements and load times. In UD, sentences with multiple roots are not allowed. Therefore, UDPipe generates only one root node, assigning the root dependency relation exclusively to this node (Straka & Straková, 2017).

[2] https://cran.r-project.org/web/packages/udpipe/index.html

3.3 Annotation Guidelines for Kyrgyz [3]

3.3.1 Tokenization and Word Segmentation

UD annotation adopts a lexicographical approach to syntax, establishing dependency relationships between words. Morphological properties are encoded as attributes of words rather than splitting them into morphemes. In Kyrgyz, words are typically separated by whitespace characters, and many punctuation marks are attached to neighboring words according to typographical rules.

These punctuation marks are usually tokenized as separate tokens (words), with some exceptions:

- A period marking an abbreviation remains part of the abbreviation token (e.g., млн.).
- A hyphen connecting a morphological suffix to a number does not act as a token separator (e.g., 200-гө).
- Multi-word tokens are occasionally segmented into individual syntactic words (e.g., 250'дөн).

3.3.2 Morphology

Kyrgyz possesses a rich inflectional and derivational morphology. Nouns in Kyrgyz adopt various case endings that change according to vowel harmony. In the Kyrgyz language, question suffixes, such as *-бы*, are written adjacent to the word. Kyrgyz does not have grammatical gender.

The Number feature includes two values: 'Sing' and 'Plur.' For NOUN, PROPN, and ADJ, the plural form is marked with a suffix and annotated as 'Plur,' while the singular form is unmarked and unannotated. Pronouns (PRON) possess both values and are treated lexically, meaning the plural pronoun has a unique lemma distinct from its singular counterpart.

Case has seven possible values: 'Nom,' 'Gen,' 'Dat,' 'Acc,' 'Loc,' 'Abl,' and 'Ins.' These occur with nominal words (e.g., NOUN, PROPN, PRON, ADJ, NUM), as well as with gerunds and participles (VERB, AUX). The Degree feature, which applies to adjectives (ADJ) and adverbs (ADV), has a single value: 'Cmp.' The basic positive form remains unmarked and unannotated. Polarity, applicable to verbs (VERB, AUX), also has one value: 'Neg,' with the positive form left unmarked and unannotated.

Finite verbs are typically annotated as having the habitual aspect ('Perf'). Infinitives and converbs may exhibit other aspect values, such as 'Imp' or 'Prog.' Finite verbs always display one of five mood values: 'Ind,' 'Imp,' 'Opt,' 'Pot,' or 'Des.' Conditional converbs use the conditional mood ('Cnd'). Verbs in the indicative mood express one of three tense values: 'Past,' 'Pres,' or 'Fut,' with the future tense ('Fut') appearing in participles. The Evident feature distinguishes the first-hand past tense ('Fh'). Lastly, the Voice feature includes one value: 'Pass,' while the active form remains unmarked and unannotated.

3.3.3 Syntax

In the UD schema, syntactic annotation establishes dependency relationships between words. As depicted in Figure 3, this involves constructing a tree structure where one word functions as the head of the sentence, linked to a conceptual ROOT, while all other words depend on another word in the structure.

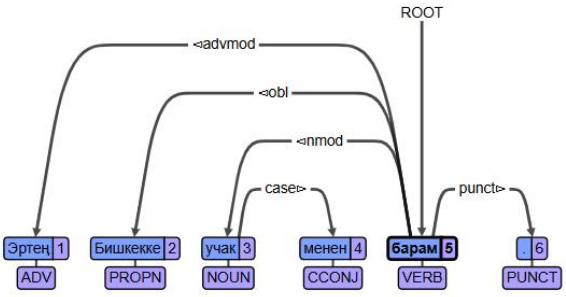

Figure 3- Dependency Representation of "*Эртең Бишкекке учак менен барам.*" clause in Kyrgyz.

In the Universal Dependencies (UD) framework, **advmod (Adverbial Modifier)** relation is used to mark adverbs or adverbial phrases that modify a verb, an adjective, or another adverb. The oblique **obl (Oblique Nominal)** relation is used for noun phrases that function as non-core arguments or adjuncts of a predicate, **nmod (Nominal Modifier)** links a noun to another noun or a nominal expression that modifies it, **nummod (Numeric Modifier** is used to connect a number to the noun it quantifies. The case **case (Case Marker)** relation links a preposition, postposition, or other case-marking word to the noun or pronoun it governs. These relations help capture the grammatical structure and semantic relationships between words in a sentence, ensuring consistent and detailed annotation across languages.

[3] https://universaldependencies.org/ky/index.html

3.4 Conversion to CoNLL-U Format

The data chosen for conversion to CoNLL-U format initially underwent manual annotation for the first 1000 words. This annotated data was then used to train a model in the R programming language. The trained model was subsequently employed to automatically tag the following 1000 words, which were then manually validated to ensure accuracy. During preprocessing, sentences were preserved in their original form, maintaining punctuation marks, capitalization, and special characters without modifications. Table 2 outlines the step-by-step process of the CoNLL-U format conversion applied in this study.

Table 2- CoNLL-U Conversion Process

Label	Process
Lemmas	Automatically annotated, Manual validated.
UPOS	Automatically annotated, Manual validated.
XPOS	Automatically annotated, Manual validated.
Features	Automatically annotated, Manual validated.
Relations	Automatically annotated, Manual validated.

3.5 Part of Speech Tagging

In the CoNLL-U format, the grammatical role of a word (e.g., Subject, Verb, Adjective, Adverb) is specified under the UPOS heading. UD v2.0 defines 17 universal POS tags, of which 13 were utilized in the Kyrgyz Treebank. These tags represent the fundamental linguistic categories. Figure 4 illustrates the tokenization and annotation of the Kyrgyz clause *"Дүкөндөн 2 көйнөк сатып алдым"*.

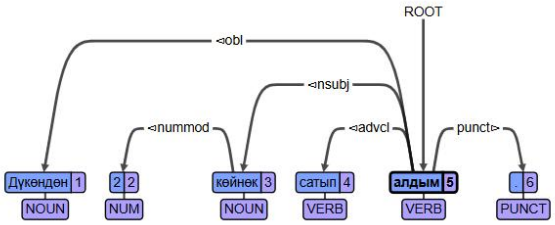

Figure 4 - Tokenization and Annotation of "*Дүкөндөн 2 көйнөк сатып алдым.*" Clause in Kyrgyz.

Additional lexical and grammatical features of words are detailed under the XPOS and Features headings. These definitions provide further specificity beyond the universal POS tags. Table 3 presents the distribution of POS tags and tokens within the Kyrgyz Treebank.

Table 3- POS and Tokens

POS	Number of Tokens
ADJ	488
ADP	21
ADV	443
AUX	75
CCONJ	338
DET	33
NOUN	4058
NUM	648
PRON	252
PROPN	285
PUNCT	1590
SCONJ	3

POS	Number of Tokens
SYM	2
VERB	2286

3.6 Universal Dependency Relations

UD v2 defines 37 universal syntactic relations, representing a revised version of the relationships initially outlined in *"Universal Stanford Dependencies: A cross-linguistic typology"* (M. C. De Marneffe et al., 2014). In our study, 25 of these universal syntactic relations were utilized. Figures 3 and 4 include examples of dependency parsing.

3.7 Word2vec

Word2Vec, developed by Mikolov et al. (Le & Mikolov, 2014), is a word vector representation method based on Artificial Neural Networks. It converts words into vectors, calculating distances to establish relationships among them. Trained on large text corpora, Word2Vec generates unique high-dimensional vectors where semantically similar words are positioned closely. Common training methods include Skip-gram and Continuous Bag of Words (CBoW) (Bilgin, 2019). In this study, pre-trained word vectors from fastText, trained on Common Crawl and Wikipedia, were used (Mikolov et al., 2019). The model employed CBoW with position weights, a 300-dimensional space, character n-grams of length 5, a window size of 5, and 10 negative samples.

3.8 Training Parameters

The Kyrgyz Treebank was developed using the R statistical programming language. Tokenization, part-of-speech tagging, lemmatization, and dependency parsing were managed through the CRAN package 'udpipe' library. The model was trained using the default parameters of the udpipe package.

4. Results and Discussion

The Kyrgyz Treebank stands as dependency parsing treebank for the Kyrgyz language included in the Universal Dependencies (UD) project. During its initial evaluation, challenges emerged due to the limited size of the dataset, leading to variability in the results.

The UD community underscores the significance of large datasets in achieving optimal efficiency and accuracy. Nonetheless, it acknowledges the importance of publishing smaller datasets for underrepresented languages without imposing strict limitations. For smaller treebanks, conducting tenfold cross-validation is advantageous, even when an official dataset split is available for experimental comparisons.

In alignment with the Universal Dependencies guidelines for developing and evaluating treebanks with fewer than 20,000 words, we employed three evaluation methods for Dependency Parsing:

Equal Split Method: The dataset was divided evenly into two subsets, with 50% allocated for training and 50% for testing, resulting in two datasets of 550 sentences each.

90/10 Split Method: The dataset was divided into 90% training data and 10% test data. This split yielded 1100 sentences (9500 words), with 990 sentences used for training and 110 sentences for testing.

K-Fold Cross-Validation Method: This technique, commonly employed in machine learning, was applied to our dataset consisting of 1100 sentences. The dataset was divided into 5 subsets (k = 5), with 90% of each subset allocated for training and 10% for testing. The choice of k = 5 was made because the dataset is relatively small, ensuring a balanced evaluation without compromising the reliability of the results. The evaluation results were averaged across all subsets to produce consolidated outcomes (Anguita et al., 2012).

Table 4 provides a detailed comparison of the evaluation results obtained using these three methods.

Table 4 - Evaluation results of KyrgyzTreebank

	Data part	Dependency parser scores	
Method	**name**	**UAS %**	**LAS %**
1	-	73.05	62.89
2	-	77.35	67.30
3	Part 1	70.55	60.40

Method	Data part name	Dependency parser scores UAS %	LAS %
	Part 2	70.27	59.93
	Part 3	74.19	60.44
	Part 4	76.10	63.53
	Part 5	80.99	67.45
	Average	74,42	62,35

Attachment scores are designed to measure the effectiveness of dependency parsing by evaluating the accuracy of head-word assignments and syntactic relations. These scores are calculated as the percentage of words with correctly assigned heads or labels.

There are two primary types of attachment scores:

- **Unlabeled Attachment Score (UAS):** This metric assesses the accuracy of dependency parsing based solely on the syntactic head, without taking the relation tag into account.
- **Labeled Attachment Score (LAS):** This score evaluates the percentage of words that correctly identify both the syntactic head and the associated dependency relation.

Together, UAS and LAS provide a comprehensive evaluation of a dependency parser's performance.

In the first evaluation method, the Unlabeled Attachment Score (UAS) was 73.05%, while the Labeled Attachment Score (LAS) reached 62.89%. The second method, utilizing a 90% training and 10% test data split, achieved higher scores with a UAS of 77.35% and an LAS of 67.30%. The third method, employing K-Fold Cross-Validation, produced an average UAS of 74.42% and an LAS of 62.35%. Among these, the second method delivered the best results, attributed to the larger volume of training data, which significantly enhanced model performance.

While the model's UAS (~80%) and LAS (~70%) may fall short of the requirements for advanced NLP applications, the dataset offers a valuable foundation for further research on Kyrgyz, a previously underexplored language. Sentences annotated with the Kyrgyz Treebank can serve as a reliable resource for NLP studies, requiring only minimal manual validation, thus reducing the demand for extensive annotation efforts in future projects.

5. Conclusion

When comparing the performance of Kyrgyz Treebank, which achieved a UAS of 77.35% and an LAS of 67.30%, with other small-sized treebanks such as the Irish Treebank (Lynn et al., 2012), Kazakh Treebank (Tyers & Washington, 2015) , and Yoruba Treebank (Ishola & Zeman, 2020), a similarity in UAS scores was evident. However, the LAS score for the Kyrgyz treebank was relatively lower. This discrepancy can be attributed to consistency issues encountered during the initial annotation process. As the study advanced and the corpus grew, inconsistencies in labeling similar syntactic structures were identified and manually corrected.

Furthermore, the limited and overlapping sources used for training and testing data may have contributed to the LAS score disparity. The resemblance between training and testing datasets could influence the results, highlighting the need for more diverse data. Nivre suggested that acceptable parsing accuracy in dependency-based treebanks requires a training set of at least 1,500 sentences. To improve the performance of future treebanks, it is recommended to use a dataset with a minimum of 2,000 sentences (Nivre, 2008).

6. Future Works

Efforts will be directed toward improving tagging operations and ensuring consistency in corpus annotations. Enhanced results are anticipated once the annotation processes are standardized and consistent. Sulubacak et al., in the updated version of the Turkish-IMST model, incorporated multiword expression annotations and achieved LAS and UAS scores of 75.4% and 83.8%, respectively (Sulubacak & Eryig It, 2018). However, these results showed no significant improvement compared to the original model. Consequently, multiword expression annotations were not employed in the Kyrgyz Treebank study. Once the dataset exceeds 20,000 words, multiword expressions will be identified and their impact compared to the model's original form.

Given the syntactic, affix, and root structure similarities among Turkic languages, similar studies are planned for these languages. The aim is to publish an expanded version of the Kyrgyz Treebank in Universal Dependencies v2.16, featuring a minimum dataset of 30,000 words. Currently,

existing Kyrgyz models developed using the BERT approach are predominantly focused on automatic speech recognition studies[4]. Future research will prioritize expanding the dataset and developing a new Kyrgyz language model based on the BERT algorithm.

7. References

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S. (2012). The 'K'in K-fold cross validation. *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 441–446.

Arnardóttir, \THórunn, Hafsteinsson, H., Jasonarson, A., Ingaon, A., & Steingr\'\imsson, S. (2023). Evaluating a {U}niversal {D}ependencies Conversion Pipeline for {I}celandic. *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, 698–704. https://aclanthology.org/2023.nodalida-1.69

Bilgin, M. (2019). Kelime Vektörü Yöntemlerinin Model Oluşturma Sürelerinin Karşılaştırılması. *Bilişim Teknolojileri Dergisi*, 141–146. https://doi.org/10.17671/gazibtd.472226

Blaschke, V., Kovačić, B., Peng, S., Schütze, H., & Plank, B. (2024). MaiBaam: A Multi-Dialectal Bavarian Universal Dependency Treebank. *ArXiv Preprint ArXiv:2403.10293*.

Buitelaar, P., Cimiano, P., McCrae, J., & Sintek, M. (2011). LexInfo: A declarative model for the lexicon-ontology interface. *Journal of Web Semantics*, *9*(1), 29–51.

Cimiano, P., Montiel-Ponsoda, E., Buitelaar, P., Espinoza, M., & Gómez-Pérez, A. (2010). A note on ontology localization. *Applied Ontology*, *5*, 127–137. https://doi.org/10.3233/AO-2010-0075

De Marneffe, M.-C., Manning, C. D., Nivre, J., & Zeman, D. (2021). Universal Dependencies. *Association for Computational Linguistics*, *47*, 255–308. https://doi.org/https://doi.org/10.1162/COLI_a_00402

De Marneffe, M. C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., & Manning, C. D. (2014). Universal stanford dependencies: A cross-linguistic typology. *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, 4585–4592.

Derin, M. O., & Harada, T. (2021). Universal Dependencies for Old Turkish. *UDW 2021 - 5th Workshop on Universal Dependencies, Proceedings - To Be Held as Part of SyntaxFest 2021*, 129–141.

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

Eli, M., Mushajiang, W., Yibulayin, T., Abiderexiti, K., & Liu, Y. (2016). Universal dependencies for {U}yghur. *Proceedings of the Third International Workshop on Worldwide Language Service Infrastructure and Second Workshop on Open Infrastructures and Analysis Frameworks for Human Language Technologies ({WLSI}/{OIAF}4{HLT}2016)*, 44–50. https://aclanthology.org/W16-5206

Hajba, G. L. (2018). Using beautiful soup. In *Website Scraping with Python* (pp. 41–96). Springer.

Ishola, O., & Zeman, D. (2020). Yorùbá dependency treebank (YTB). *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, *May*, 5178–5186.

Kuriyozov, E., Doval, Y., & Gómez-Rodríguez, C. (2020). Cross-lingual word embeddings for turkic languages. *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, 4054–4062.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *31st International Conference on Machine Learning, ICML 2014*, *4*, 2931–2939.

Lynn, T., Foster, J., Dras, M., & Dhonnchadha, E. U. (2012). Active Learning and the Irish Treebank. *Proceedings of the Australasian Language Technology Association Workshop 2012*, 23–32.

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330. https://doi.org/10.1162/coli.2010.36.1.36100

Maviş, İ., Arslan, S., & Aydin, Ö. (2020). Comprehension of word order in Turkish aphasia. *Aphasiology*, *34*(8), 999–1015. https://doi.org/10.1080/02687038.2019.1622646

McCrae, J. P., Bosque-Gil, J., Gracia, J., Buitelaar, P., & Cimiano, P. (2017). The Ontolex-Lemon model: development and applications. *Proceedings of ELex 2017 Conference*, 19–21.

[4] https://huggingface.co/models?search=kyrgyz

Menz, A. (2017). *Speakers of Turkic languages: Numbers and countries. In: Eker, Süer & Çelik Şavk, Ülkü (eds.) Tehlikedeki Türk dilleri I = Endangered Turkic Languages I: Kuramsal ve genel yaklaşmalar = Theoretical and general approaches. Ankara, Astana: Uluslararası Türk* (pp. 199–204).

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2019). Advances in pre-training distributed word representations. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, *1*, 52–55.

Nivre, J. (2008). Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, *34*(4), 513–553. https://doi.org/10.1162/coli.07-056-R1-07-027

Özateş, Ş. B., Tıraş, T. E., Genç, E. E., & Bilgin Taşdemir, E. F. (2024). Dependency Annotation of Ottoman Turkish with Multilingual BERT. *LAW 2024 - 18th Linguistic Annotation Workshop, Co-Located with EACL 2024 - Proceedings of the Workshop*, 188–196.

Peters, M., Neumann, M., & Iyyer, M. (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 (*, 2227–2237.

Petrov, S., Das, D., & Mcdonald, R. (2012). A Universal Part-of-Speech Tagset. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2089–2096.

Senuma, H., & Aizawa, A. (2019). Universal dependencies for AinU. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, 2354–2358.

Straka, M., Hajič, J., & Straková, J. (2016). UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 4290–4297.

Straka, M., & Straková, J. (2017). Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. *CoNLL 2017 - SIGNLL Conference on Computational Natural Language Learning, Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, *2*, 88–99. https://doi.org/10.18653/v1/k17-3009

Sulubacak, U., & Eryig It, G. (2018). Implementing universal dependency, morphology, and multiword expression annotation standards for Turkish language processing. *Turkish Journal of Electrical Engineering and Computer Sciences*, *26*(3), 1662–1672. https://doi.org/10.3906/elk-1706-81

Sulubacak, U., Gokirmak, M., Tyers, F., Çöltekin, Ç., Nivre, J., & Eryi\ugit, G. (2016). Universal Dependencies for Turkish. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 3444–3454. https://aclanthology.org/C16-1325

Taguchi, C., Iwata, S., & Watanabe, T. (2022). Universal Dependencies Treebank for Tatar: Incorporating Intra-Word Code-Switching Information. *Proceedings of The Workshop on Resources and Technologies for Indigenous, Endangered and Lesser-Resourced Languages in Eurasia within the 13th Language Resources and Evaluation Conference*, *June*, 95–104. https://aclanthology.org/2022.eurali-1.17

Toktonaliev, T. K. (2018). Ratio Of In Kyrgyz and Korean To The Altaic Language Theory. *ИЗВЕСТИЯ ВУЗОВ КЫРГЫЗСТАНА*, *1*, 204–207.

Tyers, F. M., Sheyanova, M., & Washington, J. N. (2017). UD Annotatrix : An Annotation Tool For Universal Dependencies. *Proceedings of the 16th International Workshop on Treebank and Linguistics Theories*, 10–17. https://ufal.mff.cuni.cz/tred/

Tyers, F. M., & Washington, J. N. (2015). Towards a Free/Open-source Universal-dependency Treebank for Kazakh. *3rd International Conference on Turkic Languages Processing, (TurkLang 2015)*, 276–289.

Zeman, D. (2008). Reusable tagset conversion using tagset drivers. *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*, 213–218.