NER-Based Token Classification for Turkish Sentence Structure Analysis

Abdurrahim Gün
Artifical Intelligence
TOBB University of Economics and Technology
a.gun@etu.edu.tr

Ali Şahin
Artifical Intelligence
TOBB University of Economics and Technology
alisahin@etu.edu.tr

***Abstract*—Understanding sentence structure is crucial for NLP applications. This project aims to identify Turkish sentence elements using Named Entity Recognition (NER) in a token classification framework. The agglutinative and rich morphological structure of Turkish poses challenges for NLP tasks. This research aims to develop a robust model to accurately classify and identify sentence components such as subjects, objects, predicates and determiners.**

A deep learning approach is used to fine-tune a pre-trained BERTurk model on a dataset of labeled Turkish sentences. The model is trained to recognize and label tokens corresponding to various sentence elements, which allows for detailed sentence structure analysis.

This study aims to provide a dataset that can meet the needs of future work in this field and to contribute to deep learning-based shallow parser approaches.

***Index Terms*—Named Entity Recognition, BERT, Token Classification, Natural Language Processing, Turkish Language**

I. Introduction

Analyzing sentence structure in the field of Natural Language Processing (NLP) is crucial for applications. This project focuses on identifying sentence elements in Turkish by leveraging Named Entity Recognition (NER) techniques within a token classification framework. The primary objective of the project is to develop a robust model capable of accurately classifying and identifying sentence components such as subjects, objects, predicates, and modifiers.

A. Identified Sentence Elements

- Subjects: The fundamental elements of the sentence, indicating who is performing the action or experiencing the state.
- Objects: The elements affected by or directed towards the verb.
- Predicates: The main verb of the sentence, defining its meaning.
- Adverbial Complements: Elements that modify the verb, indicating aspects such as place, time, amount, and reason.
- Indirect Objects: Elements indicating to whom or what the action of the verb is directed.

B. Complexity of Analyzed Sentences:

The project aims to analyze sentences with varying levels of linguistic complexity. The following linguistic phenomena will be considered:

- Nested Structures: Sentences containing other embedded sentences, creating complex structures.
- Dependencies: Identifying dependency relationships between words in a sentence is crucial, such as the relationship between a verb and its subject or object.
- Morphological Variations: Given that Turkish is an agglutinative language, the affixes attached to words and the changes in meaning these affixes bring must be considered.

C. Labeled Sentence Components

Accurate labeling of sentence components is critical for the success of the model. This labeling process involves identifying the role of each word or word group in the sentence. For example:

- Subject
- Definite Object
- Indefinite Object
- Predicate
- Adverbial Complement
- Indirect Object
- Punctuation
- Extraneous Elements (Outsides)

This detailed approach will allow for more accurate and comprehensive analyses of Turkish sentences. During the project, deep learning and NER techniques will be utilized to label these sentence components and address linguistic phenomena.

One of the most significant challenges in this field is the lack of a suitable dataset that meets the needs of such analyses. Therefore, one of the fundamental aims of this project is to create a dataset that can also benefit future research. Details on how the dataset will be constructed are thoroughly explained in section 3.

Another critical gap identified in the literature is the lack of sufficient and comprehensive studies on the use of deep learning and NER tools for this specific purpose. NER provides a useful and important infrastructure for the intended task. Breaking down Turkish sentences into their components determines the roles of words in a sentence (such as subject, object, predicate, adverbial complement, and indirect complement). In this process, the type and position of the word in the sentence are determined. NER attempts to identify named entities (such as persons, places, organizations, dates, etc.) in a sentence. These named entities are often used as specific components in a sentence (such as subjects or objects). When the labels for words or word groups are correctly structured, the problem becomes solvable using an NER approach. NER algorithms can utilize information derived from syntactic analyses to better understand the context of words. This allows for a more successful breakdown of the sentence into its components.

In this study, we will fine-tune the pre-trained BERTurk model using the dataset we create. Research has shown that the base model performs well for many NLP tasks. Therefore, the base model will be used initially. If its performance is found to be insufficient, we will transition to the larger model. The detailed steps of the project are explained in section 3.

D. Background and Developments in Turkish NER

Research on Named Entity Recognition (NER) for Turkish has seen significant developments over the years. Initially, Cucerzan and Yarowsky (1999) [1] introduced a language-independent bootstrapping algorithm that utilized contextual and internal word information to identify entities, applying this approach to Turkish and several other languages. Tür et al. (2003) [2] subsequently developed an HMM-based NER system specifically for Turkish and created the first widely-used tagged Turkish newspaper corpora for NER tasks. Bayraktar and Temizel (2008) [3] employed a local grammar approach to identify person names in Turkish financial texts, while Küçük and Yazıcı (2009a, 2009b) [4] established the first rule-based NER system for Turkish, applying it to various text domains, including news articles and children's stories. Dalkılıç et al. (2010) [] also contributed with a rule-based system for Turkish NER.

More recently, the focus has shifted towards machine learning methods for Turkish NER. Küçük and Yazıcı (2010, 2012) [5] extended their rule-based system into a hybrid recognizer to improve performance across different domains. Yeniterzi (2011) [6] developed a CRF-based NER system leveraging morphological features, while other CRF-based models were proposed by Özkaya and Diri (2011) [7] and Şeker and Eryiğit (2012) [8], who demonstrated that their system outperformed others using the same dataset. Demir and Özgür (2014) [9] introduced a neural network-based semi-supervised approach that surpassed earlier models on news articles without the use of gazetteers. Tatar and Cicekli (2011) [10] proposed an automatic rule learner system for Turkish NER.

Küçük and Steinberger (2014) [11] adapted rule-based NER systems for informal texts such as tweets and forums. Çelikakaya et al. (2013) [12] extended CRF-based approaches to spoken data, showing the versatility of these models in handling different types of text. Kisa and Karagöz (2015) [13] applied NLP from Scratch to propose generalized models tested on both formal and informal texts, emphasizing the adaptability of their approach. Çöltekin and Rama (2016) [14] introduced deep learning approaches for Turkish NER, utilizing convolutional and recurrent neural networks. Their models achieved state-of-the-art performance by training on large-scale datasets, demonstrating significant improvements over traditional machine learning methods. Akbik et al. (2018) [15] proposed the FLAIR framework, which includes contextual string embeddings, and showed substantial improvements in NER tasks for various languages, including Turkish.

Aras and Demir (2020) [16] evaluated neural sequence tagging models on the Turkish Named Entity Recognition (NER) task. They compared different model settings and highlighted the superior performance of transformer-based architectures, particularly when combined with a CRF layer, achieving state-of-the-art results.

II. Related Works

The research provides a comprehensive overview of deep learning-based solutions for Named Entity Recognition (NER), assisting new researchers in understanding the field [17]. The survey covers NER's historical background, traditional approaches, current state-of-the-art techniques, and future research directions. It consolidates essential NER resources, including tagged corpora and readily available systems, with a focus on general domain and English language applications. The authors review various deep learning models and categorize them using a novel taxonomy to aid in understanding their applicability to different NER challenges.

Another study addresses the nested NER problem using fine-tuned, pre-trained BERT-based language models [18]. By leveraging transfer learning with BERT models, this research offers a simpler and more effective solution compared to existing complex models. The approach transforms the nested NER problem into a flat NER problem, enabling the use of traditional NER models. The fine-tuned BERT models significantly outperform traditional models such as CRF and Bi-LSTM-CRF.

The shift in NER applications has moved from LSTM-based models (e.g., ULMFIT, ELMO) to transformer-based models (e.g., BERT, GPT) [19]. The study demonstrates that domain-specific pre-training from scratch can significantly exceed mixed-domain pre-training, achieving state-of-the-art results in various biomedical NLP applications. Pre-trained BERT models are expected to outperform LSTM-based methods in NER tasks.

The researchers performed several subtasks of the shallow discourse parsing pipeline on Turkish [20]. Although their work is not a complete end-to-end parser, it is the most comprehensive work on Turkish to date. All tasks are modeled as multi-class text classification problems using the Turkish BERT model. The results showed satisfactory accuracy comparable to English results despite limited training data. In our project, we aim to identify sentence components in Turkish using shallow parsing. Implementing a Turkish BERT model as shown in the referenced work will improve our accuracy in identifying subjects, objects and predicates.

In another shallow parsing study conducted by Isik University [21], they involved applying shallow parsing to 1400 Turkish sentences annotated by seven different annotators. They evaluated the performance of six different models in the context of shallow parsing, including classifiers such as Decision Tree (C45), Naive Bayes, K-Nearest Neighbors (KNN), Linear Perceptron and Multilayer Perceptron. These word embeddings were then used to improve the classifiers, providing continuous features for words without additional feature engineering. Based on this paper, we decided on the state-of-the-art machine learning based models that will be used to compare the performance of the fine-tuned BERTurk model.

In their paper, Mishra and Mujadia et al. (2024) [22] demonstrated that contextual embedding-based fine-tuning in a multi-task setup significantly enhances performance across various shallow linguistic tasks and multiple domains. They highlighted the effectiveness of MuRIL for both token-level and sentence-level improvements, suggesting that future work should explore integrating contextual embeddings with CRF for better chunking results. By leveraging contextual embeddings and exploring the potential use of CRF, we seek to improve the accuracy of identifying subjects, objects, and predicates in Turkish sentences, thus advancing the field of natural language processing for Turkish.

III. Data

The dataset used for this project is derived from the BOUN Treebank created by TABILAB, featuring 9,761 manually annotated sentences across various topics like biographical texts, national newspapers, educational texts, popular culture articles, and essays. These texts come from the Turkish National Corpus (TNC) and are annotated according to the Universal Dependencies (UD) framework. Initial morphological features and UPOS information are provided by Sak et al. (2011) and converted to UD morphology via a script.

For the dataset, 250 sentences from each category are used for training, and 30 sentences are used for testing and validation, totaling 1,250 training sentences and 150 sentences each for testing and validation.

The data is stored in Excel format, with each word annotated with attributes like root, word type, and number. These attributes are utilized in NER and Shallow Parsing but are not used for fine-tuning the BERTurk model. For BERTurk, IOB tagging is employed to label named entities in the text.

A. Features

- ”w” column: Columns ”w” contains sentences. We converted these words into embeddings and stored them in the ”w-embeddings” column. Word embeddings are numerical vector representations of words that capture their semantic meaning. This allows our model to process and understand the context of each word in the sentence.
- ”l” column: we have another independent column named ”l” which contains the lemmatized (root) forms of the words. These lemmas are also converted into embeddings and stored in a new column called ”l-embeddings”.
- ”x” column: The provided table contains a column with values representing the part-of-speech (POS) tags for words in sentences. We converted these POS tags into one-hot vectors and used these vectors in our model. This process enhances our model’s ability to accurately analyze and classify the grammatical structure of sentences.
- ”f” column: The ”f” column in the provided table contains dependency labels that represent the syntactic relationships between words in a sentence. We converted these dependency labels into one-hot vectors.
- ”Case” column: The ”Case” column in our dataset indicates the grammatical case of each word, which shows its syntactic relationship with other words in a sentence. Grammatical cases provide crucial information about the role of a noun or pronoun, such as whether it is the subject, direct object, or shows possession. To train our model, we converted these case tags into one-hot vectors.By converting these case tags into one-hot vectors, our model can be trained to recognize and classify the grammatical cases of words in sentences based on these encoded representations.
- ”Number” column: The ”Number” column in our dataset indicates the grammatical number of each word, showing whether a noun, pronoun, or verb is singular or plural. This information is essential for understanding the agreement between subjects, verbs, and objects in sentences. To utilize this feature in our model, we converted the ”Number” column into binary categorical data. This is because we believe that the grammatical number can assist in identifying subjects, objects, and other syntactic roles more effectively when kept in a categorical format.
- ”Polarity” column: The ”Polarity” column in our dataset indicates the sentiment polarity of each word, showing whether a word expresses a negative or positive sentiment. This information is essential for understanding the emotional tone and sentiment in sentences. To utilize this feature in our model, we categorized the ”Polarity” column into binary data. This is because the presence of positive or negative sentiment can significantly impact the likelihood of a word being a predicate.
- ”Evident” column: The ”Evident” column in our dataset indicates the evidentiality of verbs and auxiliaries, showing whether the information is firsthand (directly experienced by the speaker) or non-firsthand (reported, inferred, or assumed by the speaker). This information is crucial for understanding the source and reliability of the information conveyed in sentences. To utilize this feature

in our model, we categorized the ”Evident” column into binary data. This is because words with firsthand or non-firsthand evidentiality are often predicates or modifiers, making this feature valuable for syntactic parsing.

- ”Aspect” column: The ”Aspect” column in our dataset indicates the grammatical aspect of verbs and auxiliaries, showing the temporal flow of the action described by the verb. Aspect provides information about the completeness, duration, and frequency of an action. To utilize this feature in our model, we converted the ”Aspect” column into one-hot vectors.
- ”Person” column: The ”Person” column in our dataset indicates the grammatical person of pronouns, verbs, and auxiliaries, showing whether the word refers to the first person (speaker), second person (addressee), or third person (others). Understanding person is crucial for syntactic parsing and natural language processing tasks as it helps identify subjects and the relationships between entities in sentences. To utilize this feature in our model, we categorized the ”Person” column into binary data. This is because words with specific persons can help identify subjects and other syntactic roles.
- ”Tense” column: The ”Tense” column in our dataset indicates the grammatical tense of verbs and auxiliaries, showing the time at which the action described by the verb takes place. Understanding tense is crucial for syntactic parsing and natural language processing tasks as it provides information about when an action occurs. To utilize this feature in our model, we categorized the ”Tense” column into binary data. This is because words with specific tenses are often predicates, making this feature valuable for syntactic parsing.

B. Labelling

The ”Tag” columns in the provided table represent different syntactic roles of words in sentences. The tagging was performed by us. Here is a detailed explanation of each tag value and its meaning:

- PUNCT: Punctuation mark.
- B-Pred: Beginning of a predicate.
- I-Pred: Continuation of a predicate.
- B-Subj: Beginning of a subject.
- I-Subj: Continuation of a subject.
- B-AdverbialComp: Beginning of an adverbial complement.
- I-AdverbialComp: Continuation of an adverbial complement.
- B-DefiniteObj: Beginning of a definite object.
- I-DefiniteObj:Continuation of a definite object.
- B-IndefiniteObj: Beginning of an indefinite object.
- I-IndefiniteObj:Continuation of an indefinite object.
- B-IndirectObj: Beginning of an indirect object.
- I-IndirectObj: Continuation of an indirect object.
- O: Outside of any of the specified categories such as prepositions, conjuctions, interjections etc.

IV. Current Progress and Upcoming Tasks

As can be seen in Figure 1, the data was acquired, labeled, preprocessed and trained as described in the ”Data” section. The outputs of the models are described in detail in this section.

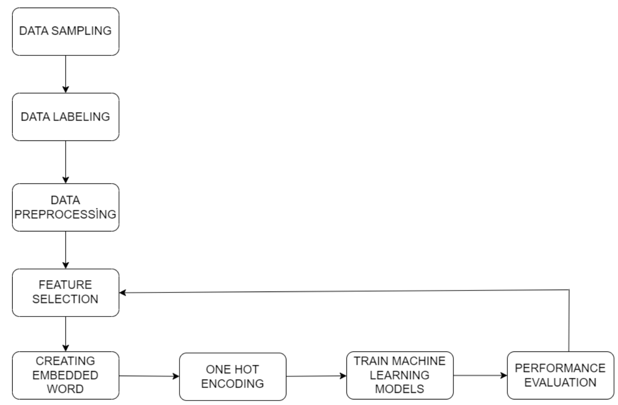

Fig. 1. Flow Chart

In Figure 2, we observe the performance metrics of various machine learning models, including K-nearest neighbors (Knn), Naive Bayes (NB), Multi-Layer Perceptron (MLP) with neuron counts of (150, 200, 250, 200, 150), and Support Vector Machine (SVM) with an RBF Kernel. The metrics evaluated are Accuracy, Precision, Recall, Weighted F1-Score, and Macro F1-Score.

- **Accuracy:** measures the proportion of correctly predicted instances among all instances.
- **Precision:** indicates the proportion of true positive results in all positive predictions.
- **Recall:** measures the proportion of true positive results out of the actual positives.
- **Weighted F1-score:** considers both precision and recall but also accounts for class imbalances by weighting classes according to their frequency.
- **Macro F1-Score:** averages the F1-Scores of each class, treating all classes equally regardless of their frequency.

	Knn	NB	MLP	SVM (RBF Kernel)
Accuracy	42,55%	37,97%	**50,77%**	43,41%
Precision	28,54%	26,87%	**38,07%**	30,85%
Recall	26,34%	26,51%	**34,21%**	25,37%
Weighted F1-Score	40%	36%	**49%**	40%
Macro F1-Score	25,78%	24,85%	**34,96%**	25,30%

Fig. 2. Model Results

As shown in Figure 2, the MLP model outperforms the other models in nearly all metrics, notably achieving the highest accuracy of 50.77%, precision of 38.07%, recall of 34.21%, weighted F1-Score of 49%, and macro F1-Score of 34.96%.

The use of macro values in the evaluation is crucial because it ensures that each class contributes equally to the overall performance metric, regardless of its frequency. This is particularly important in imbalanced datasets where some classes may be underrepresented. By using macro F1-Scores, we ensure that the model performs well across all classes, not just the majority class.

The superior performance of the MLP can be attributed to its architecture, which includes multiple layers with varying neuron counts (150, 200, 250, 200, 150). This deep and complex structure allows the MLP to capture intricate patterns and relationships in the data, leading to better performance on tasks that involve high-dimensional feature spaces, such as natural language processing (NLP).

In this analysis, we examine the Cohen's Kappa scores for various pairs of machine learning models. Cohen's Kappa is a statistical measure used to assess the level of agreement between two raters or classifiers, correcting for the agreement occurring by chance. It is a valuable metric in machine learning to determine how well different models concur in their predictions.

Models	Cohen's Kappa Scores
Knn-NB	0,49
Knn-MLP	0,48
Knn-SVM	0,6
NB-MLP	0,42
NB-SVM	0,55
MLP-SVM	0,57

Fig. 3. Cohen's Kappa Scores

Cohen's Kappa scores range from -1 to 1, where:

- 1 indicates perfect agreement
- 0 indicates no agreement beyond chance
- Negative values indicate agreement less than chance, suggesting systematic disagreement

The scores on table implicate that:

- **Knn-SVM (0.60):** The highest agreement is observed between the K-nearest neighbors (Knn) and Support Vector Machine (SVM) models. A Kappa score of 0.60 indicates a substantial level of agreement, suggesting that these two models make similar predictions on a significant number of instances.
- **MLP-SVM (0.57):** The Multi-Layer Perceptron (MLP) and SVM models also show a substantial level of agreement with a Kappa score of 0.57. This indicates that both models are likely capturing similar patterns in the data, leading to consistent predictions.
- **NB-SVM (0.55):** The Naive Bayes (NB) and SVM models have a moderate level of agreement. This score suggests that while there is a reasonable amount of concordance, the predictions from these models are not as aligned as those between Knn and SVM.
- **Knn-NB (0.49):** The Knn and NB models show moderate agreement with a Kappa score of 0.49. Although this indicates a fair level of agreement, it is lower than the agreement between the other model pairs.
- **Knn-MLP (0.48) and NB-MLP (0.42):** These pairs have the lowest Kappa scores, indicating moderate to fair agreement. The lower scores suggest that Knn and MLP, as well as NB and MLP, have more divergent prediction patterns compared to other model pairs.

Cohen's Kappa is crucial in model evaluation as it:

- **Corrects for Chance Agreement:** Unlike simple accuracy, Kappa takes into account the agreement occurring by chance, providing a more reliable measure of model consistency.
- **Model Selection:** High Kappa scores between models indicate similar performance, which can be useful for ensemble methods or selecting models for deployment.
- **Identifies Complementary Models:** Lower Kappa scores might indicate that models capture different aspects of the data, which can be beneficial for creating robust ensemble models.

A. *Limitations of Traditional Models for NLP*

Traditional machine learning models like Knn, NB, and SVM are not well-suited for NLP tasks for several reasons:

- **Feature Engineering:** These models require extensive feature engineering to process textual data, which can be labor-intensive and may not capture the full context of the language.
- **Scalability:** They often struggle with the high dimensionality and sparsity of textual data.
- **Context Understanding:** These models lack the ability to understand the context and semantics of words, which is crucial for effective NLP.

B. *Necessity of Using BERTurk*

To address these limitations, using a transformer-based model like BERTurk is essential. BERTurk, a pre-trained language model specifically designed for the Turkish language, offers several advantages:

- **Contextual Understanding:** It captures the context of words in a sentence, leading to better semantic understanding.
- **Pre-training:** Leveraging a large corpus of Turkish text during pre-training, BERTurk can provide strong language representations that enhance downstream NLP tasks.
- **Fine-tuning:** BERTurk can be fine-tuned on specific tasks with relatively little data, achieving high performance even in specialized domains.

C. *Future Works*

Progress has been made in data tagging, but it is not complete. The results are based on the labeled data. In the future, the Bi-LSTM-CRF model and the fine-tuned BERTurk model, which is our ultimate goal, will be added. Details of the BERTurk model are left for the final report. After the labeling of the obtained data is completed, it will be checked whether data should be added according to the training performance of the final model. At this stage, the "Word2vec" Python function has been used to use the words in the models and it is planned to investigate whether a more successful "embedding" can be done in the future. However, the conclusion drawn from the literature review is that the performance of machine learning models within the scope of this project is insufficient. For this reason, it is thought that changing the "embedding" process will not affect the performance of machine learning models much.

Detailed statistics of the final dataset will be provided in the final report. The reason for not providing statistics in this report is that the dataset labeling process is not completed and the dataset is not finalized.

References

[1] Silviu Cucerzan and David Yarowsky. Language independent named entity recognition combining morphological and contextual evidence. In *1999 joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, 1999.

[2] Gokhan Tur, Dilek Hakkani-Tur, and Kemal Oflazer. A statistical information extraction system for turkish. *Natural Language Engineering*, 9:181 – 210, 06 2003.

[3] Ozkan Bayraktar and Tugba Taskaya Temizel. Person name extraction from turkish financial news text using local grammar-based approach. In *2008 23rd International Symposium on Computer and Information Sciences*, pages 1–4. IEEE, 2008.

[4] Dilek Küçük and Adnan Yazıcı. Named entity recognition experiments on turkish texts. In *Flexible Query Answering Systems: 8th International Conference, FQAS 2009, Roskilde, Denmark, October 26-28, 2009. Proceedings 8*, pages 524–535. Springer, 2009.

[5] Dilek Küçük and Adnan Yazıcı. A hybrid named entity recognizer for turkish. *Expert Systems with Applications*, 39(3):2733–2742, 2012.

[6] Reyyan Yeniterzi. Exploiting morphology in turkish named entity recognition system. In *Proceedings of the ACL 2011 student session*, pages 105–110, 2011.

[7] Serap Özkaya and Banu Diri. Named entity recognition by conditional random fields from turkish informal texts. In *2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU)*, pages 662–665. IEEE, 2011.

[8] Gökhan Akın Şeker and Gülşen Eryiğit. Initial explorations on using crfs for turkish named entity recognition. In *Proceedings of COLING 2012*, pages 2459–2474, 2012.

[9] Hakan Demir and Arzucan Özgür. Improving named entity recognition for morphologically rich languages using word embeddings. In *2014 13th international conference on machine learning and applications*, pages 117–122. IEEE, 2014.

[10] Serhan Tatar and Ilyas Cicekli. Automatic rule learning exploiting morphological features for named entity recognition in turkish. *Journal of Information Science*, 37(2):137–151, 2011.

[11] Dilek Küçük and Ralf Steinberger. Experiments to improve named entity recognition on Turkish tweets. In Atefeh Farzindar, Diana Inkpen, Michael Gamon, and Meena Nagarajan, editors, *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 71–78, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[12] Gökhan Çelikkaya, Dilara Torunoğlu, and Gülsen Eryiğit. Named entity recognition on real data: A preliminary investigation for turkish. In *2013 7th International Conference on Application of Information and Communication Technologies*, pages 1–5, 2013.

[13] Kezban Dilek Onal and Pinar Karagoz. Named entity recognition from scratch on social media. In *Proceedings of 6th International Workshop on Mining Ubiquitous and Social Environments (MUSE), co-located with the ECML PKDD*, volume 104, 2015.

[14] Çağrı Çöltekin and Taraka Rama. Discriminating similar languages with linear svms and neural networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 15–24, 2016.

[15] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[16] Gizem Aras, Didem Makaroglu, Seniz Demir, and Altan Cakir. An evaluation of recent neural sequence tagging models in turkish named entity recognition, 2020.

[17] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70, 2020.

[18] Ankit Agrawal, Sarsij Tripathi, Manu Vardhan, Vikas Sihag, Gaurav Choudhary, and Nicola Dragoni. Bert-based transfer-learning approach for nested named-entity recognition using joint labeling. *Applied Sciences*, 12(3):976, 2022.

[19] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.

[20] Ferhat Kutlu, Deniz Zeyrek, and Murathan Kurfalı. Toward a shallow discourse parser for turkish. *Natural Language Engineering*, pages 1–26, 2023.

[21] Ozan Topsakal, Onur Açıkgöz, Ali Tunca Gürkan, Ali Buğra Kanburoğlu, Burak Ertopçu, Berke Özenç, İlker Çam, Begüm Avar, Gökhan Ercan, and Olcay Taner Yıldız. Shallow parsing in turkish. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pages 480–485. IEEE, 2017.

[22] Pruthwik Mishra and Vandan Mujadia. Multi task learning based shallow parsing for indian languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 2024.