A Semi-Automated Approach to the Annotation of Argument Structures in Turkish Datasets

Neslihan Cesur[1], Sabri İnce[2], Ali Hakkı Aydın[3], Ece Su Eren[2], Deniz Gücükçavuş[2], Murat Papaker[3], Kaan Bayar[2], Deniz Baran Aslan[2], Yelda Fırat[4], Olcay Taner Yıldız[1]
[1]Özyeğin University, [2]Boğaziçi University, [3]Galatasaray University, [4]Mudanya University
Correspondence: neslihan.cesur@ozu.edu.tr; olcay.yildiz@ozyegin.edu.tr

Abstract

This paper presents a PropBank annotation project for Turkish, focusing on core arguments in matrix clauses. Our dataset comprises of five different corpora with 25,580 sentences labeled for verbal predicates and core arguments. Using a semi-automatic approach, we leveraged a dependency layer to pre-assign some ARG0s and ARG1s, followed by manual corrections. Our work will be used in the development of Abstract Meaning Representations (AMRs), enhancing Turkish NLP resources for semantic parsing and higher-level language tasks.[1]

1 Introduction

Understanding argument structure is essential to the study of syntax and semantics in natural language processing (NLP). In linguistic theory, argument structure refers to the way a predicate, typically a verb, organizes its participants (arguments). These arguments can vary depending on the predicate, including roles such as the agent (who performs the action), the patient (who undergoes the action), or additional participants that further specify the event or action (e.g., a location, instrument, or beneficiary). Correctly representing the argument structure of a sentence is crucial as the arguments contain syntactic and semantic information concerning the participants of the event. This information is is used for higher-level NLP tasks such as information extraction, machine translation, question answering, and text summarization.

This understanding of argument roles has been formalized in the PropBank project (Kingsbury and Palmer, 2002, 2003; Palmer et al., 2005; Bonial et al., 2014), which provides a layer of semantic role annotations over syntactically parsed texts. It has been widely used as a resource for developing supervised machine learning models in NLP, contributing to syntactic-semantic integration. PropBank captures the relationship between predicates and their core arguments using a numbered argument schema. Each label in the schema is intended to represent generalized semantic functions: ARG0 typically corresponds to the agent or causer, while ARG1 corresponds to the theme or patient of the verb. Adjuncts and other modifiers are marked with various ARGM labels such as temporal (ARGM-TMP) or locational (ARGM-LOC) modifiers. Bonial et al. (2012) elaborates on the annotation guidelines in detail.

For our project, we conducted verbal predicate selection and argument labeling across five different datasets, comprising a total of 25,580 sentences. The annotation of our datasets was carried out as part of a broader multi-layer NLP project. Each dataset had already been labeled for various linguistic information: morphological analysis using an automatic analyzer (Yıldız et al., 2019), morphological disambiguation, and word sense disambiguation using definitions from the Turkish WordNet (KeNet) (Bakay et al., 2019, 2021). Additionally, the sentences were annotated with Universal Dependencies (UD)-style dependency relations. Before labeling the arguments, we tagged all sentences which had verbal predicates, leaving out nominal predicates and noun phrases. With the availability of a manually-annotated dependency layer, we were able to automatically assign some agents (ARG0) and direct objects (ARG1). This automatic process was followed by the manual annotation of missing arguments and the correction of automatically labeled arguments.

Since this PropBank annotation is part of a larger project aimed at constructing Abstract Meaning Representations (AMRs), we focused solely on the core arguments while excluding modifiers. The argument labeling of subordinate clauses will be conducted at the AMR stage. The labeling of ad-

[1]This work is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Project No. 123E027, titled *Learning Abstract Meaning Representation for Turkish and Building A Question Answering System.*

juncts and modifiers will be addressed right before AMR annotations, using frame elements from the Turkish FrameNet (Marsan et al., 2021).

2 Previous PropBank Datasets

Several projects have been carried out in an effort to adapt PropBank to typologically diverse languages. These projects include PropBanks for Arabic (Palmer et al., 2008; Zaghouani et al., 2010), Basque (Agirre et al., 2006; Aldezabal et al., 2010), Chinese (Xue, 2006; Xue and Palmer, 2009) and Finnish (Haverinen et al., 2015) among others.

The annotation of the first dataset, along with several studies on semantic role labeling and the development of a Turkish PropBank (Sahin, 2016; Şahin and Adalı, 2018), was carried out using the IMST Dataset (Sulubacak et al., 2016). In a related effort, Şahin (2018) conducted a verb sense annotation project using crowd-sourced annotators to create an initial PropBank resource.

Ak et al. (2018) further advanced Turkish PropBank development by using the Turkish Penn TreeBank to semantically label over 9,500 sentences and map predicate-argument structures for 1,914 Turkish verb senses. Subsequently, Ak and Yıldız (2019) introduced a method to automatically generate PropBank annotations using an English-Turkish parallel corpus.

For our annotation efforts, we utilized the frame files developed in TRopBank v2.0 (Kara et al., 2020). This version contains frame files for 17,691 Turkish verbs. A key feature of TRopBank v2.0 is that its predicate-argument structures are based not on individual verbs but on the synsets defined in Turkish WordNet (Ehsani et al., 2018; Bakay et al., 2019). While this approach is efficient in terms of time and resource management, it occasionally introduces challenges during the labeling of specific predicates.

3 Data and Annotation

The datasets used in this project are diverse in nature. Turkish ATIS (Cesur et al., 2024) and Turkish Penn Treebank (Kuzgun et al., 2020) are the adaptations of well-known NLP benchmarks, with ATIS focusing on flight-related queries (Hemphill et al., 1990) and Penn Treebank offering syntactically annotated sentences from the Wall Street Journal (Marcus et al., 1993). The Tourism dataset consists of user comments from a tourism website, reflecting informal, real-world language. FrameNet contains example sentences crafted for the Turkish FrameNet project (Marsan et al., 2021) to illustrate semantic frames. Finally, KeNet (Ehsani et al., 2018) includes example sentences extracted from the Turkish Language Association (TDK) dictionary, providing formal and structured language samples.

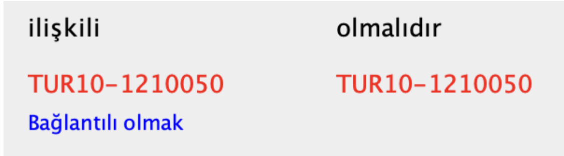

Figure 1: The interface for word sense disambiguation. The light verb construction *ilişkili olmak* (to be related) appears as a collocation in Turkish WordNet. In this context, both words are assigned the same ID.

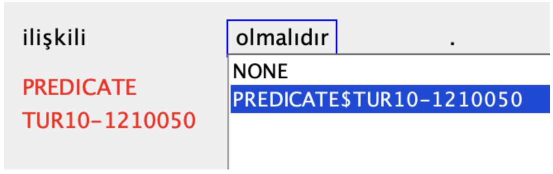

Figure 2: The interface for predicate selection. Both components of the construction *ilişkili olmak* can be marked as predicates, as they share the same WordNet ID.

3.1 Predicate selection

The first step in our annotation process was to identify and label verbal predicates across the datasets. Sentences containing only noun phrases or nominal clauses were excluded from annotation. For example, the Tourism and ATIS datasets included several instances of such structures. Sentences like *Otel iyiydi* ("The hotel was good") or *San Francisco'ya uçuşlar* ("Flights to San Francisco") were omitted because they do not contain verbal predicates.

In cases where predicates appeared as multi-word constructions, each word was labeled as a predicate only if all components shared the same WordNet ID in the word sense disambiguation layer. This ensured consistency in labeling across complex predicates. Figure 1 shows the word sense disambiguation layer for the light verb construction *ilişkili olmak* ("to be related"), where both words share the same WordNet ID, resulting in a single unified definition. Since this construction is recognized as a collocation, both words are marked as predicates in the predicate selection layer, as demonstrated in Figure 2.

Another factor influencing predicate selection is the morphological layer. Errors in morphological disambiguation can prevent the word sense layer from displaying appropriate verb definitions. If a noun sense is incorrectly selected, the predicate layer offers no options beyond "NONE." Figure 3 illustrates a case where a word has two valid morphological analyses. The word *açtı* can be interpreted either with the adjectival root *aç* ("hungry") or the verbal root *aç-* ("to open"). When combined with the past tense suffix *-DI*, the first analysis forms a nominal predicate ("he/she was hungry"), while the second forms a verbal predicate ("he/she opened (it)").

The choice made at the morphological layer directly determines the available word senses in the word sense disambiguation layer, which subsequently affects predicate selection. If the adjectival interpretation is selected, the annotator is unable to label the word as a predicate and must revise both the morphological analysis and the word sense disambiguation.

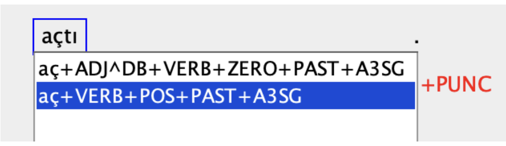

Figure 3: The interface for morphological disambiguation. The word *açtı* can either have an adjectival root and or a verbal root depending on the linguistic context.

In this phase of our project, we focused exclusively on annotating predicates and core arguments within matrix clauses, meaning that predicates appearing in embedded clauses were not labeled. However, when multiple clauses were connected through coordination, whether by a conjunction (e.g., *ve* "and," *ama* "but") or a punctuation mark (e.g., a comma or semicolon), each predicate within those coordinated structures was annotated independently. This approach ensures that all main predicates across conjoined clauses receive proper labeling, even though only the primary predicate of the matrix clause was prioritized in simpler sentence structures.

3.2 Pre-processing with the dependency layer

After predicate selection, sentences containing verbal predicates underwent an automatic labeling process, where ARG0s and ARG1s were assigned based on their dependency analysis. We assumed that most subjects would function as agents (ARG0) and therefore assigned ARG0 to tokens marked with the NSUBJ dependency relation, along with any dependents tied to them. This approach allowed entire phrases to be labeled efficiently. Figure 4 illustrates the dependency relations that result in multiple words being labeled as a single argument unit.

One challenge with this assumption arises with unaccusative predicates, whose subjects are marked as NSUBJ but semantically correspond to ARG1 rather than ARG0. Since the identification of unaccusative predicates followed the definitions in TRopBank v2.0, the algorithm ensured that for verbs which do not receive ARG0 arguments, the NSUBJ marked words would be labeled ARG1 instead of ARG0. Later the annotators corrected any exceptional cases during the manual review.

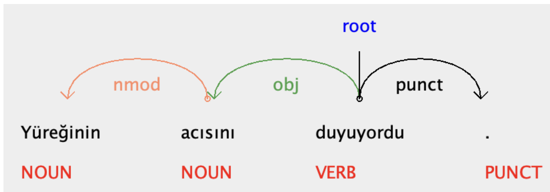

Figure 4: Dependency relations for the sentence meaning *(He/She) felt the pain in his/her heart*. The head of the noun phrase (*acısını*) is connected to the root with the OBJ tag, leading it to be labeled as ARG1. Since *yüreğinin* is attached to the head via the NMOD tag, it also inherits the ARG1 label.

Similarly, passive predicates posed another challenge. Although passive subjects should be marked as ARG1, they are often labeled NSUBJ in the dependency layer. While Universal Dependencies (UD) provides the NSUBJ:pass tag for passive subjects, inconsistencies in the datasets meant many passive subjects were marked with NSUBJ. In such cases, ARG0 was manually corrected to ARG1.

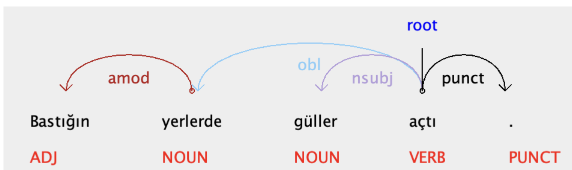

Figure 5: Dependency relations for the sentence meaning *Flowers bloomed wherever you stepped.* The noun marked as the subject (*güller*) automatically receives the ARG0 label. As there are no dependents linked to it, the other tokens are not assigned any argument labels.

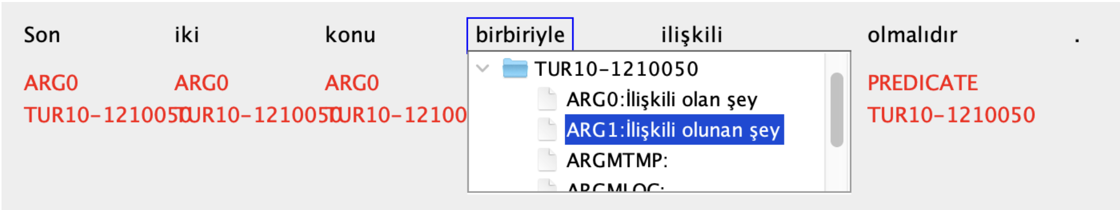

Figure 6: Interface for manual argument selection. Clicking on a word reveals frame file annotations from TRopBank v2.0 for each predicate. Annotators assign the appropriate argument label, displayed beneath the word along with its corresponding WordNet ID. To remove an argument label, the annotator clicks the predicate's ID, marking the word as "NONE."

For objects, tokens connected to the predicate with the OBJ relation were automatically labeled as ARG1. While this approach rarely resulted in over-generation, it sometimes caused under-generation. In PropBank, the scope of what constitutes an argument is more semantically driven, often including elements marked as OBL (oblique) or ADV/ADVCL (adverbial/adverbial clause) in the dependency layer. However, our automatic labeling algorithm was not designed to capture these cases comprehensively, and these arguments were manually added during the argument selection process.

Although automatic labeling reduced the amount of manual work, it was not entirely foolproof due to several challenges. The first issue arose from errors in the dependency layer: faulty dependency relations led to incorrect argument tags, which had to be corrected during manual annotation. Additionally, when the dependents of long noun phrases were not correctly linked to the head, only part of the phrase was labeled automatically, making the task of argument labeling more tedious for annotators.

Another challenge involved coordinated sentences. The algorithm only processed phrases directly linked to the predicate marked as the ROOT, resulting in automatic labeling for the arguments of only one predicate. The remaining arguments in the coordinated structure had to be annotated manually.

3.3 Argument selection

Following the automatic labeling of ARG0 and ARG1, annotators manually reviewed each sentence to identify and annotate missing arguments and correct any errors introduced by the automatic labeling algorithm. As discussed earlier, errors in lower layers—such as morphological and semantic disambiguation—also required manual correction. Since the same annotation interface, StarDust (Yenice et al., 2022), was used across all layers, the correction process was relatively straightforward for the annotators.

Figure 6 presents the interface used for manual argument selection. Clicking on a word displays the relevant frame file annotations from TRopBank v2.0 for each predicate in the sentence. Annotators open the frame file with the matching ID and assign the appropriate argument label. Both the selected label and the corresponding WordNet ID are then displayed beneath the word. If an argument label needs to be removed, the annotator clicks on the predicate's ID in the selection panel, marking the word as "NONE."

Following our choice explained earlier, in this iteration, only matrix clauses were annotated for arguments. The arguments of coordinated sentences were included in the annotation, while embedded clauses, even those with finite verbs, were not annotated separately. This decision aimed to avoid confusion, minimize time loss, and eliminate the need to modify the interface to accommodate cases where a single word plays multiple roles. Specifically, phrases often function as part of a larger constituent in the matrix clause while simultaneously carrying a distinct role in an embedded clause, which would complicate the annotation process. We plan to overcome this issue in the final phase of our project, which will be the construction of AMR structures for each sentence annotated for predicates and arguments. The arguments from the main clauses will be automatically integrated into the AMRs, while the arguments from embedded clauses will be added manually using the AMR interface.

Another decision we made was to exclude non-core arguments from our annotation process. This choice aimed to prevent redundancy, as a FrameNet layer will be incorporated before the construction

Dataset	**ARG0**	**ARG1**	**ARG2**	**ARG3**
Atis	995	16,090	698	0
Tourism	1,279	6,259	142	0
FrameNet	1,211	3,001	159	8
Penn TreeBank	18,283	31,359	756	25
KeNet	11,811	24,797	1,267	42
Total	**32,542**	**76,983**	**2,853**	**73**

Table 1: Distribution of core arguments (ARG0–ARG3) across the datasets. ARG0 typically marks agents or causers, while ARG1 often denotes patients or themes. ARG2 and ARG3 represent more specialized roles depending on the verb's semantics, with fewer occurrences in most datasets.

of the AMRs. The ARGM arguments typically convey information related to time, location, manner, and purpose, all of which are also captured within frame elements. Given that we have expanded upon the previous Turkish FrameNet project (Marsan et al., 2021) to encode frame information for each predicate, we opted to integrate this information at a later stage.

While Table 1 presents the number of annotated core arguments in each dataset, Table 2 details the counts of sentences labeled with verbal predicates, alongside those further annotated for arguments. Notably, across all datasets, a significant portion of sentences lacks core arguments. This is particularly evident in the Tourism dataset, which exhibits a lower ratio of arguments compared to predicates.

Among the labeled arguments, ARG1 is the most prevalent category. This trend highlights the frequent omission of subjects in Turkish, as indicated by the lower counts of ARG0s. The occurrences of ARG2 and ARG3 are comparatively minimal, aligning with linguistic expectations. This distribution may also be influenced by the limitations associated with missing argument specifications in the TRopBank frame files.

Dataset	**argument**	**predicate**
Atis	2,568	3,036
Tourism	3,717	7,166
Penn TreeBank	9,322	12,152
FrameNet	1,877	2,484
KeNet	9,385	11,839

Table 2: Number of sentences labeled for verbal predicates and core arguments across five datasets. The "predicate" column shows the total number of sentences with an identified verbal predicate, while the "argument" column indicates how many of those were further annotated with core arguments.

This issue encountered during this process was due to an annotation choice in the TRopBank project, which constructed frame files based on the definitions of WordNet synsets rather than on individual verbs. This approach resulted in numerous exceptional cases where certain verbs were missing specific argument definitions. While these issues could be addressed by incorporating the missing arguments for each verb, we opted not to implement these changes in this iteration due to time constraints. However, refining the previous project is essential for large-scale annotation projects like this one.

4 Discussion

This paper presented a semi-automated approach to annotating argument structures in Turkish datasets, focusing on verbal predicates and core arguments. Five datasets were annotated with a total of 25,580 sentences, with ARG0 and ARG1 assignments facilitated through dependency-based pre-labeling. Some challenges arose from dependency errors, passive structures, and unaccusative predicates, which required careful manual review to ensure alignment with TRopBank v2.0's frame files. We suggest that these frame files are updated to accommodate specific argument requirements of some Turkish verbs.

As stated earlier, this PropBank annotation is part of a larger, multi-layer NLP project aimed at enhancing Turkish semantic resources. In future work, each sentence will be annotated for frame elements according to the semantic frame of its predicate. Later on, the annotated core arguments will be integrated into AMR structures with additional layers, including subordinate clauses.

By focusing on precise predicate and core argument labeling in matrix clauses, this project lays the foundation for building high-quality AMRs for Turkish. These resources will not only support advanced NLP tasks such as semantic parsing but will also contribute to broader research efforts in multilingual natural language understanding.

References

Eneko E Agirre, Izaskun Aldezabal, Jone Etxeberria, and Eli Pociello. 2006. A preliminary study for building the basque propbank. In *Proceedings of the 5th International Conference on Language Resources and Evaluations (LREC)*.

Koray Ak, Cansu Toprak, Volkan Esgel, and Olcay Taner Yildiz. 2018. Construction of a turkish proposition bank. *Turkish Journal of Electrical Engineering and Computer Sciences*, 26(1):570–581.

Koray Ak and Olcay Taner Yıldız. 2019. Automatic Propbank generation for Turkish. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 33–41, Varna, Bulgaria. INCOMA Ltd.

Izaskun Aldezabal, María Jesús Aranzabe, Arantza Díaz de Ilarraza Sánchez, and Ainara Estarrona. 2010. Building the basque propbank. In *LREC*. Citeseer.

Özge Bakay, Özlem Ergelen, Elif Sarmış, Selin Yıldırım, Bilge Nas Arıcan, Atilla Kocabalcıoğlu, Merve Özçelik, Ezgi Sanıyar, Oğuzhan Kuyrukçu, Begüm Avar, et al. 2021. Turkish wordnet kenet. In *Proceedings of the 11th global wordnet conference*, pages 166–174.

Özge Bakay, Özlem Ergelen, and Olcay Taner Yıldız. 2019. Integrating turkish wordnet kenet to princeton wordnet: The case of one-to-many correspondences. In *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–5. IEEE.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D Hwang, and Martha Palmer. 2014. Propbank: Semantics of new predicate types. In *LREC*, pages 3013–3019.

Claire Bonial, Jena Hwang, Julia Bonn, Kathryn Conger, Olga Babko-Malaya, and Martha Palmer. 2012. English propbank annotation guidelines. *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder*, 48.

Neslihan Cesur, Aslı Kuzgun, Mehmet Kose, and Olcay Taner Yıldız. 2024. Building annotated parallel corpora using the atis dataset: Two ud-style treebanks in english and turkish. In *Proceedings of the 17th Workshop on Building and Using Comparable Corpora (BUCC)@ LREC-COLING 2024*, pages 104–110.

Razieh Ehsani, Ercan Solak, and Olcay Taner Yildiz. 2018. Constructing a wordnet for turkish using manual and automatic annotation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(3):1–15.

Katri Haverinen, Jenna Kanerva, Samuel Kohonen, Anna Missilä, Stina Ojala, Timo Viljanen, Veronika Laippala, and Filip Ginter. 2015. The finnish proposition bank. *Language Resources and Evaluation*, 49:907–926.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Neslihan Kara, Deniz Baran Aslan, Büşra Marşan, Özge Bakay, Koray Ak, and Olcay Taner Yıldız. 2020. Tropbank: turkish propbank v2. 0. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2763–2772.

Paul Kingsbury and Martha Palmer. 2003. Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, volume 3. Citeseer.

Paul R Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *LREC*, pages 1989–1993.

Aslı Kuzgun, Neslihan Cesur, Bilge Nas Arıcan, Merve Özçelik, Büşra Marşan, Neslihan Kara, Deniz Baran Aslan, and Olcay Taner Yıldız. 2020. On building the largest and cross-linguistic turkish dependency corpus. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–6. IEEE.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Büsra Marsan, Starlang Yazılım Danısmanlık, Neslihan Kara, Merve Ozçelik, Bilge Nas Arıcan, Neslihan Cesur, Aslı Kuzgun, Ezgi Sanıyar, Oguzhan Kuyrukçu, and Olcay Taner Yıldız. 2021. Building the turkish framenet. *South African Centre for Digital Language Resources (SADiLaR) Potchefstroom, South Africa*, page 118.

Martha Palmer, Olga Babko-Malaya, Ann Bies, Mona T Diab, Mohamed Maamouri, Aous Mansouri, and Wajdi Zaghouani. 2008. A pilot arabic propbank. In *LREC*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Gözde Gül Sahin. 2016. Framing of verbs for turkish propbank. *Turkish Computational Linguistics*, pages 3–9.

Gözde Gül Şahin. 2018. Verb sense annotation for turkish propbank via crowdsourcing. In *Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2016, Konya, Turkey, April 3–9, 2016, Revised Selected Papers, Part I 17*, pages 496–506. Springer.

Gözde Gül Şahin and Eşref Adalı. 2018. Annotation of semantic roles for the turkish proposition bank. *Language Resources and Evaluation*, 52:673–706.

Umut Sulubacak, Tugba Pamay, and Gülsen Eryigit. 2016. Imst: A revisited turkish dependency treebank. *Proceedings of TurCLing*, pages 1–6.

Nianwen Xue. 2006. A chinese semantic lexicon of senses and roles. *Language resources and evaluation*, 40:395–403.

Nianwen Xue and Martha Palmer. 2009. Adding semantic roles to the chinese treebank. *Natural Language Engineering*, 15(1):143–172.

Arife B Yenice, Neslihan Cesur, Aslı Kuzgun, and Olcay Taner Yıldız. 2022. Introducing stardust: A ud-based dependency annotation tool. In *Proceedings of the 16th Linguistic Annotation Workshop (LAW-XVI) within LREC2022*, pages 79–84.

Olcay Taner Yıldız, Begüm Avar, and Gökhan Ercan. 2019. An open, extendible, and fast turkish morphological analyzer. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1364–1372.

Wajdi Zaghouani, Mona Diab, Aous Mansouri, Sameer Pradhan, and Martha Palmer. 2010. The revised arabic propbank. In *Proceedings of the fourth linguistic annotation workshop*, pages 222–226.