Natural Language Engineering (2022), 1–30
doi:10.1017/S1351324922000183

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

Abstract meaning representation of Turkish

Elif Oral[1], Ali Acar[1] and Gülşen Eryiğit[1,2,*]

[1]NLP Research Group, Faculty of Computer and Informatics, Istanbul Technical University, Istanbul, Turkey and
[2]Department of Artificial Intelligence and Data Engineering, Istanbul Technical University, Istanbul, Turkey
*Corresponding author. E-mail: gulsen.cebiroglu@itu.edu.tr

(Received 9 May 2021; revised 21 March 2022; accepted 28 March 2022)

Abstract

Abstract meaning representation (AMR) is a graph-based sentence-level meaning representation that has become highly popular in recent years. AMR is a knowledge-based meaning representation heavily relying on frame semantics for linking predicate frames and entity knowledge bases such as DBpedia for linking named entity concepts. Although it is originally designed for English, its adaptation to non-English languages is possible by defining language-specific divergences and representations. This article introduces the first AMR representation framework for Turkish, which poses diverse challenges for AMR due to its typological differences compared to English; agglutinative, free constituent order, morphologically highly rich resulting in fewer word surface forms in sentences. The introduced solutions to these peculiarities are expected to guide the studies for other similar languages and speed up the construction of a cross-lingual universal AMR framework. Besides this main contribution, the article also presents the construction of the first AMR corpus of 700 sentences, the first AMR parser (i.e., a tree-to-graph rule-based AMR parser) used for semi-automatic annotation, and the evaluation of the introduced resources for Turkish.

Keywords: Abstract meaning representation; Semantic representation; Turkish

1. Introduction

Semantic representation is a formal structure that represents the meaning of language constituents. Tasks such as named entity recognition, semantic relation extraction, and co-reference resolution are considered as semantic extraction tasks, yet they can only extract a small part of sentence meaning and are not capable of representing the whole. At the sentence level, meaning representation frameworks aim to annotate sentences with their whole sentence meaning. Despite success in semantic extraction tasks listed above, there is still a lack of a standard on semantic representation frameworks to represent sentence-level meaning, and the field is still an active research area (Koller, Oepen, and Sun 2019; Xue *et al*. 2019, 2020; Žabokrtský, Zeman, and Ševčková 2020).

There are several semantic representation frameworks in the literature, each of which has its own characteristics. Oepen *et al*. (2019) categorizes semantic annotations under three types based on the nature of the relationship between the linguistic surface signal and the nodes of the graphs. In some meaning representation frameworks such as Groningen meaning bank (Basile *et al*. 2012) and Universal Conceptual Cognitive Annotation (Abend and Rappoport 2013), the represented meaning is beyond a sentence and sometimes goes as far as paragraph level. Recently introduced by Banarescu *et al*. (2013), abstract meaning representation (AMR) has become highly popular for semantic representations (Xue *et al*. 2020) and two consecutive SemEval tasks (May 2016; May and Priyadarshi 2017) have focused on it. AMR is a sentence-level semantic representation framework that represents sentences as directed acyclic graphs where nodes are the concepts (viz., predicate frames, words, or special keywords) within a sentence and edges are the semantic relations between these. This representation considers all aspects of meaning in sentences, such as named entities, semantic relations, temporal entities, and co-references. Rather than syntax, it focuses on only the meaning of sentences; in other words, AMR graphs contain sentence components that only contribute to the sentence meaning.

AMR is firstly designed for English and not intended to be an interlingua. However, studies show that structurally aligning English AMRs with their counterparts in other languages are possible by addressing language-specific issues. Morphologically rich languages (MRLs) posing interesting challenges for almost all natural language processing tasks also reveal interesting design problems for AMRs. A single word in an MRL may sometimes express a quite long English sentence due to the rich morphological structure and meanings carried by affixes. This reveals the synthesis of multiple concepts of an AMR graph from a single word. In this article, we present an AMR framework for such a language: Turkish, which is a prominent example of MRLs. Turkish is the most widely spoken and studied language within Turkic languages, and Turkish may be seen as the representative of this language family spoken by nearly 200M people spread over a wide geographical area. Turkish is an agglutinative language and has a very rich morphological structure. In the literature, there also exist alternative meaning representations offering more flexibility for representing MRLs (such as Type 1 representations in Oepen *et al.* 2019). The motivation behind our choice is the increasing interest in AMR in recent years and other recent efforts for representing MRLs with AMR. In addition, the availability of a Turkish PropBank is a facilitating factor for starting AMR studies on this language.

This article introduces the first AMR representation framework for Turkish, which poses diverse challenges for AMR due to its typological differences compared to English. In the literature, there exist other studies focusing on handling morpho-semantics for AMR, and the proposed solutions in this article are linked to these previous studies on other languages with similar linguistic phenomena (e.g., Chinese, Portuguese, Spanish, Korean, Vietnamese) in order to pave the way for a cross-lingual universal AMR framework. The contributions of the article are as follows:

- the first formal meaning representation for the Turkish language,
- the first AMR representation framework for Turkish: the introduction of the AMR related language-specific constructions of Turkish and the proposed AMR schema, as well as an annotation guideline as additional material,[a]
- the first Turkish Abstract Meaning Representation Corpus containing 700 AMR-annotated sentences,
- the first Turkish AMR parser developed to accelerate the human annotation process with a semi-automatic approach (with a Smatch score of 60%).

The article is structured as follows: Section 2 provides the related works and briefly presents the AMR fundamentals, Section 3 introduces the Turkish AMR representation framework by discussing Turkish-specific constructions, Section 4 presents the stages of the corpus construction: our semi-automated AMR annotation approach, our rule-based AMR parser, and the Turkish AMR corpus, and finally, Section 5 gives the conclusion.

2. Background and related work

AMR is a knowledge-based meaning representation heavily relying on frame semantics (e.g., resources such as PropBank Frames or Framenet) for linking predicate frames and entity

[a] This article provides only the challenging Turkish-specific constructions in terms of AMR. An extended guideline https://github.com/amr-turkish/turkish-amr-guidelines providing a wide variety of samples exemplifying both these and parallel or easily mappable structures to English AMR, which is available from https://github.com/amrisi/amr-guidelines.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

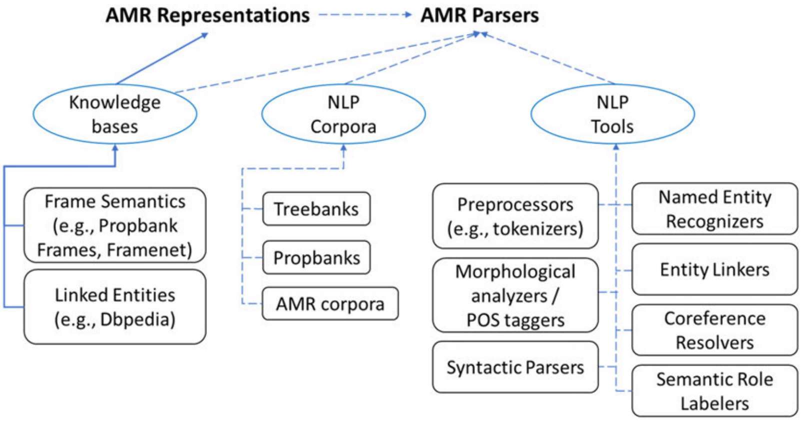


Figure 1. AMR interaction with knowledge bases and other NLP resources. (Dashed lines represent optional interactions.)

knowledge bases such as DBpedia for linking named entity concepts. While AMR representations carry mandatory links to these knowledge bases, AMR parsers optionally use these and AMR representations. Also, AMR parsers often make use of additional NLP resources, if available, to construct the AMR structures from natural language sentences (Flanigan *et al.* 2014; Werling, Angeli, and Manning 2015; Zhou *et al.* 2016; Goodman, Vlachos, and Naradowsky 2016; Damonte, Cohen, and Satta 2017) (Figure 1). These resources may be either corpora annotated at different levels (e.g., PropBanks (Palmer *et al.* 2005); (Xue and Palmer 2009), Dependency Treebanks (Nivre *et al.* 2017) and AMR-annotated corpora, e.g., LDC AMR corpora) or other NLP tools such as tokenizers, parts-of-speech taggers, syntactic analyzers, named-entity recognizers, linkers, or semantic role labelers.

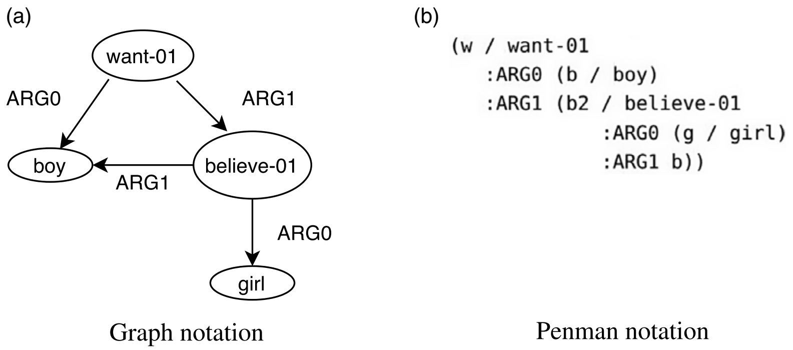
AMR offers a single framework where "balkanized" semantic annotations (e.g., named entities, co-reference, semantic relations) are gathered in the same representation. Its focus is on the meaning of sentences rather than syntax. An AMR graph does not represent the words that do not contribute to the sentence meaning. This results in a single graph for sentences with similar meanings. Figure 2 gives such a representation for the sentences: "The boy wants the girl to believe him." and "The boy wants to be believed by the girl." This figure provides the same representation in two different notations (the graph notation and the Penman notation (Kasper 1989)) used throughout the article. The AMR annotation highly depends on predicate–argument structures defined in The Proposition Bank, shortly PropBank (Palmer *et al.* 2005) where the senses of predicates alongside their argument structure are contained. *want-01* and *believe-01* in Figure 2 are the PropBank frame names for the sentence predicates. Similarly, *ARG0* and *ARG1* are the defined arguments of these frames within PropBank.

In an AMR graph, nodes are called *concepts* and edges represent *relations* between these concepts. Concepts are either words in sentences (named as lexical concepts), PropBank framesets, or special keywords (denoting special entity types, quantities, or logical conjunctions) coming from AMR specification (Banarescu *et al.* 2013).

AMR relations describe semantic dependencies between concepts. There are approximately more than 100 relations in AMR such as frame arguments and general semantic relations. Inverse of these relations is also available, such as *:arg0-of* or *:cause-of*. AMR enables a concept to participate in multiple relations. A word in a sentence might be an argument of more than one predicate. For example, in Figure 2, the boy is the argument of both predicates *want-01* and *believe-01*. This phenomenon is called *reentrancy*.

AMR has attracted the attention of many researchers in the NLP community (Bos 2016; Žabokrtský *et al.* 2020) and has been used for several applications including summarization

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press



```
(w / want-01
   :ARG0 (b / boy)
   :ARG1 (b2 / believe-01
             :ARG0 (g / girl)
             :ARG1 b))
```

Graph notation

Penman notation

Figure 2. A sample AMR representation in graph and Penman notations.

(Dohare, Karnick, and Gupta 2017; Liu *et al.* 2018a; Liao, Lebanoff, and Liu 2018), text generation (Song *et al.* 2016; 2018; Damonte and Cohen 2019; Wang, Wan, and Jin 2020a; Mager *et al.* 2020; Zhao *et al.* 2020; Fan and Gardent 2020; Wang, Wan, and Yao 2020b; Bai, Song, and Zhang 2020; Jin and Gildea 2020), machine translation (Song *et al.* 2019), sentence compression (Takase *et al.* 2016), event extraction (Huang *et al.* 2016; Li *et al.* 2020), human–robot interaction (Bonial *et al.* 2020), and natural language understanding in dialogue systems (Bonial *et al.* 2020; Bonn *et al.* 2020).

AMR studies mostly focus on English, for which AMR is originally designed, and required knowledge bases are available. However, in recent years, there have been quite a lot of studies reporting the adaptation of AMR to non-English languages. Li *et al.* (2016) introduced an annotation specification for Chinese AMR (known as CAMR), which specifies differences between English and Chinese and releases a corpus containing 1562 AMR-annotated sentences from the Chinese translation of the novel "The Little Prince." The project continues with the annotation of more sentences from the CTB Chinese treebank (Xue *et al.* 2005). While Chinese AMR corpus (Li *et al.* 2019) is the largest among non-English languages with around 10K sentences, it is still very small when compared to the English AMR corpus with around 60K AMR-annotated sentences. Other studies reporting on the adaptation of AMR to other languages include Migueles-Abraira, Agerri, and Diaz de Ilarraza (2018) for Spanish, Xue *et al.* (2014) for Czech, Anchiêta and Pardo (2018a), Sobrevilla Cabezudo and Pardo (2019) for Brazilian Portuguese, Choe *et al.* (2019a, 2020) for Korean, and Linh and Nguyen (2019) for Vietnamese. The corpora released by these studies are very modest in size (ranging between 50 and 1.5K sentences) compared to English and Chinese. Furthermore, Zhu *et al.* (2019) present a cross-lingual semantic representation that can be described as a simplified version of AMR since it expresses only essential semantic features and other important features of a sentence, such as predicate roles and linguistic relations. Feng (2021) proposes a modified version of AMR by replacing PropBank arguments with predefined roles mapped to proper argument relations. The study aims to overcome some PropBank related issues such as fine-grained sense disambiguation and high start-up costs and shows that this modification reduces annotation times and increases parsing accuracy.

AMR parser development is another branch of AMR research. There are four significant approaches in AMR parsing including (i) graph-based approaches, (ii) transition-based approaches, (iii) Seq2seq-based approaches, and (iv) sequence-to-graph (seq2graph) approaches. Graph-based approaches (Flanigan *et al.* 2014, 2016; Werling *et al.* 2015; Foland and Martin 2017; Lyu and Titov 2018; Zhang *et al.* 2019a) first identify concepts in sentences and then construct the possible edges. Transition-based parsing (Wang, Xue, and Pradhan 2015a; Damonte *et al.* 2017; Ballesteros and Al-Onaizan 2017; Wang and Xue 2017; Guo and Lu 2018; Liu *et al.* 2018b; Peng, Gildea, and Satta 2018; Naseem *et al.* 2019; Astudillo *et al.* 2020) uses a series of actions that process a sentence and generate an AMR graph by either inserting a new node or adding a new

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

edge. Seq2seq-based approaches use sequence-to-sequence models for AMR parsing by linearizing AMR graphs (Barzdins and Gosko 2016; Konstas *et al.* 2017; Van Noord and Bos 2017; Xu *et al.* 2020; Blloshmi, Tripodi, and Navigli 2020). As the last approach, sequence-to-graph approaches build the AMR graphs incrementally in a way that the models jointly predict new nodes along with their connections at each time step (Zhang *et al.* 2019b; Cai and Lam 2020).

Although many AMR parsing studies continue on English, there are significant efforts for non-English languages. Damonte and Cohen (2018) introduce a multi-lingual AMR parser which adapts a transition-based English AMR parser trained on automatically annotated data for Italian, Spanish, German, and Chinese. Blloshmi *et al.* (2020) use several transfer learning techniques for the multi-lingual AMR parsing. Brazilian Portuguese is another language in which AMR parsing studies actively continue. Anchiêta and Pardo (2018b) present a rule-based parser, and Anchiêta and Pardo (2020) present an aligner enriched with word representations for this language.

As stated above, researchers use several resources during AMR studies for either AMR annotations or parser development. Similarly, several Turkish resources and tools support our study; these are the Turkish PropBank (Şahin and Adalı 2018), dependency parser (Eryiğit, Nivre, and Oflazer 2008), IMST dependency treebank (Sulubacak, Eryiğit, and Pamay 2016), and ITU NLP pipeline (Eryiğit 2014).

3. Turkish AMR

Turkish is a morphologically rich and agglutinative language. This nature of the language allows the attachment of multiple suffixes to the word lemmas, resulting in quite long words, sometimes corresponding to a whole sentence in English. Due to this complex structure, it is undeniable that the suffixes are one of the most important components of sentences with their ability to establish relationships between sentence constituents, that is, embedding the grammar into word level and constructing new words by the use of derivations. The fact that the suffixes have such functionalities causes differences in concept creation and relationship building stages of AMR: Derivational suffixes (DSs) produce new words by changing the base word in meaning and sometimes also changing its main parts-of-speech class, for example, nominals may easily turn into verbs or vice versa. This reveals the need for multiple AMR concepts for a single such word. Similarly, inflectional suffixes (ISs) may be attached to words in order to show some aspects of their grammatical functions, such as plurality and tense. One word may have multiple ISs carrying different meanings (e.g., the subject information) which may not be directly transformed into a single AMR concept; its corresponding AMR representation could be a complex AMR graph. While some suffixes describing relationships between constituents of a sentence should be mapped to proper AMR relations, some others having specific meanings need to be mapped to concepts. This mapping could be straightforward in some cases (please refer to the guideline[a] for full list), for example, the Turkish location case marker, the suffix *-de*, can be easily mapped to the relation *:location* or *:topic* in AMR according to the context. However, the majority of the suffixes cannot be mapped directly because of Turkish-specific constructions. In this section, we point out only the challenging Turkish-specific constructions in terms of AMR and our proposed solutions to create Turkish AMR representations parallel to English. The definition of Turkish grammar, the full list of possible suffixes, and their AMR mappings are left out of the scope of this article. A more detailed specification with further examples (also including straightforward mappings) has been prepared and shared with the researchers as a separate guideline.[a]

In contemporary everyday Turkish, words have about 3–4 four morphemes including the stem, such as in the word "görüştürüldü" (with separated morphemes as "gör-üş-tür-ül-dü" meaning *S/he is made to have an interview with someone*) which has 1 derivational, 2 voice (causative and passive), and 1 inflectional (past tense 3rd person singular) morphemes accordingly. Şahin (2016) states that according to the Turkish Language Association (TDK), there are 759 root verbs, 2380 verbs derived from nouns, and 2944 verbs derived from other root verbs via DSs. The functionality

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

of ISs in Turkish varies based on the class of the stem. They indicate the relationships between constituents of a sentence by marking case, possession, and number when they attach to nominals. On the other hand, when the stem is a verb, they express functional relations such as tense, person, and modality. Independent of their types, while some morphemes add a single meaning to the stem, others have more than one meaning.

Since AMR focuses on actions, predicates are one of the most important components. In the following subsections, we start by introducing the differences due to verbal structures and then continue with nominals.

3.1 Verbal derivation from nominals

Verbal derivation from nominals is a phenomenon frequently observed in Turkish. There exist more than 10 suffixes deriving verbs from nominals (shortly nominal verbs hereinafter); however, not all of them are very productive. The suffixes *-lA*, *-lAş*, *-lAn*[b] diverge from the others with their high productivity. They can be attached to a vast number of nominals and convert them to either direct or passive verbs. They can dynamically derive nominal verbs in daily use, and a native speaker easily understands them even though the resulting verbs (e.g., "eflatun-laş" (*to take lilac-color*)) do not take place in the dictionary. For the sake of simplicity, hereafter, we will use the abbreviations HPS for these highly productive suffixes and HPVs for nominal verbs derived with these.

Şahin (2016) claims that creating a nominal bank and linking nominal verbs with the entries from the nominal bank would be more appropriate for framing nominal verbs in general. However, in their follow-up study (Şahin and Adalı 2018), they suggest a different strategy and include the most frequent ones (excluding HPVs) to the Turkish PropBank, and for HPVs, they tried to solve this dynamic derivation issue by creating x-rooted frames as xlA, xlAş, xlAn where "x" represents the noun root. We also believe that producing frames for the most frequent nominal verbs[c] is a necessity due to the fact that some verb meanings formed over the years may be quite different than the nominal roots (to be detailed below). Although x-rooted frames seem an appropriate solution to incorporate such highly productive structures into the PropBank, we believe that this approach has its own shortcomings and does not suit AMR as a meaning representation. First of all, this approach treats all HPVs as the same. Thus, although they seem to be grammatically the same, there could appear differences between their argument structures.

In order to cover all possible verbs diverging from x-rooted frames (i.e., HPVs), a similar approach would be to add new frames into the PropBank; however, similar to Şahin (2016), we believe that this approach makes the verb framing process complicated and prone to framing mistakes. Additionally, AMR is interested in making events out of nouns and adjectives and represents these as root nodes of the graphs and sub-graphs. Finally, as we are looking for a graph that is easily readable by both humans and machines, we need meaningful concepts (rather than hardly understandable x-rooted frames) inside the AMR structure. A meaningful frame can be either created for the verb or selected from PropBank frames carrying the same meaning. Considering these, we believe that one should adopt a different approach for Turkish AMR.

To represent nominal verbs in general (excluding HPVs), we use their existing PropBank frames, if any, otherwise we create/suggest new frames for them. One should note that missing predicate frames are also encountered in other languages and solved by adding a -00 tag to AMR predicate concepts as a suggestion to be included later into the knowledge base (e.g., Banarescu

[b] The use of capital letters in the representation of suffixes is a tradition in Turkish NLP to depict the possible phonological changes under different circumstances (vowel/consonant harmony rules); A denotes "a" or "e", H: "ı", "i", "u", or "ü", and I: "ı" or "i", C: "c" or "ç". In this representation, a parenthesis used around a letter means that the use of that letter may be omitted under different phonological occurrences. For example, the suffix -(I)ş may be seen as -ş, -ış, or -iş.

[c] There also exist derivational suffixes which form verbs from verbs. We also suggest producing new frames for these.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

et al. 2012 English AMR spec 1.2 to OntoNotes). For frame creation, we follow the previous efforts for Turkish (Şahin 2016), and we create the new frames using the predicate framing editor introduced in Choi, Bonial, and Palmer (2010) according to the PropBank framing guidelines (Babko-Malaya 2005).

For the representation of HPVs, in order to avoid creating too many frames, we only create a new frame for an HPV if it satisfies all of the following conditions at the same time:

(1) The verb should exist within the Turkish dictionary,
(2) One should not be able to represent the verb with another verb frame from PropBank,
(3) One should not be able to represent the verb as the passive form of another verb frame from PropBank.

The remainder of this section explains the rationale for setting these conditions. Some verbs may gain additional meanings in time rather than the one added by the DS (e.g., the suffix -lAn generally adds the meaning of getting the thing expressed with the noun lemma), and the main reason that we expect a verb to be present in the Turkish dictionary to create a new frame (the 1st condition above) is that the dictionary lists all these (additional or main) meanings. For example, the verb "ev**len**" derived from "ev" (*home*) with the HPS *-lAn* means "to get married" and rarely used in daily life with its literal[d] meaning "to get a house." Thus according to the context, if this verb is used in its ordinary sense ("to get married"), its own frame should be used in AMR annotation. On the other hand, if it is used in its literal meaning, it should be treated differently, as detailed below. The suffix -lAn may be attached to almost every noun and can derive verbs dynamically. Although these verbs are grammatically correct, they are not frequently used in formal Turkish (and not included in the dictionary), but they are still meaningful for native Turkish speakers. For instance, someone may say "Arabalandım." (*I got a car.*) where the noun "araba" (*car*) is converted to "araba**lan**" (*to get a car*) to express that s/he purchased a new car in daily speech. As stated previously, creating a new frame for all dynamically formed HPVs is not feasible. We solve this issue by considering the meanings of such HPVs. For example, if a derived verb with this suffix means to have the item represented by that nominal, it is mapped to the frame "ol.4" (*to get*) (linked with the nominal concept) instead of creating a new frame. Similarly, if it means to become to the state of that nominal (e.g., "hüzünlenmek" (*to become sad*)), it is mapped to the frame "ol.2" (*to become*) (linked to the concept *sad*) instead of creating a new frame, although the verb exists within the dictionary (due to the violation of the 2nd condition above). Figure 3 shows two such HPVs. Since "güneşlen" appears in the dictionary but not in the Turkish PropBank and has a special meaning (*to sunbathe*) (different than the literal one added with the DS such as "to get a sun"), we suggest to create and use a new verb frame for it. On the other hand, "arabalanmak" is represented using the frame "ol.4" (*to get*) attached to the lexical concept "araba" (*car*) as explained above.

Another characteristic of the suffix *-lAn* is converting nominals, mostly adjectives, to passive verbs like "yasaklan" (*to be banished*) or "kurulan" (*to be dried*). Since AMR is only interested in verbs, not their passive forms, it is unnecessary to create new verb frames for such HPVs (3rd condition above). The point to note is that some verbs derived with *-lAn* can be used as both active and passive verbs. For example, the verb "av**lan**" (*to hunt*) is passive within the sentence "Balıklar ayı tarafından avlandı." (*The fish were hunted by the bear.*) whereas it is active in "Dişi aslan bozkırda avlandı." (*Female lion hunted in steppe.*).

[d] Hereinafter, the adjective "literal" is used to express the most basic sense obtained by the addition of a derivational suffix (with a predefined meaning).

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

Güzel bir sonbahar gününde *güneşleniyordu*.
beautiful - one - autumn - in the day- was sunbathing
S/he was sunbathing on a beautiful autumn day.

```
(g / güneşlen.00
   :ARG0 (o / o)
   :time (g2 / gün
            :time (d / date-entity
                     :season (s / sonbahar))
            :mod (g3 / güzel)))
```

Arabalandım.
got car - I
I got a car.

```
(o / ol.04
   :ARG0 (b / ben)
   :ARG1 (a / araba)
```

Figure 3. AMR representations for nominal verbs produced with -lAn.

(a)

Type	Example	Verb Frame
Noun	Kız **geleceğini** söyledi. the girl - **that she will come** - said *The girl said that she will come.*	gel.01 (come.01)
Adjective	**Konuşan** kişi benim babam. **who is speaking** - the person - my father *The person who is speaking is my father.*	konuş.01 (speak.01)
Adverb	**Sorulunca** cevap ver. **when you are asked** - answer *Answer when you are asked.*	sor.01 (ask.01)

Noun, adjective and adverb samples

(b)

Kız geleceğini söyledi.
the girl - that she will come - said
The girl said that she will come.

```
(d / de.01
   :ARG0 (k / kız)
   :ARG1 (g / gel.01
            :ARG0 k ))
```

AMR representation of 'geleceğini'

Figure 4. Nominalized verb samples.

3.2 Verbal nominalization

Nouns that invoke predicates are considered as one of the challenges of semantic annotation tasks. Unlike the other nominals, they give a sense of actions to a sentence part without any predicate. From the following clause "The boy's promise not to lie to his parents," it is understandable that the boy promised to his parents that he would not lie to them. The noun *promise* indicates an event, and *the boy*, *the parents*, and *the lying* are the arguments of this event, respectively. For English, studies use different sources for representing such constructions in semantic annotations. While semantic role labeling systems use Nominal Bank (NomBank) (Meyers *et al.* 2004) that provides frames for such nouns, English AMR uses sense-tagged verbs from OntoNotes (Weischedel *et al.* 2011). Similar to English, Turkish also has such nominals invoking predicates. The counterparts of the samples above (about promising) may be produced in Turkish (see guideline[a] for samples) and represented in parallel to English AMR. However, in addition to this phenomenon, several types of nominals (i.e., nouns, adjectives, adverbs) may be dynamically produced from verbs using suffixes. There exist different views of naming this as a derivational (Adomako 2012) or inflectional process (Göksel and Kerslake 2004). Stems provide the direct link between verbs and nominalized verbs, which allows to directly link these to their related verb frames in the PropBank. Figure 4 provides such examples. In Figure 4b, the nominalized verb "geleceğini" (*that s/he is going to come*) is derived from the verb "gel" (*to come*) by the subordinating suffix -*AcAk* and then inflected by the 3rd person possessive suffix. As shown in the example, we easily annotate it with the verb frame ("gel.01").

Although it is straightforward to link the nominalized verbs to related verbs in Turkish, some phenomena (i.e., adverbial subordination and headless relative constructions) pose some issues that one needs to handle in terms of AMR.

Table 1. Some suffixes forming converbs and their corresponding AMR relations

Suffixes	Relation	Example
-Ip	:prep-after	Arabaya bin**ip** gitti to the car - by getting into - went *S/he got into the car and left*
-ArAk	:prep-by	Ağla**yarak** yanımıza geldi crying - next to us - s/he came *S/he came to us crying*
-mAdAn	:prep-without	Bugün kahvaltı yap**madan** okula gitti Today - breakfast - without doing- to school - went *S/he went to school today without having breakfast*
-dHkçA	:prep-as	Tren hızlan**dıkça** ağaçlar sıklaşmaya başladı train - as moved faster - trees - denser - got *As the train speeded up, the trees were getting denser*
-HncA	:time	Eve gid**ince** beni ara to home - when you go - me - call *Call me when you go home*
-kAn	:prep-while	Telefonla konuşur**ken** kapı çaldı with the phone - while speaking - the door - rang *The door rang while I was speaking on the phone*

3.2.1 Non-finite adverbial subordination

There exist finite and non-finite adverbial clauses in Turkish, where the non-finite forms are more numerous and more widely used (Göksel and Kerslake 2004). The subordinate verb forms in non-finite adverbial clauses are called converbs, and converbial suffixes form these by transforming verbs into adverbs. We map these suffixes to proper AMR relations to indicate the relationships between sentence constituents. Table 1 provides the mapping of some such suffixes to AMR relations. However, we should point out that the meanings of these suffixes may differ within different contexts. Therefore during the annotations, they should be mapped to proper relations accordingly. All such relations in English AMR start with the prep-X prefix, which holds for prepositions. One should note that these are rather postpositions in Turkish carrying the same meaning with X. Korean AMR studies (Choe *et al.* 2019b, 2020) also discuss adverbial subordination in general. As opposed to these, we prefer to use the relation names as they are, to be in parallel with English AMR rather than renaming them as postp-X. We believe these prefixes are syntactic issues rather than semantic and should be removed in a universal schema.

In some cases, the needed relation type may not exist in AMR predefined relation list. For example, in Table 1, we define new relations *:prep-while* and *:prep-after* because of the absence of any relationship covering the meaning of these suffixes.

3.2.2 Headless relative constructions

Headless relative constructions are relative clauses without an explicit noun head implicitly inferred most of the time. Chinese AMR studies (Li *et al.* 2016, 2019) also investigate this phenomenon and our proposed solution originates from these. Turkish is a pro-drop language, and the omission of object or subject pronouns is possible in the case of nominalized verbs. In the

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

```
Çok kitap okuyanlar yarışmayı kazandı.
many - book - who reads - the competition - won
Those who read a lot of books won the competition.

(k / kazan.01
   :ARG0 (p / person
                :ARG0-of (o / oku.01
                              :ARG1 (k2 / kitap
                                         :quant (ç / çok ))))
   :ARG1 (y / yarış.01))
```

Figure 5. Annotation of omitted pronouns in nominalized verbs.

AMR representations, we add the omitted pronouns, which can be either a person, a thing, or an event, according to the context. Figure 5 provides such an example where the concept "person" is added since the readers pointed by the pronoun *those* have to be human. We use the concept "thing" to depict the omitted pronouns referring to objects, events, or ideas.

3.3 Verbal inflection

The verbal inflection in Turkish occurs in many ways, such as negative markers, tense/aspect/modality markers, person markers, and voice markers. This section investigates the last three of these phenomena that require special consideration for AMR.

3.3.1 Person marking and the null subject

In Turkish, a predicate must contain a person marker. The doer of an event that the predicate represents is revealed by the personal suffixes concatenated to the end of the predicates. The explicit usage of the subject is optional. In the sentence "Kitap oku**yorum**." (*I am reading a book.*), the suffix "*-m*" (the last letter of the verb which stands for *I*) indicates who is reading the book. This type of subject usage is highly common in Turkish and called "null-subject." We should also note that personal markers may also appear on nominalized verbs (Figure 4b). In AMR representation, in the case of a null subject, we accept the subject indicated by the personal suffix (depicted with a nominative pronoun in the AMR notation parallel to English) as the related argument of the predicate. It is worth noting that, in case of a missing explicit subject within the sentence, the absence of any person marker on the predicate indicates the 3rd person singular subject. Figure 3 sample on the left provides such a case where ":ARG0 (o/o)" is the omitted pronoun s/he.

Spanish is also a null-subject language and Migueles-Abraira *et al.* (2018) discuss this feature in terms of AMR. However, contrary to Turkish, Spanish has gender and they need to handle the 3rd person null subjects in a different manner. Although not as much as Turkish, Brazilian Portuguese is also seen as a partial null-subject language (Holmberg, Nayudu, and Sheehan 2009), and Anchiêta and Pardo (2018a) discuss this situation in terms of AMR. In the case of null subject, they also fill the related argument implicitly inferred.

3.3.2 Modality

Modality is the phenomenon in which possible situations are discussed. In Turkish, modality suffixes are used to express modalities such as possibility, obligation, and permission. English AMR simply represents syntactic modals using predicate frames such as *possible-01*, *likely-01*, *obligate-01*, *permit-01*, and *recommend-01*. Linh and Nguyen (2019) also mention syntactic modalities for Vietnamese but do not follow the grouping of modalities proposed by the English AMR.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

Table 2. Modality samples

Modality sense	Turkish suffix	Verb frame	Example
Possibility	-Abil,	mümkün.01	Çekilişi kazan**abil**irim (I may won the lottery)
Permission	-Abil	ver.09	Polis, “gid**ebil**irsiniz”, dedi (Policeman said: “you can go”)
Necessity	-mAlI	gerek.01	Enfekte olmamak için maske tak**malı**sın (You must wear mask not to get infected)
Obligation	-mAlI	zorla.01	Ödevimi yap**malı**ym (I have to do my homework)
Recommendation	-mAlI	öner.01	Bu filmi kesinlikle izle**meli**sin (You should definitely watch this movie)

(a)

Bu soruları çöz**ebil**irsin.

these questions - solve + (AbIl) modality case + you

You can solve these questions.

```
(m / mümkün.01
     :ARG1 (ç / çöz.02
                 :ARG0 (s / sen)
                 :ARG1 (s2 / soru
                             :mod (b / bu))))
```

Possibility in Turkish AMR

(b)

Bu soruları çöz**ebilmeli**sin.

these questions - solve + (AbIl) modality case

+ (mAlI) modality case + you

You have to be able to solve these questions.

```
(z / zorla.01
   :ARG2 (m / mümkün.01
               :ARG1 (ç / çöz.02
                          :ARG0 (s / sen)
                          :ARG1(s2 / soru
                                    :mod (b / bu)))))
```

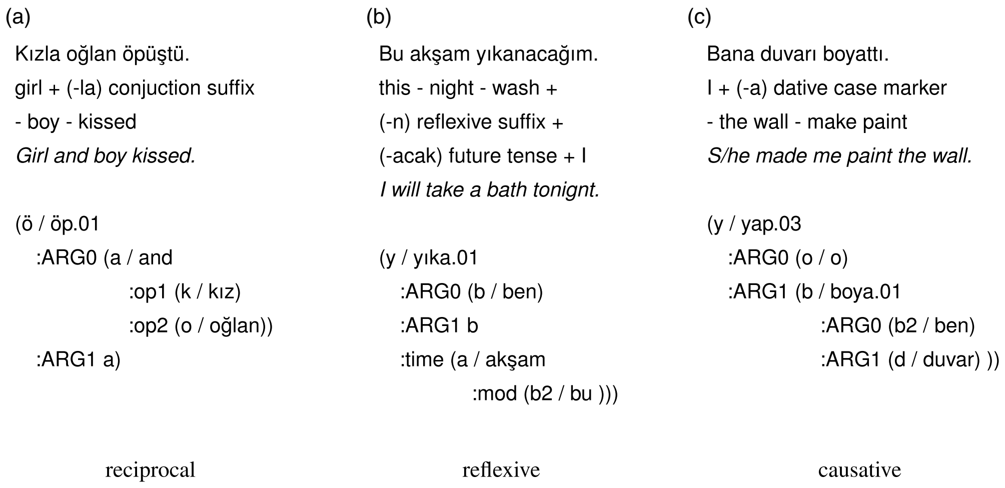
-Abil and -mAlI are attached to the verb *gel* (to come)

Figure 6. Modality representation in Turkish AMR.

For Turkish, we map Turkish modality suffixes to some selected predicates without changing the sentence meaning in parallel with English AMR. This seems straightforward, but there are considerations to be made. Firstly, Turkish does not have a predicate for the sense of possibility like in English. While the English PropBank provides a frame for the sense of possibility, the Turkish PropBank does not. Therefore, we create a special frame “mümkün.01” (*possible*) which has one argument :ARG1 to represent the possible event (in Figure 6a). Secondly, modality markers may carry more than one sense, and as a result, one could map them to more than one predicate according to the context.

Table 2 shows some common modality suffixes with their corresponding verb frames. Sentences in the first two rows and the last three rows have the same modality markers (*-Abil* and *-mAlI*), although their senses are entirely different. Furthermore, a verb can have more than one modality suffix at the same time (Figure 6b). In this case, each suffix should be mapped to a proper predicate separately and represented in AMR. One should note that the expression of modalities is not provided only by modality markers; there are nominals which give a modality expression to the sentence. To make annotation consistent, we map these nominals to the same frames with the modality markers.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press



(a)

Kızla oğlan öpüştü.
girl + (-la) conjuction suffix
- boy - kissed
Girl and boy kissed.

```
(ö / öp.01
  :ARG0 (a / and
           :op1 (k / kız)
           :op2 (o / oğlan))
  :ARG1 a)
```

reciprocal

(b)

Bu akşam yıkanacağım.
this - night - wash +
(-n) reflexive suffix +
(-acak) future tense + I
I will take a bath tonignt.

```
(y / yıka.01
  :ARG0 (b / ben)
  :ARG1 b
  :time (a / akşam
          :mod (b2 / bu )))
```

reflexive

(c)

Bana duvarı boyattı.
I + (-a) dative case marker
- the wall - make paint
S/he made me paint the wall.

```
(y / yap.03
  :ARG0 (o / o)
  :ARG1 (b / boya.01
           :ARG0 (b2 / ben)
           :ARG1 (d / duvar) ))
```

causative

Figure 7. AMR representation of the verb voice structures.

3.3.3 Voices

Turkish has four voice structures (viz., reciprocal, reflexive, causative, and passive) constructed through voice suffixes (VSs) attached to verbs. Voices describe the relationship between the predicate and the subject. As a result, when a verb takes a VS, its arguments' number and type may change or stay the same (Göksel and Kerslake 2004). The change of argument structure of verbs affects their AMR representation as expected, which brings some issues. The Turkish PropBank does not have frames for such verbs and uses their stems with some additional features to represent them. From the AMR point of view, there are two possible solutions to address the issue. The first one is to create verb frames for all verbs inflected by VS; however, this approach causes a vast amount of verb frames, which is a situation that we avoid, as we discussed above. Furthermore, VSs are not DS and do not derive new verbs. Classes and the meanings of the stems stay the same; the only change is on their argument–predicate relations. Thus, we believe that this approach does not provide a proper solution. A second and more appropriate solution is to represent VS-inflected verbs by the use of their stem frames as suggested in Şahin and Adalı (2018). However, instead of adding additional arguments to verb frames as in Şahin and Adalı (2018), we propose a more AMR-oriented approach also compatible with the English AMR framework. In the following paragraphs, we detail the proposed approach. We should state that since passive voice does not cause any changes on the verb argument structure as in English, we handle it by leaving the argument ARG0 empty as has been done for Spanish (Migueles-Abraira *et al.* 2018).

Reciprocal verbs express actions that are performed together or against each other. They are formed with the reciprocal suffix -*(I)ş* which could be affixed to only a few transitive and intransitive verb stems, for example, "öp**üş**mek" (*to kiss each other*), "özleşmek" (*to miss each other*), and "gül**üş**mek" (*to laugh together*). Şahin and Adalı (2018) benefit from the number of agents who do the action; however, this approach is not suitable for AMR because it is insufficient to represent the meaning of verbs in cases of mutual involvement of the agents to the action. We propose that the agents that perform the action reciprocally have to be both ARG0 and ARG1 of the verb. In Figure 7a, the subjects first linked via the conjunction *and* and then are used as the arguments.

Reflexive verbs are formed by combining the reflexive suffix -*(I)*n only with transitive verbs. Reflexive verbs are type of verbs that indicate actions that affect the person who performs the action either directly or indirectly, for example, "yıka**n**mak" (*wash oneself—to take a bath*),

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

"tara**n**mak" (*to comb one's hair*), and "giy**in**mek" (*to wear oneself—to get dress*). Şahin and Adalı (2018) suggest to define a new semantic role such as A0A1 which accounts for multi-role for representing such verbs, as a future work. The reason for this suggestion is stated as the PropBank conventions not allowing to annotate one argument with two different roles. However, this is possible in AMR. Thus, for Turkish AMR, we solve this issue by making ARG0 and ARG1 of the verbal stem as the same. We believe, our solution is more convenient since it increases the compatibility of the representation with the other AMR frameworks and the solution presented above for reciprocal verbs. Figure 7b shows the AMR representation of the reflexive verb *yıkan*.

Our solution using reentrancy for reciprocal and reflexive voices is similar to the solution proposed for the pronoun "se" in Spanish (Migueles-Abraira *et al.* 2018), except that Migueles-Abraira *et al.* (2018) add an extra concept in the case of reciprocal usage of this pronoun. In our solution, we intend not to distinguish reciprocal and reflexive representations since (1) in both cases the ones who do the action and the ones who are affected by the action are the same and (2) the original AMR conventions suggest that "AMR should abstract away from coreference gadgets like pronouns, zero-pronouns, reflexives, control structures, etc." (Banarescu *et al.* 2013). However, we also agree with Migueles-Abraira *et al.* (2018) that the use of some specific pronouns would help to differentiate the meaning. An alternative to our current solution might be to use some specific pronouns (e.g., "birbiri" (*each other*) for reciprocity and "kendi" (*oneself*) for reflexivity) in ARG1 of the predicates to differentiate the two phenomena. We believe the mentioned AMR convention may be reconsidered in the case of a universal schema covering MRLs.

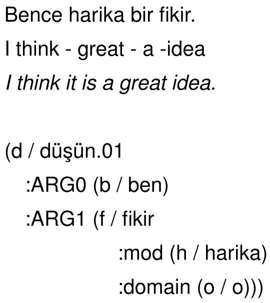
The causative suffixes (*-dIr*, *-t*, *-It*, *-Ir*, *-Ar*, *-Art*) attach to transitive or intransitive verbs (Göksel and Kerslake 2004) to construct causative structures such as "boya**t**mak" (to make somebody paint something), "yap**tır**mak" (*to make somebody do something*), and "kes**tir**mek" (*to make somebody cut something*). Şahin and Adalı (2018) introduce a new role ArgA to show the causer of an action. Although this approach seems a fairly neat solution to incorporate the verb framing of Turkish causative structures, we prefer not to use it with AMR compatibility concern in mind. In English, there is no need of an additional role to represent the causative structure since it is constructed by the predicate *make* whose arguments indicate the agents who do the action and who cause the action done. To make Turkish AMR parallel to English AMR, we prefer to create a new verb frame "yap.03" (an equivalent for *make-02* in the English PropBank) and use it in the AMR representation of Turkish causative verbs. Figure 7c illustrates the AMR representation of the causative verb "boyat" (*make somebody paint*). It is worth mentioning that all these voices may be used as nested structures and the meaning should be considered according to the context during the AMR annotation. The addition of two consecutive causative suffixes may or may not mean differently than the single occurrence of the causative voice. For example, for the sentence "Bizim mimara evi boyattırdım" (*I made our architect to make somebody to paint the house*), two nested *yap.03* predicates would be necessary in the AMR annotation.

3.4 Nominal derivation from nominals

The representation of DSs could be complicated. They may either correspond to some AMR relations and frames attached to the root word's concept or derive a new sense (i.e., AMR concept) that will replace the root word's concept in the AMR tree. These two scenarios may appear on the same suffix under different roles, and one should form appropriate AMR representations according to the sense within the current context.

DSs requiring the creation of a new AMR concept independent from the root word are the ones that generally add an exceptional meaning to the root word, which is not easily deducible from this root's meaning. The produced nominals appear dictionaries as separate lemmas. An example to this may be the word "gün**ey**" (*south*) which is derived from the root word "gün" (*day*). These newly derived words should appear as standalone concepts in AMR.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press



Bence harika bir fikir.
I think - great - a -idea
I think it is a great idea.

```
(d / düşün.01
   :ARG0 (b / ben)
   :ARG1 (f / fikir
               :mod (h / harika)
               :domain (o / o)))
```

Figure 8. AMR representation of the -CA suffix.

DSs, which may be expressed using AMR relations or frames attached to the root words' concepts, are the ones which generally have one or more predetermined literal meanings, and the derived nominal may be easily understood by relating this meaning of the suffix to the root word's meaning. It is possible that the derived words do not exist in the dictionary, such as "araba**sız**" (*without a car*). -*CA*, -*lI*, -*sIz* are the most common of such DSs having multiple meanings and multiple AMR representations. As an example, the suffix -*CA* that attaches to nominals results in many different meanings mostly depicted by :manner, :quant, and :duration AMR relations. However, when it attaches to pronouns, the word expresses a person's viewpoint and is considered as an independent event. Therefore, we use the predicate "düşün.01" (*to think*) for the representation of this case. Figure 8 provides an example annotation.

As stated above, the two presented scenarios may appear on the same suffix under different roles. For example, the suffix -*sIz* almost always denotes that the entity described lacks whatever is expressed by the root when added to nouns to form adjectives such as "sınır**sız**" (*unlimited*) or when added to nouns or pronouns to form adverbs denoting the non-involvement in an event of whatever expressed by the root such as "sen**siz**" (*without you*). However, although rarely, the same suffix may also add meanings outside the literal derivation meaning such as "ayna**sız**" ((slang) *police officer*) where the literal meaning would be *without mirror*. In the latter case, one should represent the word as a standalone concept parallel to the dictionary.

3.5 Pronoun dropping

A similar situation to the null-subject phenomenon appearing on verbs also appears on nominals with possessiveness. In Turkish, possessiveness is expressed through possessive suffixes attached to nominals and/or the possessor (another nominal in genitive case or possessive pronoun). The possessor may be easily dropped. However, one can still infer the dropped pronoun due to the possessive suffix attached to the possessed nominal. In AMR, we handle this situation similar to our solution to null subject by representing the dropped pronoun as an AMR concept. We then relate this concept to the possessed nominal with the ":poss" relation. As stated above, since Turkish does not have gender on third-person possessive pronouns, no ambiguity appears during this representation as opposed to Spanish (Migueles-Abraira *et al.* 2018) and Portuguese (Anchiêta and Pardo 2018a) pronoun representations. These later studies discuss ambiguities for representing third-person possessive pronouns but not within the context of pronoun dropping as in Turkish.

3.6 Reduplication

In Turkish, prefixation is used to a very limited extent. Some form of reduplication (i.e., emphatic reduplication accentuating the quality of an adjective) is an example of this and can be seen as another form of derivation. Since the meaning of the derived new word is directly deducible from

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

(a)

upuzun
long + emphatic reduplication marker
too long

```
(u / uzun
   :degree (ç / çok))
```

emphatic reduplication

(b)

Kitap mitap okuduk.
book - and stuff - we read
We read books and stuff.

```
(o / oku.01
   :ARG0 (b / biz)
   :ARG1 (a / and
            :op1 (k / kitap)
            :op2 (ş / şey
                     :ARG0-of (b2 / benze.01
                                  :ARG1 k))))
```

m-reduplication

Figure 9. AMR representation of emphatic reduplication and m-reduplication.

the meaning of the parent word, we again represent the derived word using its parent concept together with the relevant AMR relation (:degree). Figure 9 provides some reduplication samples. Another type[e] of reduplication is m-reduplication which involves the repetition of a word or phrase in a modified form, for example, “kitap mitap” (the word *book* followed by the second word which is just the same word with the changed initial letter). M-reduplication is a partial reduplication process that is used to widen the domain of the first word. We use the verb frame “benze.01” (*to seem like*) to depict the widening.

Li *et al.* (2019) also visit reduplication for Chinese AMR and mention two types of reduplications. However, they report that for the moment, they do not represent the one similar to our emphatic reduplication. For the second type adding extra meaning to the duplicated word (e.g., “every”), which is not available in Turkish, they add an abstract concept.

3.7 Copula

The Turkish copula is one of the more distinct features of Turkish grammar and has many forms such as zero-copula, *be* copula, past, evidential, and conditional copula. Parallel to English AMR, we mostly represent copula markers with the *:domain* relation in AMR. However, *:domain* does not fully cover the meaning of some nominals with copula markers and the conditional copula. To solve this problem, we use the reification approach (i.e., conversion of a role into a concept Banarescu *et al.* 2012) for the nominals which do not fit the *:domain* relation and for the conditional copula (the *:condition* relation). Choe *et al.* (2020) also mention this issue for Korean. In Figure 10, the noun “yaş” (*age*) takes the locative case suffix *-dA*, then it is inflected by the copula marker[f] and becomes the predicate of the sentence. The reification frame of *:age* relation which is “yaşlan.01” (*to age*) is used. Since the frame “yaşlan.01” does not have ARG2 in the Turkish PropBank, we propose to use an updated version of this frame which has the same argument structure as its English counterpart (i.e., age.01). Our solution to copula follows the one used for Korean in that they both use reification.

[e] In Turkish, there is also a third type of reduplication “doubling,” which is similar to English and examples of which are provided in the guideline.

[f] It also takes the first personal suffix *-Im*.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

On yaşında**y**ım.
ten - age + (-dA) locative case + (-y) be copula + I
I am ten years old.

```
(y / yaşlan.01
   :ARG1 (b / ben)
   :ARG2 (t / temporal-quantity)
            :unit (y / yıl)
            :quant 10))
```

Figure 10. AMR representation of a copula marker occurring after a locative marker.

4. Corpus construction

In line with the literature, we started to manually annotate the Turkish translation of the novel "Little Prince" from scratch[g] according to the Turkish AMR framework described above. Although English AMR representations of the same sentences helped the annotation process, AMR annotation from scratch is a quite time-consuming process that requires knowledge about the PropBank structure and in-depth analysis of the sentence meaning. The process may speed using semi-automatic annotation or adaptation of previous resources such as Treebanks and PropBanks. Turkish has such a resource "the Turkish PropBank"[h] (Şahin 2016) built upon the IMST Turkish Treebank (Sulubacak *et al.* 2016). As the second stage of corpus construction, we used this resource and a semi-automatic annotation approach to build the first Turkish AMR corpus more rapidly. For the semi-automatic annotation, we develop a rule-based parser that takes the PropBank sentences and automatically converts them into AMR graphs according to the framework introduced in Section 3. Human annotators work on these output graphs to build the final output instead of annotating from scratch. The following subsections introduce this rule-based tree-to-graph parser, its evaluation in terms of its impacts on the human annotation process and Smatch score between human annotations, and the first Turkish AMR corpus.

4.1 Rule-based tree-to-graph parser

The adopted idea in the development of our rule-based tree-to-graph parser is similar to the transition-based tree-to-graph parser introduced in Wang, Xue, and Pradhan (2015b), the input of which is the output of a dependency parser. Wang *et al.* (2015b) follow a supervised approach and align concepts and words (tokens) at first using JAMR (Flanigan *et al.* 2014), which is where our parsing approach diverges due to the following limitations and difficulties. First, we had very few AMR-annotated sentences during the parser development stage[i], and there was no previously developed aligner for Turkish. Thus, with these limited resources, developing an aligner from scratch was not an easy task due to the complex Turkish morphology. It is worth reminding that we aimed to develop an assistant tool to increase the number of annotated sentences faster. We believe that an unsupervised approach that maximizes the use of available resources (e.g., PropBank) and handcrafted lists is better suited to our problem.

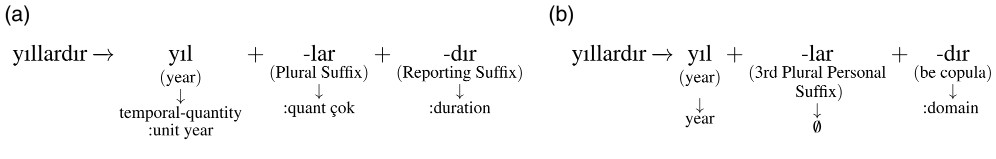
We design our parser as a rule-based one in which the ruleset includes the parsing rules and the mappings of sentence components to AMR concepts. The sentence components are the morphemes (e.g., -sız, -li, -ca) that need unique treatments and word spans that invoke abstract

[g] A preliminary investigation on these data was done in Azin and Eryiğit (2019) which reports the initial findings by focusing on the English AMR of the same 100 sentences and their alignments onto Turkish.

[h] The Turkish PropBank is available from http://tools.nlp.itu.edu.tr/Datasets.

[i] There were only 100 manually annotated sentences from the novel *Little Prince* while the parser was under construction.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press



Alignment of the word *yıllardır* that means "for years"

Alignment of the word *yıllardır* that means "these are years"

Figure 11. Alignment of the word *yıllardır*.

concepts. With this predefined mapping, we try to cover the compositional semantics defined at the morphological level in nominals. In line with the literature, we call this mapping alignment. We use the words "mapping" and "alignment" interchangeably hereafter. On the other hand, our parser uses semantic features together with syntactic ones in order to represent verb semantics at word and morphology levels. The Turkish PropBank provides the frames with their arguments where the most frequent verb frames are available. The remaining (x-rooted HPVs) need to be adapted to our representation (as we discussed in Section 3.1), and we try to handle them by expanding our ruleset with syntax-aware rules. For example, xlAn frames are represented with either ol.04 (to get) if x is a noun or ol.02 (to become) if x is an adjective. In our parser, we realize the AMR graph construction and the selection of the correct alignments between tokens and AMR concepts simultaneously.

The main reasons for this decision are that (i) a word in a sentence may be represented with complex AMR structures, and updating the tree/graph during parsing is easier rather than integrating such complex structures into the tree-to-graph transformation, (ii) several suffixes have multiple meanings, and dependency relations and morphological features provide helpful information about distinguishing their uses and functions. As an example, Figure 11 shows the alignments for the word "yıllardır," which may carry different meanings (i.e., "*for years*" or "*these are years*") according to its usage.

Since Turkish is an MRL, our parser highly relies on morphological features. As we discussed in Section 3, a suffix may form a concept or establish relationships between concepts. To detect the morphemes and their types, we use morphological analysis outputs and handle them according to Turkish AMR specifications. Our rule-based tree-to-graph parser takes its input in the CoNLL form[j] also used in the Turkish PropBank (Şahin and Adalı 2018), which added a semantic layer on top of the Turkish dependency treebank (IMST Sulubacak and Eryiğit 2018) sentences. Table 3 shows an example sentence in a shortened CoNLL format[k] where the first seven columns came from the dependency treebank and the last column was added during the PropBank annotations. This representation provides our parser (i) the dependency tree of a sentence (6th and 7th columns), as well as (ii) words' morphological analyses (4th and 5th columns), and (iii) PropBank frames of the verbs and their arguments (8th column). Although the information in the 8th column of the figure is given in a condensed form, in the original format they are given within multiple columns added to the end where every column after the ninth holds to indicate the arguments of a specific predicate, in the order that they appear within the sentence.

Our parsing rules determine transformation actions. First, we transform the CoNLL structure into a tree (called as "inter-step tree" from now on) by merging the dependency tree nodes and relations with the PropBank tags. Then, we transform the inter-step tree into an AMR graph by some actions determined with the ruleset. The parser, detailed in the following subsections, is developed as an open-source GitHub project[l] and shared with the researchers for further studies.

[j] Universal Proposition Banks https://github.com/System-T/UniversalPropositions.

[k] Some similar columns are removed due to space constraints, for example, minor POS tag.

[l] https://github.com/amr-turkish/turkish-amr-parser.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

Table 3. A sentence "Bu ilişkiyi bitirelim, böyle yürütemeyeceğim, dedi." (*Let's end this relationship, I can't run it like this, she said*) in the Turkish PropBank. The columns provide words' position within the sentence, surface form, lemma, parts-of-speech tags, morphological features, head word index, dependency relation, and the PropBank tag, respectively. The annotation "Y" indicates that the following tag is a verb frame

Ind.	Word	Lemma	PoS	Morphological features	Head	Dependency Rel.	Semantic layer				
1	Bu	bu	Det		2	DETERMINER					
2	ilişkiyi	ilişki	Noun	–	3	OBJECT	–	–	'A1'	'A0'	–
3	bitirelim	bitir	Verb	A3sg-Pnon-Acc	6	COORDINATION	'Y'	'bit.01'	–	–	–
4	,	,	Punc	Pos-Opt-A1pl	3	PUNCTUATION					
5	böyle	böyle	Adv	–	6	MODIFIER	–	–	–	'AM-MNR'	–
6	yürütemeyeceğim	yürü	Verb	Caus-Able-Neg-Fut-A1sg	8	OBJECT	'Y'	'yürü.01'	–	–	'A1'
7	,	,	Punc	–	6	PUNCTUATION					
8	dedi	de	Verb	–	0	PREDICATE	'Y'	'de.01'	–	–	–

4.1.1 Inter-step tree

The parser takes an input sample I=(V, A, *morph*, t, Prop), where

- $V = \{v_i \mid i \in [0,n], i \in \mathbb{N}\}$ is a set of nodes representing word tokens in the sentence,[m]
- $A = \{a_{ij} \mid i,j \in [0,n], i \neq j, i,j \in \mathbb{N}\}$ is a set of dependency relations between nodes v_j (the head) and v_i (the dependent),
- *morph* represents morphological features of words,
- t represents parts-of-speech tags of words,
- $Prop = \{prop_{ik} \mid i \in (0,n], i \in \mathbb{N}^+, k \in [0,m], k \in \mathbb{N}\}$ represents a set of semantic layer tags, where $prop_{ik}$ corresponds to k^{th} annotation of node v_i and m the number of semantic layer tags that the node v_i has.

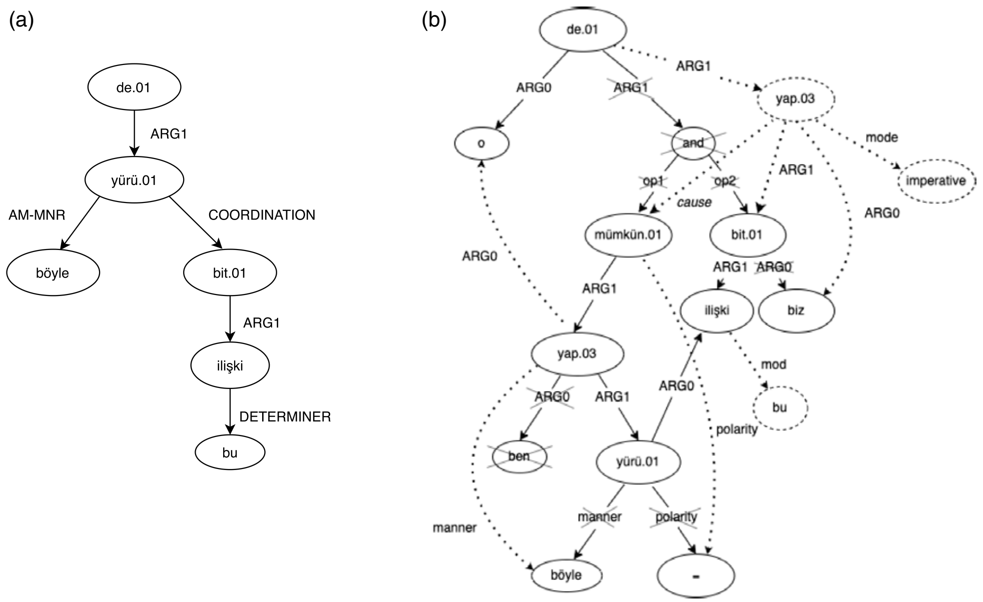
We define the inter-step tree $D = (C, R, NodeProperties)$, where $C = \{c_i \mid i \in (0,n], i \in \mathbb{N}^+\}$ represents a set of nodes, $R = \{r_{ji} \mid i,j \in (0,n], i \neq j, i,j \in \mathbb{N}^+\}$ represents a set of edges, and *NodeProperties* is a quadruple <*morph*, t, head node, dependency relation> consisting of the features of each node c_i. c_i and r_{ij} are defined as below, where orderof(j) represents the order of the predicate within the sentence. Since k=0 and k=1 are reserved for predicate declaration (see Table 3), the argument roles start from k=2.

$$c_i = \begin{cases} prop_{i1} & \text{if } prop_{i0} = \mathrm{Y} \\ v_i & \text{otherwise} \end{cases} \quad \text{and} \quad r_{ij} = \begin{cases} prop_{ik} & \text{if } k = \mathrm{orderof}(j) + 1 \\ a_{ij} & \text{otherwise} \end{cases}$$

Since the semantic layer tag $prop_{ik}$ can be a verb frame or an argument relation or the letter "Y," it can be expressed by a node or relation in the inter-step tree, depending on its type. When a node has more than one relation tag, the very first tag becomes c_i, the rest is used to establish reentrancy connections (details in Section 4.1.2). The dependency components directly participate in the construction of D if they do not have any semantic layer tags. Figure 12a shows the inter-step tree of the sentence given in Table 3. The inter-step tree is constructed by the semantic layer tags which are verb frames ("bit.01" (*end*), "yürü.01" (*walk*), "de.01" (*say*)), relations (AMR-MNR, ARG1), and the TreeBank nodes ("bu" (*this*), "ilişki" (*relationship*)) and dependency relations

[m] v_0 represents the root node in the dependency tree; it does not exist in the sentence.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press



Inter-step Tree

Automatically generated AMR Graph (The erroneously generated concepts and relations by the parser are depicted with a cross sign on the concept or the relation, and their correct counterparts are indicated with dashed lines)

Figure 12. Inter-step tree and AMR graph for "Bu ilişkiyi bitirelim, böyle yürütemeyeceğim, dedi." (*Let's end this relationship, I can't run it like this, she said*).

(DETERMINER, COORDINATION). The word "ilişki" (*relationship*) has two argument relation tags A0 (ARG0) and A1 (ARG1) (Table 3), and A1 is used in the inter-step tree since it is the first tag.

4.1.2 Parser

We use a similar notation to Wang *et al.* (2015b) for the introduction of our parser. However, our parser does the alignment between text spans and AMR concepts simultaneously, and it differs from the mentioned study having different actions in a rule-based setting rather than a transition-based one.

We define our rule-based tree-to-graph parser as
$Cr = (Cr, Actions, Cr_0, Rules)$.

- *Cr* is a set of parsing states,
- *Actions* is a set of actions A: Cr→Cr,
- Cr_0 is an initialization step where inter-step tree is built,
- *Rules* is a set of conversion rules.

A parsing state is a couple (D, q), where q holds node indices according to the sentence word order, and it is used as a queue to process all nodes of the inter-step tree. The graph conversion starts with the construction of the inter-step tree and then continues with processing q. The parser starts with the first element of q and iterates by giving its related node in D and its properties to Rules where the next action is determined. At each iteration, the Rules set returns a set of actions

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

according to the given node properties ($[Rule(c_i, NodeProperties_i) \rightarrow Actions_a]$), and the parser applies the action on D.

We have eight types of actions (Table 4) that will cover all possible situations in the conversion process. $Pr(i)$ returns the parent index of a node at index i, $Ch(i)$ returns all the children indexes of a node at index i, $\gamma{:}C \rightarrow R$ is a function that establishes an AMR relation between two input concepts, where the second argument of the function becomes the parent node after the action, $\zeta{:}C \rightarrow R$ deletes the relations between the current node and its parent. The function takes two arguments (i.e., the current node and its parent in focus). Since the initial inter-step tree is constructed from the dependency tree, the dependent could only have one head at the beginning but could have multiple heads as the AMR graph gets constructed. The focused parent may not be found directly and should be given as an argument to this function. $\delta{:}C \rightarrow C$ is a function that creates a new concept node from an existing node due to its morphological features and creates a relation between the new node and its parent (the current node). $\varphi{:}C \rightarrow C$ deletes a node given as an argument. $\iota{:}R \rightarrow L$, where L is the AMR relation set, assigns a label to the given edge as an argument. The eight actions are as follows:

- Add Edge: It simply adds an edge between the node in the queue with index i (c_{q_i}) and the other node with index j (c_j) in the inter-step tree. The newly created edge $r_{c_{q_i}c_j}$ is included into the edge set R. It also assigns a label l from AMR label set L to $r_{c_{q_i}c_j}$.
- Delete Edge: It deletes the edge between the node in the queue with index i (c_{q_i}) and its parent. The removed edge $r_{c_{q_i}c_{Pr(q_i)}}$ is excluded from R.
- Add Node: It creates a new node c_k based on the node in the queue with index i (c_{q_i}) and establishes an edge between c_k and c_{q_i} where the parent node is c_{q_i}. The newly created node c_k is included into the node set C.
- Replace Head: It replaces the node in the queue c_{q_i} with a new one c_k. It first takes all children nodes of the node c_{q_i} and then creates edges between the children and c_k. The newly created node c_k is included into the node set C and c_{q_i} is excluded from C.
- ReAttach: It deletes the edge between a node in the queue (c_{q_i}) and its parent. A new edge is established between c_{q_i} and a node c_k.
- Swap: It deletes the edge between a node in the queue (c_{q_i}) and its parent. It creates a new edge between these two nodes in the opposite direction.
- Merge: It creates a new node c_k by combining the node in the queue with index i (c_{q_i}) and its parent and connects c_k to the grandparent of c_{q_i}. The nodes c_{q_i} and $c_{Pr(q_i)}$ are removed from the node set C.

The parser processes q twice consecutively. The first process normalizes D either by a node addition or deletion and converts D to the graph form by adding reentrancies. The second process gives the graph its final shape by mapping nodes and relations with their AMR counterparts. We name these two steps graph conversion and post-process.

The graph conversion consists of three sub-steps: node removal, reentrancy, and suffix alignment. Nodes to get removed are for the words that do not contribute to the sentence meaning. These are determiners or intensifiers of the other nodes. The parser removes nodes connected to their heads with the relations DETERMINER and INTENSIFIER in the inter-step tree. However, this does not mean that all intensifiers and determiners do not contribute to the sentence meaning. Their meaning contributions depend on their usage and the whole sentence meaning. Our parser is not capable of distinguishing which ones should be removed or not. Reentrancies emerge when the same node participates in multiple relations. We call such nodes reentrancy nodes. The reentrancy nodes are the ones having more than one argument tags in their semantic layer (Table 3

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

Table 4. Actions

Action name	Action performed	Assigned labels
Add Edge	$\gamma[(c_{q_i}, c_j) \rightarrow \{r_{c_{q_i}c_j}\} \cup R], c_j \in C$	$\iota[(r_{c_{q_i}c_j}) \rightarrow l], l \cup L$
Delete Edge	$\zeta[(c_{q_i}, c_{Pr(q_i)}) \rightarrow R \setminus r_{c_{q_i}c_{Pr(q_i)}}]$	None
Add Node	$\delta[(c_{q_i}) \rightarrow \{c_k\} \cup C, \gamma(c_k, c_{q_i})]$	None
Delete Node	$\varphi[(c_i) \rightarrow C \setminus c_i]$	None
Replace Head	$s = Ch(Pr(q_i)), c_k \cup C, \gamma(c_j, c_k), j \in s, \gamma(c_k, c_{Pr(Pr(q_i))}), \varphi(c_{Pr(q_i)})$	$\iota[(r_{c_jc_k}) \rightarrow l], l \cup L$
ReAttach	$\zeta(c_{q_i}, c_{Pr(q_i)}), \gamma(c_{q_i}, c_k), c_k \in C$	$\iota[(r_{c_{q_i}c_k}) \rightarrow l], l \cup L$
Swap	$\zeta(c_{q_i}, c_{Pr(q_i)}), \gamma(c_{Pr(q_i)}, c_{q_i})$	$\iota[(r_{c_{Pr(q_i)}c_{q_i}}) \rightarrow l], l \cup L$
Merge	$c_k = c_{q_i} \cup c_{Pr(q_i)}, \gamma(c_k, c_{Pr(Pr(q_i))}), \varphi(c_{q_i}), \varphi(c_{Pr(q_i)})$	$\iota[(r_{c_kc_{Pr(Pr(q_i))}}) \rightarrow l], l \cup L$

node at the 2nd indice). As we mentioned in the previous subsection, the first tag is embedded in the inter-step tree. For the rest, in this step, the parser establishes new relations between reentrancy nodes and the most suitable nodes selected by the ruleset. As a result of this process, the inter-step tree turns into a graph. In Figure 12b, it is shown that the previously absent relation ARG0 (A0) is added into D between "ilişki" (*relationship*) and "yürü.01" (*work*).

Converting morphological suffixes to proper AMR components is the most important step of the Turkish AMR parsing. As discussed before, the majority of the meaning contributions come from these suffixes. The parser uses the given morphological properties of nodes ($Node-Properties$). The following operations may be performed in accordance with the Turkish AMR framework (Section 3):

- adding a null subject,
- adding modalities,
- adding polarity,
- adding new nodes and relations coming from voice structures,
- adding relations coming from case markers.

Figure 12b gives the automatically generated AMR graph for the studied example. As may be seen from the figure, the previously absent nodes "biz" (*we*) and "o" (*s/he*) are revealed by personal suffix markers extracted from *NodeProperties* and become the agents of the predicates "bitire**lim**" and "dedi." The word "yürü**temey**eceğ**im**" (the verb *run*[n] in future tense with modality and negativity markers) has one causative suffix and multiple ISs (i.e., modality, negativity, and personal markers). The parser adds the concepts "yap.03" (*make*), "mümkün.01" (*possible*), "-" (*minus*), and "ben" (*I*) to represent causativity, modality, negativity, and the agent who does the action, respectively. It should be noted that the word "bitir" (the verb *end*) is constructed from the root word "bit" (*to end*) by the causative suffix *-ir*. However, since the morphological analyzer outputs its lemma as "bitir" instead of "bit" and misses to output the causative structure (Table 3 node at the 3rd indice), our AMR parser fails to extract this information from the node properties and to add the "yap.03" (*make*) concept in this example to represent causativity.

Post-processing maps non-AMR components that the previous stage has not mapped to AMR concepts and relations. The nodes that have abstract concepts in their representations are aligned

[n] The verb "yürüt" is constructed from the verb "yürü" (*walk*) by the causative suffix *-t* and gains the meaning of "make it work." However, its meaning in this sentence can be translated as "*run*" ("run a relationship").

Table 5. Direct mapping of PropBank relations to AMR relations

PropBank	AMR	PropBank	AMR
AM-MNR	:manner	AM-LOC	:location
AM-ADV	:mod	AM-TMP	:time
AM-INS	:instrument	AM-COM	:accompanier

with their AMR representations. On the other hand, the relations mapping could be either edge renaming or transformation of an edge to an equivalent AMR sub-graph. If the AMR specification has a relation that has the same meaning, edge renaming is straightforward as shown in Table 5. In Figure 12b, the parser maps AMR-MNR to the relation :manner and transforms COORDINATION to a sub-graph adding the node "and." One should note that our parser is mostly developed on top of the syntactic features of words and sentences and is not good at capturing semantic relationships between sentence constituents. In Figure 12b, we see that the parser fails to construct the *:cause* relation since it could not get any clue about this semantic relation from the node properties.

4.1.3 Evaluation

We evaluate the effectiveness of the parser (1) by comparing its outputs to gold standards and (2) using it for semi-automatic annotation. For the first set of evaluations, we use the Smatch score (Cai and Knight 2013), an AMR evaluation metric that calculates the degree of overlapping between two formal semantic structures.[o] In the AMR case, two AMR graphs to be compared with each other are rewritten as logical propositions (i.e., triples), and the f-score between these triples in the graphs in terms of the propositional overlap against each other is calculated. For example, the triplet < ARG0(a, b)> shows that the two variables a and b are related in the AMR graph with the relationship ARG0. The produced variable names for the same concept in the two graphs may be different from each other, and Cai and Knight (2013) solve this problem by getting all possible triples and finding a subset that gives maximum f-score with the help of integer linear programming.

Our parser achieved a Smatch Score of 0.65 and 0.60 (on the Turkish AMR corpus Section 4.2) at the end of the first and second MAMA cycle iterations. One should note that similar to many parsing tasks in NLP, this is not an end-to-end parser and designed to be used with gold-standard dependency and PropBank annotations. In a real-world setting, our rule-based tree-to-graph AMR parser performance will be affected by the errors introduced by automatic morphological analysis, dependency parsing, and semantic role labeling. Still, we believe that this first Turkish AMR parser will act as a strong baseline for future studies on Turkish AMR parsing. As will be detailed in Section 4.2, our corpus contains 600 sentences with gold-standard dependency and PropBank annotations where the parser's performance is measured as 0.61 and 100 sentences with automatically produced dependency and PropBank annotations where the parser's performance is measured as 0.54 with an overall average of 0.60 Smatch score as given above. One should note that the automatic dependency parsing (Sulubacak and Eryiğit 2018) and semantic role labeling (Şahin and Steedman 2018) performances in Turkish are still not on par with English due to the low training data resources.

For the second set of evaluations, we create two experimental setups to measure the effects of the parser in the annotation process. First, we select two sets of 10 sentences from IMST with similar syntactic and semantic structures. The selected sentences are also similar in terms of sentence length and structural complexity. We then record the time spent by a single human

[o] Smatch tool is available from https://amr.isi.edu/evaluation.html.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

Table 6. Annotation times

Annotation style	Total time (in sec)
Manual annotation	1438
Semi-automatic annotation	**545**

annotator who annotates these two sets separately; for one of the sets, the annotation is realized from scratch, and for the other one, it is done via semi-automatic annotation, where the experienced human annotator corrects the outputs of the introduced parser. The elapsed times in both annotation processes are given in Table 6, which reveals a remarkable reduction in annotation times (of around two-thirds) when the parser is used as a pre-processor, and the human annotator corrects its outputs rather than annotating from scratch (manual annotation). One should note that the selected sentences were not very difficult and the time spent for the annotation of a single sentence may not be generalized.

In the second experiment, we randomly select 25 additional sentences not annotated before from IMST. Two human annotators annotate these sentences, one working from scratch and one working on the outputs of the tree-to-graph parser. The inter-annotator agreement between the two human annotators is measured as 0.85 Smatch score. We also make an error analysis on the sentences where there is no agreement between our annotators and observe that the annotations produced by the human annotator working on the parser's outputs better conform to the predicate frame names than the ones produced by the human annotator working from scratch. This is an expected outcome since our parser uses gold-standard predicate frame tags, which should be replaced with an automatic predicate disambiguator in a real scenario, while the human annotator working from scratch try to select them each time manually, which is error prone. On the other hand, we see that the parser directs the annotator to use more conjunctions (as exemplified in the previous section (Figure 12b) the use of "and" instead of the *:cause* relation) and possessiveness (instead of *:topic*, *:part-of*, etc.) in the complex sentences than needed. We observe that the human annotator corrected these most of the time (as may be observed from the Smatch scores between the parser and the human annotator above), but in some sentences with complex semantic structures, these could be missed. The parser also helps extensively to the human annotator in cases morphologically inferable (such as null subject, dropped pronouns, modality), which could be missed by the human annotator working from scratch.

4.2 Turkish AMR corpus

Linguistic annotations are not as straightforward as one might think. Generally, the specifications are needed to be updated frequently during data annotation. Bunt (2015) gives the details of this process and name it the MAMA cycle (model-annotate-model-annotate). In our annotations, we experienced a similar cycle. We had two iterations to achieve the final framework and the corpus.

The data set was annotated by two native-speaking annotators. In the first iteration, we worked with a foreign linguist who was experienced in AMR through her previous work in different AMR projects for other languages and was familiar with Turkish. The linguist collaborated with the team during the preliminary investigations of Turkish-specific structures and a warm-up annotation period which will be detailed below (Azin and Eryiğit 2019). As the annotation environment, we have used an updated version of Hermjakob (2013) to cover non-English characters in Turkish, which were processable with the original tool.

The novel "The Little Prince" by Antoine de Saint-Exupéry published in 1943 was used in many AMR corpus studies for different languages (Banarescu *et al.* 2013; Li *et al.* 2016; Anchiêta and Pardo 2018a), which provides an opportunity to compare AMR representations on the same text between different languages. The first iteration of our MAMA process started with a warm-up annotation period where we used the first 100 sentences of the same novel to make a preliminary investigation of Turkish AMR structures, which could be defined in parallel with English or not.[p] As a result of this warm-up annotation period, we used our findings to build the first draft of the Turkish AMR specifications (named as specs hereinafter) and the backbone of our tree-to-graph parser. Due to our limited human annotation resource, the semi-automated annotation approach introduced in Section 4.1 was used to speed up the annotations after the warm-up period. With this purpose, the annotation was continued on the IMST (Sulubacak *et al.* 2016; Sulubacak and Eryiğit 2018; Şahin and Adalı 2018) Turkish Treebank sentences (instead of "the Little Prince") which provide gold-standard linguistic annotations in lower levels (i.e., morphology, dependency, PropBank annotations) used by the parser. During the annotations, the specs were continued to be updated with a data-driven approach by making use of (i) the sections of Turkish grammar books about the grammatical phenomena appearing in the data in focus and (ii) the English AMR guideline. At the end of this first iteration, the first version of the specs and the Turkish AMR corpus containing 700 sentences (100 sentences from Little Prince, 600 sentences from IMST) were built.

In the second iteration, a knowledge-driven approach has been adopted aiming to build the formal specs. In this iteration, we tried to cover all the Turkish-specific phenomena regardless they appear in the data in focus or not and to introduce generalizable solutions to these, which yield considerable updates in the specs and the need for the re-annotation of the data set. As detailed in the previous sections, the Turkish grammar books, the Turkish dictionary, and the previous semantic annotation efforts have been investigated during these analyses. Additionally, the AMR studies in other languages (e.g., Korean) were examined to develop a framework consistent with the literature. As a result of this iteration, the Turkish AMR annotation framework introduced in Section 3 has been developed, and the re-annotation was accomplished in compliance with it. This iteration also revealed the collection or generation of many samples outside of the corpus to be included in the Turkish AMR guideline.

IMST contains texts gathered from eight genres (Buchholz and Marsi 2006) (e.g., news, novels, interviews, etc.). The average sentence length of the 600 IMST sentences in our corpus is 11 tokens where 16% of them (i.e., 99 sentences[q]) consists of less than 5 words. One should note that the sentence length is not a reliable metric to make a conclusion about the sentence complexity since a short Turkish sentence may be very complex in terms of AMR (e.g., "Aradığımı buldum sandım" (*I thought I found the thing that I was looking for.*) On the other hand, the complex sentences, which contain at least one subordinate clause in addition to the main clause (Göksel and Kerslake 2004), are common in IMST. 60% of the sentences (357 sentences) have a complex structure.[r]

In order to measure the inter-annotator agreement between our human annotators, we randomly selected 100 hundred sentences from IMST at the end of the second iteration of the MAMA cycle, and a second annotator re-annotated them in terms of AMR, the linguistic phenomena they possess, and their place (graph fragment) within the AMR graph for further evaluation (detailed below). Table 7 presents the results of the inter-annotator agreements on different subsets of these 100 sentences based on the linguistic phenomenon. We calculated two different Smatch scores: one on the entire sentence's AMR graph as usual and the other on the AMR graph fragment concerning only the mentioned linguistic phenomenon. In the table, we provide these two scores under the columns named "full sentence" and "phenomenon fragment." Since personal markers are obligatory, we excluded this phenomenon from the evaluations. Ninety of the sentences were tagged as comprising one or more of the phenomena investigated in Section 3. Since a single sentence may comprise more than one phenomena, the total number of sentences within the subsets

[p] The first 100 sentences were annotated by the linguist and one of the annotators simultaneously, and the inter-annotator agreement between them was measured as 92% in terms of Smatch score.

[q] The Smatch score of our parser on short sentences is 0.75.

[r] The parser achieves a Smatch score of 0.58 on complex sentences.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

Table 7. Annotation agreements based on linguistic phenomena

		Smatch score	
	# of sentences	Full sentence	Phenomenon fragment
Randomly selected sentences	100	0.89	–
Sentences comprising linguistic phenomena visited in this article	90	0.89	–
Verbal derivation from nominals	3	0.75	0.77
Noun that invoke predicates	49	0.89	0.91
Non-finite adverbial subordination	11	0.93	0.86
Headless relative constructions	8	0.96	1
Null subject	46	0.90	0.98
Modality	12	0.85	0.90
Voices	9	0.91	0.86
Nominal derivation from nominals	10	0.84	0.87
Pronoun dropping	14	0.92	0.97
Reduplication	1	0.68	0
Copula	12	0.86	0.92

(based on separate phenomena) in the second half of the table is greater than 90. When we investigate the inter-annotator agreements, we see that our annotators systematically agreed on most of the AMR annotations of the mentioned linguistic phenomena (e.g., pronoun dropping, modality, null subject) with a Smatch score greater than 80%. Two phenomena obtained scores lower than 80%. These are "reduplication" and "Verbal Derivation from Nominals" which were mistakenly annotated by one of our annotators. However, the sample size (1 and 3 sentences) is too small to deduct any conclusions.

As stated in the previous sections, in some situations, we needed to update the Turkish PropBank, either creating new predicate frames or adding new arguments to existing ones. While we created seven verb frames for idiomatic expressions, the rest were for the verbs whose frames were missing and the representation issue of possibility (i.e., "mümkün.01) as stated in Section 3.3.2. These yielded to the addition of 14 predicate frames and the update of 2 predicate frames in total. We believe this shows that our proposed solution is reasonable and does not yield a high number of new predicate frame generations.

5. Conclusion

MRLs pose particular problems to syntactic and semantic representation frameworks that stand as a challenge to establishing universal frameworks. Turkish is a prominent example of MRLs, and its agglutinative morphology yields the need for reconsideration of the AMR framework originally developed for English. For the first time in the literature, this article introduced a Turkish AMR

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

representation framework, which we believe will shed light on further studies for similar languages and will help create multilingual frameworks. The article discussed Turkish constructions which needed special treatment for AMR representations and introduced a rule-based AMR parser to speed up the manual annotation process and the very first AMR corpus for Turkish.

Designed as a result of both data- and knowledge-driven modeling, the framework mainly reveals the mechanisms to deal with the highly productive derivational and inflectional morphology of Turkish. The research shows that the rich derivational morphology of the language in focus cannot be used directly in AMR as represented in existing knowledge bases or dictionaries, and AMR-oriented definitions need to be made. As expected, the rich inflectional morphology reveals the synthesis of multiple concepts of an AMR graph from a single word. The use of the introduced rule-based tree-to-graph AMR parser has been shown to accelerate the annotation speed. We believe the introduced resources will speed up the construction of larger AMR corpora for Turkish and the development of more successful data-driven end-to-end parsers consequently.

We should also point out that AMR is not the only option for the semantic representation of Turkish; it is possible to apply alternative representations in the coming years. We believe that our study, which is the first attempt to reveal the fundamental challenges in the formal meaning representation of Turkish, will also shed light on these future studies.

Acknowledgments. The authors would like to offer special thanks to Zahra Azin for helpful discussions during the preliminary stage of this work and her assistance during the data annotation stage.

References

Abend O. and Rappoport A. (2013). Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria. Association for Computational Linguistics, pp. 228–238.

Adomako K. (2012). Verbal nominalization as a derivational process: The case of Akan. *Ghana Journal of Linguistics* **1**(2), 43–64.

Anchiêta R. and Pardo T. (2018a). Towards AMR-BR: A SemBank for Brazilian Portuguese language. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, Miyazaki, Japan.

Anchiêta R. and Pardo T. (2020). Semantically inspired AMR alignment for the Portuguese language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, pp. 1595–1600.

Anchiêta R.T. and Pardo T.A.S. (2018b). A rule-based AMR parser for Portuguese. In Simari G.R., Fermé E., Gutiérrez Segura F. and Rodríguez Melquiades J.A. (eds), *Advances in Artificial Intelligence - IBERAMIA 2018*, Cham. Springer International Publishing, pp. 341–353.

Astudillo R.F., Ballesteros M., Naseem T., Blodgett A. and Florian R. (2020). Transition-based parsing with stack-transformers. arXiv preprint arXiv:2010.10669.

Azin Z. and Eryiğit G. (2019). Towards Turkish abstract meaning representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Florence, Italy. Association for Computational Linguistics, pp. 43–47.

Babko-Malaya O. (2005). Guidelines for propbank framers. Unpublished manual, September.

Bai X., Song L. and Zhang Y. (2020). Online back-parsing for AMR-to-Text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, pp. 1206–1219.

Ballesteros M. and Al-Onaizan Y. (2017). AMR parsing using stack-lstms. arXiv preprint arXiv:1707.07755.

Banarescu L., Bonial C., Cai S., Georgescu M., Griffitt K., Hermjakob U., Knight K., Koehn P., Palmer M. and Schneider, N. (2012). Abstract Meaning Representation (AMR) 1.0 specification. In *Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL*, pp. 1533–1544.

Banarescu L., Bonial C., Cai S., Georgescu M., Griffitt K., Hermjakob U., Knight K., Koehn P., Palmer M. and Schneider, N. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, Sofia, Bulgaria. Association for Computational Linguistics, pp. 178–186.

Barzdins G. and Gosko D. (2016). Riga at semeval-2016 task 8: Impact of smatch extensions and character-level neural translation on AMR parsing accuracy. arXiv preprint arXiv:1604.01278.

Basile V., Bos J., Evang K. and Venhuizen N. (2012). Developing a large semantically annotated corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

Blloshmi R., Tripodi R. and Navigli R. (2020). XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, pp. 2487–2500.

Bonial C., Donatelli L., Abrams M., Lukin S.M., Tratz S., Marge M., Artstein R., Traum D. and Voss C. (2020). Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, pp. 684–695.

Bonn J., Palmer M., Cai Z. and Wright-Bettner K. (2020). Spatial AMR: Expanded spatial annotation in the context of a grounded Minecraft corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, pp. 4883–4892.

Bos J. (2016). Squib: Expressive power of Abstract Meaning Representations. *Computational Linguistics* **42**, 527–535.

Buchholz S. and Marsi E. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, New York City. Association for Computational Linguistics, pp. 149–164.

Bunt H. (2015). On the principles of semantic annotation. In *Proceedings of the 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11)*.

Cai D. and Lam W. (2020). AMR parsing via graph-sequence iterative inference. arXiv preprint arXiv:2004.05572.

Cai S. and Knight K. (2013). Smatch: An evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria. Association for Computational Linguistics, pp. 748–752.

Choe H., Han J., Park H. and Kim H. (2019a). Copula and case-stacking annotations for Korean AMR. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pp. 128–135.

Choe H., Han J., Park H., Oh T., Park S. and Kim H. (2019b). Korean Abstract Meaning Representation (AMR) guidelines for graph-structured representations of sentence meaning. In *Proceedings of the 31th Annual Conference on Human and Cognitive Language Technology*, pp. 252–257.

Choe H., Han J., Park H., Oh T.H. and Kim H. (2020). Building Korean Abstract Meaning Representation corpus. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, Barcelona Spain (online). Association for Computational Linguistics, pp. 21–29.

Choi J.D., Bonial C. and Palmer M. (2010). Propbank frameset annotation guidelines using a dedicated editor, Cornerstone. In *LREC*.

Damonte M. and Cohen S.B. (2018). Cross-lingual Abstract Meaning Representation parsing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana. Association for Computational Linguistics, pp. 1146–1155.

Damonte M. and Cohen S.B. (2019). Structural neural encoders for AMR-to-text generation. arXiv preprint arXiv:1903.11410.

Damonte M., Cohen S.B. and Satta G. (2017). An incremental parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Valencia, Spain. Association for Computational Linguistics, pp. 536–546.

Dohare S., Karnick H. and Gupta V. (2017). Text summarization using Abstract Meaning Representation. arXiv preprint arXiv:1706.01678.

Eryiğit G. (2014). ITU Turkish NLP web service. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden. Association for Computational Linguistics, pp. 1–4.

Eryiğit G., Nivre J. and Oflazer K. (2008). Dependency parsing of Turkish. *Computational Linguistics* **34**, 357–389.

Fan A. and Gardent C. (2020). Multilingual AMR-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, pp. 2889–2901.

Feng L. (2021). *WISeN: Widely Interpretable Semantic Network for Richer Meaning Representation*. PhD Thesis, Emory University, Atlanta, GA. Undergraduate Honors Thesis, Emory University, Atlanta, GA, 2021.

Flanigan J., Dyer C., Smith N.A. and Carbonell J. (2016). CMU at SemEval-2016 task 8: Graph-based AMR parsing with infinite ramp loss. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California. Association for Computational Linguistics, pp. 1202–1206.

Flanigan J., Thomson S., Carbonell J., Dyer C. and Smith N.A. (2014). A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland. Association for Computational Linguistics, pp. 1426–1436.

Foland W. and Martin J.H. (2017). Abstract Meaning Representation parsing using LSTM recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. Association for Computational Linguistics, pp. 463–472.

Göksel A. and Kerslake C. (2004). *Turkish: A Comprehensive Grammar*. Routledge.

Goodman J., Vlachos A. and Naradowsky J. (2016). Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany. Association for Computational Linguistics, pp. 1–11.

Guo Z. and Lu W. (2018). Better transition-based AMR parsing with a refined search space. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics, pp. 1712–1722.

Hermjakob U. (2013). AMR editor: A tool to build Abstract Meaning Representations. Technical report, ISI.

Holmberg A., Nayudu A. and Sheehan M. (2009). Three partial null-subject languages: A comparison of Brazilian Portuguese, Finnish and Marathi. *Studia linguistica* **63**(1), 59–97.

Huang L., Cassidy T., Feng X., Ji H., Voss C.R., Han J. and Sil A. (2016). Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany. Association for Computational Linguistics, pp. 258–268.

Jin L. and Gildea D. (2020). Generalized shortest-paths encoders for AMR-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online). International Committee on Computational Linguistics, pp. 2004–2013.

Kasper R.T. (1989). A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, 21–23 February 1989*.

Koller A., Oepen S. and Sun W. (2019). Graph-based meaning representations: Design and processing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, Florence, Italy. Association for Computational Linguistics, pp. 6–11.

Konstas I., Iyer S., Yatskar M., Choi Y. and Zettlemoyer L. (2017). Neural AMR: Sequence-to-sequence models for parsing and generation. arXiv preprint arXiv:1704.08381.

Li B., Wen Y., Qu W., Bu L. and Xue N. (2016). Annotating the little prince with Chinese AMRs. In *Proceedings of the 10th Linguistic Annotation Workshop held in Conjunction with ACL 2016 (LAW-X 2016)*, Berlin, Germany. Association for Computational Linguistics, pp. 7–15.

Li B., Wen Y., Song L., Qu W. and Xue N. (2019). Building a Chinese AMR bank with concept and relation alignments. In *Linguistic Issues in Language Technology, Volume 18, 2019 - Exploiting Parsed Corpora: Applications in Research, Pedagogy, and Processing*. CSLI Publications.

Li M., Zareian A., Zeng Q., Whitehead S., Lu D., Ji H. and Chang S.-F. (2020). Cross-media structured common space for multimedia event extraction. arXiv preprint arXiv:2005.02472.

Liao K., Lebanoff L. and Liu F. (2018). Abstract meaning representation for multi-document summarization. arXiv preprint arXiv:1806.05655.

Linh H. and Nguyen H. (2019). A case study on meaning representation for Vietnamese. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pp. 148–153.

Liu F., Flanigan J., Thomson S., Sadeh N. and Smith N.A. (2018a). Toward abstractive summarization using semantic representations. arXiv preprint arXiv:1805.10399.

Liu Y., Che W., Zheng B., Qin B. and Liu T. (2018b). An AMR aligner tuned by transition-based parser. arXiv preprint arXiv:1810.03541.

Lyu C. and Titov I. (2018). AMR parsing as graph prediction with latent alignment. arXiv preprint arXiv:1805.05286.

Mager M., Astudillo R.F., Naseem T., Sultan M.A., Lee Y.-S., Florian R. and Roukos S. (2020). Gpt-too: A language-model-first approach for AMR-to-text generation. arXiv preprint arXiv:2005.09123.

May J. (2016). SemEval-2016 task 8: Meaning representation parsing. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California. Association for Computational Linguistics, pp. 1063–1073.

May J. and Priyadarshi J. (2017). SemEval-2017 task 9: Abstract Meaning Representation parsing and generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada. Association for Computational Linguistics, pp. 536–545.

Meyers A., Reeves R., Macleod C., Szekely R., Zielinska V., Young B. and Grishman R. (2004). The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, Boston, Massachusetts, USA. Association for Computational Linguistics, pp. 24–31.

Migueles-Abraira N., Agerri R. and Diaz de Ilarraza A. (2018). Annotating Abstract Meaning Representations for Spanish. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Naseem T., Shah A., Wan H., Florian R., Roukos S. and Ballesteros M. (2019). Rewarding smatch: Transition-based AMR parsing with reinforcement learning. arXiv preprint arXiv:1905.13370.

Nivre J., Agić Ž., Ahrenberg L., Antonsen L., Aranzabe M.J. et al. (2017). Universal Dependencies 2.1. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University - Corpus - Project code: 15-10472S; Project name: Morphologically and Syntactically Annotated Corpora of Many Languages. Konya, Turkey.

Oepen S., Abend O., Hajic J., Hershcovich D., Kuhlmann M., O'Gorman T., Xue N., Chun J., Straka M. and Uresova, Z. (2019). MRP 2019: Cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, Hong Kong. Association for Computational Linguistics, pp. 1–27.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

Palmer M., Gildea D. and Kingsbury P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational linguistics* **31**, 71–106.

Peng X., Gildea D. and Satta G. (2018). AMR parsing with cache transition systems. In *AAAI*, pp. 4897–4904.

Şahin G.G. (2016). Framing of verbs for Turkish propbank. *The First International Conference on Turkic Computational Linguistics at CICLING 2016*, Konya, Turkey.

Şahin G.G. and Adalı E. (2018). Annotation of semantic roles for the Turkish Proposition Bank. *Language Resources and Evaluation* **52**, 673–706.

Şahin G.G. and Steedman M. (2018). Character-level models versus morphology in semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 386–396.

Sobrevilla Cabezudo M.A. and Pardo T. (2019). Towards a general Abstract Meaning Representation corpus for Brazilian Portuguese. In *Proceedings of the 13th Linguistic Annotation Workshop*, Florence, Italy. Association for Computational Linguistics, pp. 236–244.

Song L., Gildea D., Zhang Y., Wang Z. and Su J. (2019). Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics* **7**, 19–31.

Song L., Zhang Y., Peng X., Wang Z. and Gildea D. (2016). AMR-to-text generation as a traveling salesman problem. arXiv preprint arXiv:1609.07451.

Song L., Zhang Y., Wang Z. and Gildea D. (2018). A graph-to-sequence model for AMR-to-text generation. arXiv preprint arXiv:1805.02473.

Sulubacak U. and Eryiğit G. (2018). Implementing universal dependency, morphology, and multiword expression annotation standards for Turkish language processing. *Turkish Journal of Electrical Engineering & Computer Sciences* **26**, 1662–1672.

Sulubacak U., Eryiğit G. and Pamay T. (2016). IMST: A revisited Turkish Dependency Treebank. In Karaoğlan B., Kşla T. and Kumova S. (eds), *Proceedings of TurCLing 2016, the 1st International Conference on Turkic Computational Linguistics*, Turkey. EGE University Press, pp. 1–6.

Takase S., Suzuki J., Okazaki N., Hirao T. and Nagata M. (2016). Neural headline generation on Abstract Meaning Representation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas. Association for Computational Linguistics, pp. 1054–1059.

Van Noord R. and Bos J. (2017). Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. arXiv preprint arXiv:1705.09980.

Wang C. and Xue N. (2017). Getting the most out of AMR parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics, pp. 1257–1268.

Wang C., Xue N. and Pradhan S. (2015a). Boosting transition-based AMR parsing with refined actions and auxiliary analyzers. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Beijing, China. Association for Computational Linguistics, pp. 857–862.

Wang C., Xue N. and Pradhan S. (2015b). A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado. Association for Computational Linguistics, pp. 366–375.

Wang T., Wan X. and Jin H. (2020a). AMR-To-Text generation with graph transformer. *Transactions of the Association for Computational Linguistics* **8**, 19–33.

Wang T., Wan X. and Yao S. (2020b). Better AMR-To-Text generation with graph structure reconstruction. In Bessiere, C. (ed), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. International Joint Conferences on Artificial Intelligence Organization. Main Track, pp. 3919–3925.

Weischedel R., Hovy E., Marcus M., Palmer M., Belvin R., Pradhan S., Ramshaw L. and Xue N. (2011). OntoNotes: A large training corpus for enhanced processing. In *Handbook of Natural Language Processing and Machine Translation*, vol. **59**. Springer.

Werling K., Angeli G. and Manning C. (2015). Robust subgraph generation improves Abstract Meaning Representation parsing. arXiv preprint arXiv:1506.03139.

Xu D., Li J., Zhu M., Zhang M. and Zhou G. (2020. Improving AMR parsing with sequence-to-sequence pre-training. arXiv preprint arXiv:2010.01771.

Xue N., Bojar O., Hajič J., Palmer M., Urešová Z. and Zhang X. (2014). Not an interlingua, but close: Comparison of English AMRs to Chinese and Czech. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, Reykjavik, Iceland. European Language Resources Association (ELRA), pp. 1765–1772.

Xue N., Bos J., Croft W., Hajič J., Huang C.-R., Oepen S., Palmer M. and Pustejovsky J. (eds) (2020). *Proceedings of the Second International Workshop on Designing Meaning Representations*, Barcelona Spain (online). Association for Computational Linguistics.

https://doi.org/10.1017/S1351324922000183 Published online by Cambridge University Press

Xue N., Croft W., Hajic J., Huang C.-R., Oepen S., Palmer M. and Pustejovksy J. (eds) (2019). *Proceedings of the First International Workshop on Designing Meaning Representations*, Florence, Italy. Association for Computational Linguistics.

Xue N. and Palmer M. (2009). Adding semantic roles to the Chinese Treebank. *Natural Language Engineering* **15**(1), 143–172.

Xue N., Xia F., Chiou F.-D. and Palmer M. (2005). The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering* **11**, 207–238.

Žabokrtský Z., Zeman D. and Ševčıková M. (2020). Sentence meaning representations across languages: What can we learn from existing frameworks? *Computational Linguistics* **46**(3), 605–665.

Zhao Y., Chen L., Chen Z., Cao R., Zhu S. and Yu K. (2020). Line graph enhanced AMR-to-text generation with mix-order graph attention networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics, pp. 732–741.

Zhang S., Ma X., Duh K. and Van Durme B. (2019a). AMR parsing as sequence-to-graph transduction. arXiv preprint arXiv:1905.08704.

Zhang S., Ma X., Duh K. and Van Durme B. (2019b). *Broad-coverage semantic parsing as transduction*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, pp. 3786–3798.

Zhou J., Xu F., Uszkoreit H., Qu W., Li R. and Gu Y. (2016). AMR parsing with an incremental joint model. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas. Association for Computational Linguistics, pp. 680–689.

Zhu H., Li Y. and Chiticariu L. (2019). Towards universal semantic representation. In *Proceedings of the First International Workshop on Designing Meaning Representations*, Florence, Italy. Association for Computational Linguistics, pp. 177–181.

Cite this article: Oral E, Acar A and Eryiğit G. Abstract meaning representation of Turkish. *Natural Language Engineering* https://doi.org/10.1017/S1351324922000183