Two languages, one treebank: building a Turkish–German code-switching treebank and its challenges

Özlem Çetinoğlu¹ · Çağrı Çöltekin²

Accepted: 17 December 2021
© The Author(s) 2022

Abstract This paper presents the SAGT Turkish–German code-switching treebank, and observations and annotation challenges we encountered during its development. The treebank consists of transcriptions of bilingual conversations annotated with several layers: language IDs, lemmas, POS tags, morphological features, and dependency relations. The annotations follow the Universal Dependencies annotation scheme and the conventions used in monolingual treebanks as much as possible. We present and discuss a number of issues that arise because of the need for consistent multilingual annotation within a single treebank, as well as the informal language, which is where code-switching is observed most. Besides proposing solutions to these issues, we present some observations about code-switching phenomena that are only possible to observe in a data set with rich linguistic annotation. The treebank was annotated with a focus on quality of annotations through an iterative process of detecting and correcting annotation errors. We also present quantitative measures for indication of annotation quality. The code-switching treebank created in this study is released to the public through Universal Dependencies repositories.

Keywords Turkish · German · Code-switching · Treebank · Universal dependencies

✉ Özlem Çetinoğlu
ozlem@ims.uni-stuttgart.de
Çağrı Çöltekin
ccoltekin@sfs.uni-tuebingen.de

¹ IMS, University of Stuttgart, Stuttgart, Germany

² Department of Linguistics, University of Tübingen, Tübingen, Germany

1 Introduction

Code-switching (CS) is the process of mixing more than one language in written or spoken communication (Myers-Scotton 1993; Poplack 2001; Toribio and Bullock 2012). It is a phenomenon commonly observed in multilingual societies (Auer and Wei 2007), mainly in informal settings such as social media and spoken communication. For instance, (1) shows a sentence from a dialogue that mixes Turkish and German (in bold).¹ The speaker starts with Turkish, switches to German, goes back to Turkish, and ends the sentence with a mixed word where the German noun *Gastfamilie* ‘host family’ is inflected with the Turkish Locative suffix *-de*.

- (1) Eh orada iki **Wochen** kal-dı-m **ehm Gastfamilie-de** ehm .
Eh there two week-PL stay-PAST-1SG ehm guest family-LOC ehm .
 ‘I stayed there two weeks, in a host family.’

The sentence is relatively simple and the overall meaning is derivable from the individual words. Yet, its syntax is not standard. The main predicate *kaldım* ‘I stayed’ is in Turkish and the whole sentence seemingly follows the Turkish syntax, except the noun phrase *iki Wochen* ‘two weeks’. Nouns modified by numbers are in singular in Turkish, but *Wochen* is in plural. The construction is more complex than using the German equivalent of ‘week’ in a Turkish phrase. It seems, when the speaker switches to German on the surface, s/he inherently switches to the German syntax as well, where the noun should be plural.

Such CS-specific constructions vary from non-canonical morphological marking to creating new syntactic representations, to applying a linguistic phenomenon of one language to the other. They make structural analysis of code-switching linguistically interesting and computationally challenging. Several approaches tackle these challenges by utilising labelled and unlabelled monolingual and parallel data, e.g., by creating artificial CS data and using them in training models for processing CS (Pratap et al. 2018; Zhang et al. 2018). However, to be able to capture unique cases like the singular-to-plural mapping for ‘week’ in (1), those models need to see such instances. Thus, to observe the characteristics of CS and address them with data-driven tools, we present a treebank, namely the SAGT treebank, of Turkish–German transcriptions with language ID, lemma, part-of-speech (POS), morphology, and dependency layers.

We have chosen Universal Dependencies (UD, Nivre et al. 2016, 2020; de Marneffe et al. 2021) as our annotation scheme. The UD project aims to define morphosyntactic annotation guidelines that are consistent across languages. Its unified tag sets and annotation standards facilitate the annotation of multiple languages within a single treebank. Furthermore, annotations parallel to monolingual resources are useful for making use of these resources, e.g., for transfer learning (Bhat et al. 2018).

¹ Throughout the paper, when an example contains both Turkish and German tokens, German tokens are represented in bold. The morphosyntactic tags used in glosses are explained in Table 2 in Appendix 1.

Despite clear advantages of the UD framework for annotating CS treebanks, the annotation of multiple languages in a single treebank needs additional considerations that have not been studied before. Although there have been a few UD treebanks with code-switching (Bhat et al. 2018; Partanen et al. 2018; Seddah et al. 2020; Braggaar and van der Goot 2021), the papers describing these treebanks do not document or discuss the code-switching aspects of the annotation process, except for a brief section in (Braggaar and van der Goot 2021).

In this paper we address this gap and outline some of the challenges and interesting phenomena that surface during the annotation of a Turkish–German code-switching treebank. Our main contribution is the publicly available treebank, that also has accompanying audio files,² which enable multimodal studies. Our goal is, however, to go beyond resource description and contribute to both code-switching and treebanking communities with our observations and discussions. The observations on code-switching, independent of the annotation scheme, help in understanding in what forms it occurs. The annotation solutions we propose explore how to handle CS within the UD framework. Working with spoken data brings another aspect and also opens speech annotation under UD to discussion.

2 Related work

Many well-known linguistic theories on CS syntax, e.g., Free Morpheme and Equivalence Constraints (Poplack 1980), Closed-class Constraint (Joshi 1982), Matrix Language Frame (Myers-Scotton 1993), Functional Head Constraint (Belazi et al. 1994) define their formalism and constraints on constituency structures. Eppler (2005) argues that these constraints are too restrictive from a data-driven perspective and favours Word Grammar (Hudson 1990), a dependency-based formalism, where the scope of the constraints is head-dependent pairs. Her annotations on German–English transcriptions and the Chinese–English treebank (Wang and Liu 2013), which also follows Word Grammar, are the only CS dependency treebanks that do not follow UD, to the best of our knowledge.

The starting point for our work is the monolingual UD treebanks of both languages in our study. The current 2.8 release of UD includes eight Turkish and four German treebanks. Turkish treebanks include IMST-UD (Sulubacak et al. 2016b), which is semi-automatically converted from the IMST treebank (Sulubacak et al. 2016a) which, in turn, is a re-annotation of the METU-Sabancı treebank (Ofłazer et al. 2003). Turkish GB (Çöltekin 2015) is a manually annotated treebank consisting of grammar book examples. The Turkish BOUN treebank (Türk et al. 2020) is a more recent treebank annotating sentences from five different text types. Version 2.8 of UD introduced four new Turkish treebanks.³ To the best of our

² The audio files are available for research purposes, subject to a licence agreement, due to EU data privacy regulations.

³ https://universaldependencies.org/treebanks/tr_framenet/index.html
https://universaldependencies.org/treebanks/tr_kenet/index.html
https://universaldependencies.org/treebanks/tr_penn/index.html
https://universaldependencies.org/treebanks/tr_tourism/index.html.

knowledge, these new treebanks are not yet described in a publication. There are PUD treebanks consisting of parallel (translated) sentences for both languages. The PUD treebanks were automatically converted from another dependency scheme for the CoNLL 2017 multilingual parsing shared task (Zeman et al. 2017). The first German UD treebank is the GSD treebank (McDonald et al. 2013), which is also automatically converted from a different dependency formalism. Version 2.4 introduced two new additions to German treebanks; HDT, a conversion of Hamburg Dependency Treebank (Foth et al. 2014; Hennig and Köhn 2017), and LIT, a treebank of German literary history.⁴ There is also a treebank of German tweets, tweeDe (Rehbein et al. 2019), which has not yet taken its place on the UD repositories.

Most of our annotation decisions and the discussions below are based on the version 2.4 of the UD treebanks, particularly Turkish IMST, and German GSD and HDT. There are, however, inconsistencies across languages, and across treebanks of the same language. For most annotation decisions, we follow the annotations in the monolingual treebanks as much as possible. In case of inconsistencies across treebanks, our policy is to choose the alternative closest to the general UD guidelines, so as to ensure cross-lingual consistency within our multilingual treebank.

None of the treebanks noted above include spoken language, let alone code-switching. Quite a few UD treebanks, on the other hand, contain spoken language partially (Danish DDT, English GUM, English LinES, Greek GDT, Latvian LVTB, Khunsari AHA, Nayini AHA, Persian Seraji, Polish LFG, Scottish Gaelic ARCOSG, Skolt Sami Giellagas, Soi AHA, South Levantine Arabic MADAR, and Swedish LinES) or fully (Beja NSC, Cantonese HK, Chinese HK, Chukchi HSE, French Spoken, Naija NSC, Norwegian NynorskLIA, and Slovenian SST). These treebanks have extended the UD dependency relations with subtypes, in addition to using the existing ones to cover linguistic phenomena mainly observed in speech. For example, Slovenian SST (Dobrovoltjc and Nivre 2016) annotates correcting disfluencies either with `reparandum` or `parataxis:restart`. Another `parataxis` subtype, `parataxis:discourse` is defined to cover sentential parentheticals with fixed semantics that serve as discourse elements (e.g., *you know*). French Spoken (Gerdes and Kahane 2017) and Naija NSC (Courtin et al. 2018) employ the same tag too. They define a separate tag called `parataxis:dislocated` for clauses that precede the sentence they are dislocated from. The other relation that is commonly extended is `discourse`. Slovenian SST separates filler sounds from other discourse elements and assigns them the relation `discourse:filler`. Norwegian NynorskLIA (Øvrelid and Hohle 2016) follows the same approach. Cantonese HK and Chinese HK (Leung et al. 2016) define the tag `discourse:sp` for sentence particles common in spoken language. So far we are more conservative in extending relations with subtypes and have introduced two that are described in Sect. 4.1.

The Hindi-English UD treebank (Bhat et al. 2018) annotates the mixed language of social media and has no extension to UD dependencies. The major annotation

⁴ https://universaldependencies.org/treebanks/de_lit/index.html.

augmentation is the language IDs assigned to each token. Komi-Zyrian IKDP (Partanen et al. 2018) consists of spoken language, and some utterances include Russian phrases. In those utterances mixed and Russian tokens are marked with respective language IDs, and the Russian syntax is applied. However, the authors do not claim any consistency with the annotations of the monolingual Russian UD treebank. Similar to these treebanks, we also assign a language ID to each token following the tag set in (Çetinoğlu 2016). Many other treebanks include words or phrases from a foreign language. Most of them mark foreign tokens with `Foreign=Yes`, and annotate the internal structure of foreign phrases with `flat` relations. However, a few treebanks, e.g., Irish IDT (Lynn and Foster 2016), annotate foreign tokens according to their respective language. Recently, two code-switching treebanks are released. The first contains Romanised Algerian sentences from social media, hence the language is a mix of Northern African Arabic with mainly French and Modern Standard Arabic (Seddah et al. 2020). The second one is a Frisian-Dutch treebank that annotates radio transcriptions (Braggaar and van der Goot 2021). Both treebanks use the existing UD relations and subtypes, although the former one does not fully comply with UD yet.

3 Data collection and transcription

The data collection, all steps of annotation, and anonymisation processes are handled by a team of three bilingual Computational Linguistics students.⁵ In this section we give the details of data collection and transcription.

3.1 Collection

The data collection is done by the annotators as conversation recordings. The annotators approached Turkish–German bilinguals mostly from their circle for an informal setting, assuming this might increase the frequency of code-switching. Similarly we recommended the annotators to open topics that might induce code-switching. Example topics are work and studies (i.e., typically German-speaking environments) in a dialogue started in Turkish, or holidays and food in Turkey (hence Turkish-specific words) in a German-dominated conversation.

There are 48 distinct conversations in our collection. 20 participants took part in the recordings. The majority of the speakers are university students; hence, the most frequent age range is 18–25. Common conversation themes include studies, work, travel, future plans, and free time activities such as sports, books, and TV.

⁵ Born and raised in Germany to families with Turkish migration background, they are native speakers of both languages.

3.2 Transcription layers

The transcription and annotations are done using Praat.⁶ We created six tiers for each audio file: `spk1_verbal`, `spk1_norm`, `spk2_verbal`, `spk2_norm`, `lang`, `codesw`. The first four tiers contain the verbal and normalised transcriptions of speakers 1 and 2. Tier definitions are given in Sect. 3.3. The tier `lang` corresponds to the language of intervals and can have `TR` for Turkish, `DE` for German, and `LANG3` for utterances in other languages. The first five tiers are intervals, while the last one is a point tier that denotes sentence and code-switching boundaries. The labels on the boundaries are `SB` when both sides of the boundary are in the same language `SCS` when the language changes from one sentence to the next (intersentential), `WCS` when the switch is between words within a sentence (intrasentential).

Since Turkish is agglutinative and case markers determine the function of NPs, non-Turkish common and proper nouns with Turkish suffixes are commonly observed in CS conversations. We mark such words in the `codesw` tier as an intra-word switch and use the symbol § following Çetinoğlu (2016). Example (2) depicts the representation of a mixed word where the German noun *Semester* ‘semester’ (in bold) is followed by the Turkish locative case marker *-da*. Figure 5 in Appendix 1 demonstrates the Praat representation of this word as part of a full sentence. The § and `WCS` boundaries, and tiers could be observed from the figure.

- (2) **Semester** § -da
semester -LOC
 ‘in the semester’

For many proper names and for some loan words, Turkish and German orthography are identical. Here, the speech data in parallel becomes an advantage, and the language is decided according to the pronunciation. If the word is pronounced in German and followed by a Turkish suffix, a § switch point is inserted. Otherwise it follows the Turkish orthography. *Master§da* ‘in Master’s’ from (5) is such an example where German *Master* is followed by Turkish locative suffix *da*. While Turkish has a translation for Master’s (i.e., ‘yüksek lisans’), it is also common to use the term as *Master* in academic communities. Thus, orthographically *Masterda* is ambiguous. However, in German *Master* is pronounced similar to /*mast3*/, while in Turkish pronunciation would be /*mastuɯ*/. Following the pronunciation in the audio, we take *Master* as a German word hence *Master§da* becomes a mixed one.

3.3 Transcription guidelines

For speech analysis it is important to transcribe utterances close to how they are pronounced. In some transcription guidelines, capitalisation and punctuation are omitted (e.g., in the SEAME corpus by Lyu et al. 2015);⁷ in some others they are used to mark speech information (e.g., in the Kiezdeutsch corpus by Rehbein et al.

⁶ <http://www.fon.hum.uva.nl/praat>.

⁷ <https://catalog.ldc.upenn.edu/docs/LDC2015S04/SEAME.V4.0.pdf>.

2014).⁸ Text analysis on the other hand generally relies on standard orthography. This raises a conflict between two tasks on how to transcribe speech. To avoid this problem, we introduced two tiers of transcription. The verbal tier follows the speech conventions. If a speaker uses a contraction, the word is transcribed as contracted. The acronyms are written as separate characters. Numbers are spelled out. Recurring characters are represented with the single character followed by a colon. The normalised tier follows the edited text conventions. Words obey the orthographic rules of standard Turkish and German, e.g., characters of acronyms are merged back. Punctuation is added to the text, obeying the tokenisation standards (i.e., separated from the preceding and following tokens with a space).

Example (3) gives a sentence showing the verbal and normalised tiers for a Turkish sentence. The *r* sound in the progressive tense suffix *-yor* is not pronounced, hence omitted in the verbal tier. The vowel of the interjection *ya* is extended during speech, and the colon representation is used to reflect it in the verbal tier, yet the normalised tier has the standard form. Also, the question mark is present in the normalised tier.

- (3) verbal: ne di-yo-sun ya:
 norm: Ne di-yor-sun ya ?
 What say-PROG-2SG INTJ
 ‘What are you saying??’

If a made-up word is uttered, it is preceded with an asterisk mark in the transcription. Note that dialectal pronunciation or using a valid word in wrong context is not considered within this class. Partial words are marked with two hyphens instead of the common use of one hyphen, as the latter is used in German to denote the initial part of a compound when two compounds share a common part and the first compound is written only as the unshared part (e.g., *Wohn- und Schlafzimmer* ‘living room and bedroom’).

We also marked [silence], [laugh], [cough], [breathe], [noise], and put the remaining sounds into the [other] category. Overlaps occur usually when one speaker is talking and the other is uttering backchannel signals and words of acknowledgement.⁹ There are also cases when both speakers tend to speak at the same time. In all such cases, both voices are transcribed, one speaker is chosen to be the main speaker, and an [overlap] marker is inserted to the secondary speaker’s verbal and normalised tiers. The *codesw* and *lang* tiers are decided according to the main speaker’s transcription.

⁸ <http://www.kiezdeutschkorporus.de/files/kidko/downloads/KiDKo-Transkriptionsrichtlinien.pdf>.

⁹ Backchannel signals are a listener’s responses to a speaker to express reactions such as attention, understanding, or confirmation. Common backchannel and acknowledgement utterances from our corpus include *mh, ja/evet* ‘yes’ in German and Turkish, *genau* ‘exactly’ in German and *doğru* ‘right’ in Turkish.

3.4 Anonymisation

In data collection, processing, and maintenance we closely follow EU General Data Protection Regulation (GDPR)¹⁰ in collaboration with the Central Data Protection Office of Baden-Württemberg Universities (ZENDAS).¹¹

As part of these data protection policies, we anonymise sensitive data. In deciding how to anonymise data, we had two concerns in mind. First, the anonymised sentence should be plausible syntactically and semantically so that it is not possible to infer anonymisation is applied. Second, the semantics of the data should change as minimally as possible so that common world knowledge could still be utilised (e.g., via word embeddings).

We employed a selective pseudonymisation strategy for data protection (Medlock 2006). We replaced information that is sensitive in context. In (4) the city where the speaker lives is personal information and should be anonymised. However, the second sentence is an opinion that could not be attributed to a specific person. Thus, we anonymise *Dresden* with *Leipzig*, but keep the cities in the second sentence untouched.

- (4) I live in *Dresden*. Berlin, Hamburg, and Munich are too big to live.
I live in *Leipzig*. Berlin, Hamburg, and Munich are too big to live.

We paid attention to three criteria in choosing replacements: phonological parallelism (for Turkish), syntactic structure, and semantic consistency. Turkish phonology employs vowel harmony and consonant alternations that change the surface realisation of suffixes. For instance, when female names *Ece* and *Bahar* are inflected in Genitive case they have the forms *Ece'nin* and *Bahar'ın*, respectively. When we anonymise *Ece*, we choose another female name ending with the same vowel, e.g., *Ayşe*, so that the inflected form *Ayşe'nin* preserves the correct orthography.

Sometimes the information to anonymise consists of multiple tokens with a syntactic structure, e.g., titles, organisations, or locations. In such cases we find replacement with exact number of tokens and syntactic structure, e.g., *Esslingen am Neckar* ‘Esslingen on the Neckar’ is replaced with *Erlenbach am Main* ‘Erlenbach on the Main’. We also take into account the broader context; that is, the transcribed conversation. For instance if other sentences of the conversation explicitly mention that a city is by the seaside and has a small population, we found a replacement city with similar properties. With these criteria, we made sure that anonymisation process does not change annotation layer properties. All anonymisation is done manually. Each transcription is anonymised by two annotators to ensure privacy. Disagreements are resolved via meetings.

¹⁰ <https://gdpr-info.eu/>.

¹¹ <https://www.zendas.de/>.

4 Treebank annotation and statistics

The treebank sentences to annotate consist of the normalised tier of the transcriptions. In this section we first define the annotation layers and process, then give statistics about the overall treebank. All annotations, except attaching punctuation, are done manually. Segmentation, lemmatisation, POS tagging, and morphological analysis are handled together as the first step. Dependency annotations follow as the second step.

4.1 Annotation layers

Segmentation and Intra-word CS Following UD, we take *syntactic* words¹² as units of annotation. This results in segmenting some of the surface tokens in both German and Turkish. For German, the only case that require segmentation is the contraction of prepositions and definite articles. For example, the word *ins* ‘to the’ is tokenised into its parts as *in* and *das*. The segmentation of Turkish syntactic words is more involved. We follow previous work (Çetinoğlu and Çöltekin 2016; Sulubacak et al. 2016b) and segment copular suffixes and a group of productive derivational suffixes. For instance, in *öğrenciyim* ‘I am a student’, the copula *yim* ‘I am’ is split, thus the segmented form is *öğrenci yim*. The productive suffix *-li* derives nouns and adjectives from a noun (N) with the meaning of ‘with N’ (e.g., *mavi elbise* ‘blue dress’ → *mavi elbise* ‘li(the one) with a blue dress’). *elbiseli* is segmented into *elbise li* in this case. However, we do not segment lexicalised derived words. For instance, *yaşlı* ‘old’ comes from *yaş-li* ‘age-with’ but it is completely lexicalised.

Intra-word CS boundaries coming from transcriptions are kept as a feature in the MISCellaneous column of the CoNLL-U file (see Fig. 6 in Appendix 1). Namely, the CSPoint feature indicates the switch, e.g., CSPoint=Termin\$im, where *Termin* ‘appointment’ is in German and *-im* ‘my’ is Turkish possessive suffix.

Language IDs We follow the tag set of Çetinoğlu (2016) for language IDs. We take the TR, DE, and LANG3 labels of intervals from the transcriptions and map them to each token of the interval. When a token has intra-word CS, then the LangID value is MIXED. Punctuation and special symbols get an OTHER value. Segmentation might cause MIXED words to split such that segments’ language IDs change, e.g., to DE and TR (cf. (Çetinoğlu and Çöltekin 2016)). In such cases annotators manually assign the new tags. The language IDs are kept as the LangID feature in the MISC column of the CoNLL-U file.

Lemmas, Part-of-Speech Tags and Morphological Features We use the Universal POS tag set of Universal Dependencies¹³ for POS tagging and employ the individual morphological tag set of Turkish and German treebanks.¹⁴ Note that the tag sets used might cause different semantics of the same representation between

¹² Segmentation is not in the morpheme level, yet words are not necessarily phonological or orthographic. <http://universaldependencies.org/u/overview/tokenization.html>.

¹³ <https://universaldependencies.org/u/pos/index.html>.

¹⁴ https://universaldependencies.org/treebanks/tr_imst/index.html
https://universaldependencies.org/treebanks/de_gsd/index.html.

Fig. 1 Sentence length distribution (left) and distribution of number of code-switches per sentence (right)

two languages. For instance in German prepositional phrases `Dat` case marker indicates a ‘state’ meaning, whereas in corresponding Turkish phrases the same value indicates a ‘movement’, and there is a separate `Loc` case for ‘state’.

Dependency Relations Our tag set consists of the combination of tag sets of Turkish and German treebanks. Only two new subtypes are added to the tag set: *appos:trans* and *parataxis:trans*. Both are introduced to handle *translation pairs*; the details are discussed in Sect. 6.1.4.¹⁵

4.2 Annotation process

For Turkish tokens we obtained all possible lemma, POS tag, and morphological feature combinations using a finite-state morphological analyser for Turkish (Çöltekin 2010). This analyser also gives the segmentation boundaries as part of the analyses. For German, we derived such possible analyses from the GSD treebank¹⁶ since there is no UD features-compatible morphological analyser.

We gave the annotators all possible analyses in a random order and asked them to disambiguate the tokens with ambiguous analyses manually, using an in-house tool. The tool provided a dropdown list for ambiguous analyses to choose from as well as individual dropdown lists for tags and features so that the annotators can overwrite values or assign their own analyses. The latter was useful especially when a `MIXED` or foreign word did not have any provided analyses, and when German words are not covered by the GSD treebank.

During dependency annotation, the language ID, lemma, POS tag, and morphological features of each token were available to the annotators from the previous annotation step. The annotators were provided with an annotation guidelines document covering the union of Turkish and German dependency relation sets, as well as instruction on how to handle CS related issues. Once all manual annotation tasks were complete, punctuation is attached automatically using UDApi (Popel et al. 2017).

¹⁵ The UD dependency descriptions can be found in (de Marneffe et al. 2021) as an overview table and at <https://universaldependencies.org/u/dep/all.html> in detail. The additional documentation for the SAGT treebank could be found at <https://universaldependencies.org/qtd/index.html>.

¹⁶ When we built the morphological dictionaries, GSD was the only large German treebank available.

Fig. 2 The POS tag distribution in the CS treebank (middle bar in each group) in comparison to IMST (left) and HDT (right). Each group presents percentage of the POS within each treebank. The bars representing CS treebank is partitioned based on the languages of the tokens. PUNCT, X and SYM tags are excluded

Since the standard parsing task is defined on sentences as input units, intersentential CS boils down to parsing monolingual sentences. For this reason, we include in our treebank only sentences with intrasentential and intra-word CS. Due to limited time and funding, we transcribed only sentences that are included in the treebank.

4.3 Statistics

The treebank contains 2184 sentences and 36,940 (surface) words. Since some of the words from both languages were segmented as multiple syntactic words following UD guidelines, the treebank contains 37,233 syntactic tokens, after segmenting 290 of the words. The length of the sentences in the treebank is on average 17.05 tokens, with a minimum of 2, maximum 83 and a standard deviation of 9.66. The distribution of length of the sentences are presented in the left panel of Fig. 1. All sentences in the treebank contains at least one CS point, with a maximum of 20 in one example.

The overall distribution of the number of CS per sentence is given in the right panel of Fig. 1. Single CS points have relatively balanced patterns; in 461 cases, speakers start with German and finish with Turkish, and in 489 cases, speakers start with Turkish and finish with German. For the cases of two CS points, it is slightly more common for speakers to insert Turkish words and phrases into otherwise German sentences (277 cases) than inserting German words in Turkish sentences (310 cases).

The distribution of language IDs across two languages is slightly imbalanced in favour of German. The treebank contains 18,741 German (DE) and 14,179 Turkish (TR) tokens. Note that this imbalance does not necessarily indicate more dominant usage of German. The number of tokens in German is expected to be higher since

Fig. 3 Distribution of dependency labels per language ID. We do not include OTHER and LANG3 language IDs, as well as dependency types that are used only for one language (advmod:emph, aux:q for Turkish, and aux, aux:pass, compound:prt, expl, iobj for German)

many linguistic functions expressed by words in German are expressed by means of morphemes in Turkish (see Sect. 5.1.3 for examples). There are also 429 words with intra-word CS, most of which are German nouns and proper nouns with Turkish suffixes. Although Turkish also has an even larger set of verbal affixes, the common way to modify a German verb through Turkish suffixes seems to be the use of light-verb constructions discussed in Sect. 6.1.3. Less interestingly, the treebank also

contains 179 tokens from other languages (predominantly English, tagged as `LANG3`), and 3704 added punctuation tokens tagged as `OTHER`.

We present the POS tag distribution of the CS treebank in comparison to monolingual Turkish (IMST) and German (HDT) treebanks in Fig. 2. As expected from the spoken aspect of the data, our treebank is characterised by more frequent use of interjections, pronouns and adverbs in comparison to the monolingual treebanks covering standard/written language. The difference in the distributions of POS tags across languages in our treebank is also expected. Notably, there are proportionally more German pronouns (presumably because the differences in pro-drop), and subordinating conjunctions (since subordination in Turkish is typically indicated with suffixes rather than subordinating conjunctions) in comparison to Turkish. A similar trend is also visible for auxiliaries, determiners and adpositions.

The analysis of dependency labels in comparison to monolingual treebanks indicates similar, expected observations. The spoken CS treebank has a substantially large rate of `advmod` dependency, and also dependencies that are very rare or non-existent in monolingual treebanks, such as `reperandum`, `discourse` and `parataxis`. We also present the distribution of dependency labels within the language IDs in our treebank in Fig. 3 in Appendix 1. In general, the dependency distributions also show expected differences between two languages, e.g., more `det`, `nsubj` and `mark` relations for German parallel to the usage of POS tags `DET`, `PRON` and `ADP` discussed above. Our two new dependency types `appos:trans` and `parataxis:trans` deserves a closer look. The dependency `appos:trans`, indicating an apposition relation with exact translation of the head, have almost equal numbers of dependents in both languages. The difference in `parataxis:-trans`, which marks longer parenthetical expressions that are translations of (the constructions headed by) their head, indicates that the speakers more often explain their German expressions in Turkish. We look into the details of these new dependency types as well as many other dependency relations in the following sections.

5 Annotation differences in individual languages

Any annotation project is bound to make non-trivial choices (Gerdes and Kahane 2016). Most non-trivial choices for a code-switching treebank comes either because of the multilingual nature of the resource, or, as noted earlier, the fact that code-switching is prevalent in informal language, and annotation of informal or spoken language has been more challenging than more standard/written language. Most of the problems related to the multilingual nature of the data stem from different annotation choices established for individual languages. In this and the following sections we group these challenges and observations to discuss each of them with examples. This section focuses on differences that come from inherent properties of individual languages and from annotation decisions of monolingual treebanks.

5.1 Individual language characteristics

As in any language pair, Turkish and German express some semantic equivalents in different (morpho)syntactic representations. These phenomena are noteworthy for our treebank as it was sometimes confusing for annotators to follow individual approaches.

5.1.1 Copula

One of the principles of Universal Dependencies is the primacy of the content words. For copular constructions, this means marking the copula as the dependent rather than the head. Turkish has no explicit copular marker for third person singular in present tense and for other cases, they are realised as suffixes (Göksel and Kerslake 2005, p. 78) that are segmented in the UD representation (Çetinoğlu and Çöltekin 2016). Therefore the Turkish treebanks naturally follow ‘copula as a dependent’ for all types of copular constructions. On the other hand, in German the copula *sein* is a separate word (Eisenberg 2013, p. 79), and the German GSD, PUD and LIT treebanks seem to make distinction where some its uses are annotated as the main verb. For example, these treebanks suggest that copula *ist* in *Die Frau ist Ärztin* ‘the woman is a doctor’ should be annotated using `cop` (with head *Ärztin*), while in *Der Vortrag ist in dem großen Saal* ‘The lecture is in the great hall’, it should be marked as the main verb. HDT, on the contrary, marks them with `cop`, in accordance with the general and Turkish guidelines. Thus, we follow HDT in copula annotation.

5.1.2 Expletives

As in many other non-pro-drop languages, German has an expletive pronoun, namely *es* ‘it’ (Eisenberg 2013, p. 174). In UD, it is attached to the main predicate of the sentence with the `expl` relation as exemplified in (5).¹⁷ Turkish on the other hand is pro-drop and constructs semantic structures that correspond to expletive uses in German either with copular constructions, as in (6) or with intransitive verbs (e.g., in case of weather verbs). Existential clauses are formed with the adjectives *var/yok* ‘there is/there isn’t’ (Underhill 1976, p. 103) and follow the standard UD analysis for non-verbal predicates. Therefore, `expl` relation is used only for German in the treebank.

¹⁷ We shorten this example and some of the examples presented later for the sake of readable representation within page margins. We opted for removing tokens that have relatively less weight in the syntactic representation such as punctuation, interjections, adverbs, unless they are the focus of the discussion. All CS examples are from the presented treebank and their original version are available in the UD repositories.

5.1.3 Subordination

Many linguistic phenomena that are realised at the word level in German are realised at the morpheme level in Turkish. For example, the German preposition in (7a) corresponds the Turkish case marker in (7b).

- (7) a. in die Schule
in the school
‘to the school’
- b. okul-a
school-DAT
‘to the school’

The majority of subordinate clauses is one of such constructions where German employs subordinating conjunctions and Turkish uses derivational suffixes. (8) shows the use of the German subordinating conjunction *wenn* ‘when’.

The sentence in (9) demonstrates the use of *ken* ‘while’ in Turkish. The adverbial suffix is attached to the main predicate of the subordinate clause. Since the UD policy is to preserve the base POS tag, the derived word is still a *VERB*. The adverbial derivation is annotated in the *VerbForm=Conv* feature.¹⁸ The subordinate clause is attached to the main clause with the *advcl* relation.

When *ken* is attached to a nominal predicate, it is subject to segmentation as all other copular suffixes. The suffix *ken* is the bound form of the unbound morpheme *iken*, which bears the stem *i* ‘be’ (Kornfilt 1997, p. 72). Due to this copular nature, the split suffix is attached to the clausal head with *cop* and the nominal predicate becomes the *advcl* of the main clause, as in (10).

5.2 Language-specific annotation choices

To be able to benefit maximally from monolingual treebanks, one of the principles we follow is to annotate the tokens that belong to each language following the annotation standards in the monolingual treebank(s) of the corresponding language. However, we have observed that even for linguistic phenomenon that are represented in individual languages in parallel ways, there could be different annotation choices. When combined with the different annotation choices within the treebanks of a single language, design decisions get harder to make and often require some compromises. In this section we provide examples of such cases.

¹⁸ *Conv* is the abbreviation of *converb*.

5.2.1 Titles

A relatively simple difference between existing monolingual German and Turkish treebanks is the annotation of titles, e.g., as in *President Obama*. The UD guidelines prescribe the use of `flat` relation here. However, the different treebanks follow slightly different practices.¹⁹ German treebanks seem to annotate names using `appos` relation. In Turkish treebanks, similar to a few other treebanks in the UD distribution, the `nmod` relation is used. Although this is a relatively trivial issue, it demonstrates the trade-offs of the annotation choices. On the one hand, choosing one of three relations and applying to both languages would cause inconsistency with the (larger) monolingual treebanks and tools based on these treebanks. On the other hand, following the conventions of both languages causes inconsistency within the multilingual treebank, potentially confusing users querying the treebank, or automatic tools that are trained on it. There are only a few instances in our treebank, e.g., *Tolkien Reis* ‘Master Tolkien’, and we annotated them with the `flat` relation.

5.2.2 Possessive pronouns

In Turkish, possessive pronouns are personal pronouns in the genitive case as shown in (11) (Kornfilt 1997, p. 306). Hence, they are tagged as `PRON` and their dependency relation is `nmod`. In German, their behaviour is different. German possessive pronouns show Gender agreement with the nouns they modify (similar to determiner and adjectives) (Imo 2016, p. 86) as demonstrated in (12), where *Hut* ‘hat’ is masculine and *Tasche* ‘bag’ is feminine. The UD guidelines²⁰ recommend POS tag `DET` in such a case. The current German treebanks have different implementations. GSD identifies possessive pronouns as `DET` and employs `det:poss` as the dependency relation. HDT assigns a `PRON` label but `det` dependency attributing to both properties. The possessive information is stored in morphological features as `Poss=Yes`. We follow HDT in this case, which also makes the German annotation parallel to the Turkish one: possessive pronouns are labelled as `PRON` and possessivity is marked in morphological features. The dependency relations remain language-specific.

¹⁹ See <https://universaldependencies.org/workgroups/mwe.html> for a discussion.

²⁰ <https://universaldependencies.org/u/overview/morphology.html#pronominal-words>.

5.2.3 Zero derivation

A mismatch between the present treebank and the existing Turkish treebanks arises due to the way morphology of adjectives are annotated. Most adjectives in Turkish can be used as nouns without change to their form (so-called ‘zero derivation’). For example, the adjective *eski* ‘old’ in (13a) can be used as a noun meaning the ‘the old (one/item)’ (with an accusative case marker in this case) as in (13b).

- (13) a. Bana *eski* kitab-ı ver-di
I.DAT old book-ACC give-PAST.3SG
 ‘S/he gave me an old book.’
- b. Bana eski-yi ver-di
I.DAT old-ACC give-PAST.3SG
 ‘S/he gave me the old (one).’

In nominal usage of an adjective, Turkish (and also other Turkic language) UD treebanks keep the POS tag as ADJ, but include the features such as *Case* and *Number*. This practice is open to discussion since Turkish adjectives do not inflect.²¹ The motivation is to keep the zero derivation information implicitly, which would otherwise be lost in the UD annotation scheme. On the other hand, the German adjectives are marked for inflectional features indicating *Case* and *Number* as well as *Gender* and *Degree*. This result in parallel annotations within our multilingual treebank with different semantics. As a result, diverging from the annotation traditions of Turkish treebanks, we set the POS tags of zero-derived adjectives to NOUN.

5.2.4 Overt vs. context-based morphological annotation

The annotation traditions between two languages also differ with respect to the source of the morphological features. In Turkish, a morphological feature is marked only if there is an overt inflectional marker. For example, while *kitabı* in (13a) above is marked as *Case=Acc* because of the explicit accusative suffix *-ı* attached

²¹ The annotation of Turkish zero-derivation in the UD treebanks was discussed in earlier work (Çöltekin 2016; Tyers et al. 2017) without a clear solution.

to definite objects, since there is no morphological marker indicating the case, the indefinite version given in (14) is marked with the default case (Case=Nom) even though it fills an object position.

- (14) Bana *eski* kitap ver-di
 I.DAT *old book* give-PAST.3SG
 ‘S/he gave me an old book.’

Traditionally, German treebanks follow a different principle. The morphological features are assigned based on the context of the word. For example, even though the form does not change, the word *Buch* ‘book’ is annotated nominative, accusative and dative in examples sentences (15a), (15b) and (15c) respectively.

- (15) a. Das Buch gefällt mir.
 the book please.3SG I.DAT
 I like the book.
 b. Ich mag das Buch.
 I like the book
 I like the book.
 c. Das Bild ist in dem Buch.
 the picture is in the.DAT book
 The picture is in the book

Similarly, the Case feature of the determiner *das* in examples (15a) and (15b) are also annotated differently. To our knowledge, the UD specifications is also not clear which approach is preferable. We follow the tradition set in earlier treebanks for each language separately, making features that are indicated by the explicit morphological markers for Turkish, and indicating the features based on syntactic context for German. This annotation difference between treebanks of two languages does not affect most treebanks or applications built on them. However, as we demonstrate in Sect. 6.1.1, this type of language-specific annotation decisions may lead to conflicts in a CS treebank.

5.2.5 OBJs vs IOBJs vs OBLs

When it comes to canonical direct objects, both Turkish and German treebanks annotate them with the obj relation. However, they have different perspectives on annotating non-canonical or indirect objects.²² In German, canonical objects are marked with the ACC case; in Turkish, they bear the Nom or ACC case, depending on specificity. In ditransitives, the German treebanks assign the recipient an iobj label. However, in the Turkish IMST the GB treebanks, any dependent of a verb in

²² An overview of the relations between case markers and syntactic functions could be found in (Göksel and Kerslake 2005, Sect. 14.3.3) and in (Eisenberg 2013, Sect. 3.2.2) for Turkish and German, respectively. This overview could also be helpful for case marking discussed in Sects. 6.1.1 and 6.1.2.

cases other than nominative and accusative are annotated as `obl`.²³ This is an extension of mapping case markers to syntactic functions. In both languages there are verbs subcategorising for an object with a non-canonical case marker. UD denotes that if there is just one object, it should be labelled `obj`, regardless of the morphological case or semantic role. German UD treebanks follow this approach but Turkish UD treebanks do not.²⁴ Instead, they are labelled as obliques. We follow the approaches of respective treebanks. When the relationship is between a head-dependent pair from different languages, we choose the language of the head in deciding the approach.

6 Issues related to the nature of the data

In Sect. 5 we discuss the differences coming from annotation decisions or the languages themselves. Albeit non-standard, these differences can exist side by side when they do not interact. However, when code-switching occurs and these languages *interact* they give rise to new syntactic constructions or conflicts. Below we focus on the results of these multilingual interactions as well as noting some of the issues that are due to the informal and spoken language.

6.1 CS-specific issues

6.1.1 Conflicting case assignment

As discussed in Sect. 5.2.4, the treebanks of two languages in our study use different approaches for annotating morphological features. This brings challenging cases of annotation for a CS treebank. (16) presents an example of this conflict, with a German noun that functions as an object of a Turkish predicate.

According to German annotation standards, the word in the object position should be tagged as `Case=Acc`. However, there is no overt case marker,²⁵ thus the

²³ Genitive dependents in certain constructions are also marked as `obj`. Although there is no clear consensus, the main criterion of determining ‘coreness’ of a dependent in Turkic UD treebanks seems to be changing the case when the verb is passivised or causativised (Tyers et al. 2017).

²⁴ IMST has dative objects but they are labelled inconsistently.

²⁵ Note that *Bauingenieurwesen i*—the version with the Turkish accusative case marker—would also be grammatical indicating a definite subject instead of an indefinite one as in this example.

tag should be *Case=Nom* according to Turkish annotation standards. The principle of following the annotation scheme of the token’s language does not work well here, causing the loss of the distinction between definite and indefinite objects in Turkish. In such cases, we chose the language of the head as reference.

6.1.2 Double case marking

Annotating case marking can get more complicated when it is overt in both languages. In (17), the article *dem* ‘the’ and the number *dritten* ‘third’ carry the dative case marking to indicate the static meaning. The noun *Semester* normally does not carry an explicit marker and the German phrase *in dem dritten Semester* ‘in the third semester’ would be completely grammatical. Thus, the token *Semester* would normally have the tag *Case=Dat* in its morphological annotation in agreement with its modifiers. However, the speaker has chosen to mark the static meaning *also* in Turkish and following the Turkish grammar rules, there is a locative case marker *-da* attached to the noun, which entails a *Case=Loc* tag in its morphological representation.

The conflict between case markers does not have a perfect solution within the current UD representation. If we choose *Case=Dat* to follow the German rules, the surface form *-da* would not match the morphological tag, furthermore it would change the semantics of the word, as the dative case represents motion towards something in Turkish. Thus, we choose the *Case=Loc* tag at the expense of losing the agreement between the determiner and number, and the noun. We keep the German case by introducing a new feature *DeCase=Dat* in the MISC column.²⁶

6.1.3 Bilingual light verb constructions

The use of CS creates new constructions too. One quite common new construction is the use of German verbs followed by a Turkish light verb *etmek* ‘do’ or *yapmak*

²⁶ Another possibility is indicating both cases with notation *Case=Dat, Loc*. This is used when the word may have one of the values, but it cannot be decided from the available context. In this particular case, however, there is no ambiguity. Both case values are correct depending on the language.

Table 1 Inter-annotator agreement

Language	Tokens	UAA	LAA	LA	LA κ
Turkish	806	78.91	64.27	74.81	72.37
German	1155	79.65	71.95	83.29	81.28
Mixed	14	78.57	64.29	78.57	72.37
All	1976	79.35	68.78	79.81	77.68

Language indicates the language of the dependent token. *UAA* Unlabelled attachment agreement, *LAA* labelled attachment agreement, *LA* label agreement, κ : Cohen's kappa calculated for label agreement (multiplied by 100 for ease of comparison). Scores are calculated over 114 sentences annotated by two annotators. 130 punctuation tokens are ignored

‘make’, which is also observed in Turkish–German tweets (Çetinoğlu 2016) as well as Turkish–Dutch (Backus 2009).²⁷ The German verb is in infinitive form and the Turkish light verb takes inflectional and derivational suffixes. The core semantics of the construction comes from the German verb. These constructions are similar to noun-light verb constructions common in Turkish (e.g., *yardım etmek* lit. ‘help do’ – ‘to help’). In the Turkish UD, noun-verb constructions are labelled with the `compound:lvc` relation where `lvc` denotes light verb constructions. We adopt the same label for German-Turkish constructions. (18) demonstrates a sentence where the German verb *schnorcheln* ‘snorkel’ is coupled with the Turkish light verb *yap* ‘make’, that undergoes derivation with the suffix *-ken* ‘While’. The combined meaning of the compound is ‘while snorkelling’.

(18)

‘While you were snorkelling you had dropped your glass.’

6.1.4 Translation pairs

Another CS-specific language use we have observed is uttering a word, phrase or clause in one language and repeating it as a translation in the other language. (19) shows such an example where German *gehe auch* ‘I go too’ is repeated again as Turkish *de gidiyorum*. Since there are no relations in UD that would capture this phenomenon, we extend the relation `parataxis` by introducing a `trans` subtype.

²⁷ Such constructions are not specific to CS that mixes Turkish with another language. It is observed in many language pairs (Edwards and Gardner-Chloros 2007), such as Kyrgyz-Russian as one of our reviewers pointed out.

The relation connects the head of the second constituent to the head of the first constituent as a dependent.

‘I also go to the library now and then.’

Specific for noun phrases, we employ *appos:trans* where the dependent noun phrase is the translation of the head. In (20) for instance, Turkish *sen* ‘you’ is repeated again as German *du*. Note that (23) is a similar case but it is only *appos* rather than *appos:trans* as there is no translation involved.

‘But you always wanted that, right?’

6.1.5 Bilingual *m*-reduplication

In Turkish, it is possible to generalise the meaning of a word by so-called *m*-reduplication (Göksel and Kerslake 2005, p. 91). To realise *m*-reduplication, the first word is reduplicated, and an *m* prefixes the duplicate if the word starts with a vowel, or the first character of the duplicate is replaced with an *m* if it is a consonant as in (21).

- (21) Çay **may** iç-er mi-sin?
Tea etc. drink-AOR -QUES-2SG
 ‘Would you like to drink tea and the like?’

While this is a Turkish-specific phenomenon, bilinguals also apply it to other languages. In (22) we see that the German word *Trank* ‘potion’ undergoes *m*-reduplication. This is not only a new lexical alternation in German, its syntactic representation is new to German UD as well. *m*-reduplications are represented as

```
27. REGEX: cut -f 2,4,8 */*/wcs.annot.dep.conllu \
| grep -P "(A|a)ma\t|(V|v)e\t" | sort | uniq -c | sort -n -r | grep -v '#'
```

CHECK: The conjunctions annotated with 'mark' are most probably part of clauses attached with 'advcl' to the main clause. If they are corrected to 'CCONJ cc', then the clause head should be corrected to 'conj'.

Also, normally there is no 'ADV' use of 'ama' in Turkish that is similar to German. But check nevertheless, perhaps there are exceptions.

```
14 Ama ADV cc
13 ama ADV cc
7 ama CCONJ mark
5 ama ADV advmod
3 ama ADV mark
2 ve ADV cc
2 ama CCONJ discourse
1 ama CCONJ parataxis
1 Ama CCONJ advmod
1 Ama ADV advmod
```

```
39. REGEX: cut -f 2,4,8 */*/wcs.annot.dep.conllu | grep -P "yoksa" \
| sort | uniq -c | sort -n -r | grep -v '#'
```

CHECK: 'yoksa' could be used in three different ways:

- 'Fazla ödevim yoksa gelirim.' the predicative meaning, equivalent to 'yok ise'. The sentence is parallel to 'Fazla ödevim varsa gelemem'. Here 'yoksa' is segmented. 'yok' is ADJ and 'sa' is AUX.
- 'Mühendis misiniz yoksa teknisyen mi?' coordinating 'yoksa', here it is CCONJ cc.
- 'Kediden mi korkuyorsun yoksa?' adverbial use, 'yoksa' is ADV advmod.

Check and correct accordingly.

```
4 yoksa ADV advmod
2 yoksa CCONJ mark
1 yoksa CCONJ cc
1 yoksa ADV cc
```

```
72. REGEX: cut -f 2,3,4,6 */*/wcs.annot.dep.conllu | grep -P '~ein.*\t' \
| grep NUM | sort | uniq -c | sort -n -r
```

CHECK: Check if they are really 'NUM', not 'DET'. If it is 'NUM', remove all features except 'NumType=Card'.

```
4 ein ein NUM NumType=Card
2 eins ein NUM NumType=Card
2 eine ein NUM Case=Nom|Gender=Fem|Number=Sing|NumType=Card
1 eins eins NUM NumType=Card
1 eins ein NUM Case=Acc|NumType=Card
1 einen ein NUM Case=Acc|Gender=Masc|Number=Sing|NumType=Card
1 ein ein NUM Case=Dat|Gender=Masc|Number=Sing|NumType=Card
1 eine ein NUM Case=Acc|Gender=Fem|Number=Sing|NumType=Card
```

Fig. 4 Examples from the annotation correction work cycle. Regular expressions help identify possible mistakes. The explanations also serve as our own guidelines in addition to the UD guidelines. The annotators are asked to check the given cases and correct them if necessary. Questions that arise during the process and any mistakes that are encountered contribute to the next set of correction items

compound:redup in the Turkish UD treebanks; we apply it also to German in this case.

(22)

‘Let me show you also how potion et cetera is made.’

6.2 Issues related to spoken language

We also observe some linguistic phenomena more frequently than corresponding monolingual treebanks due to the medium we collect the data. Spoken language contains many disfluencies, repetitions, run-on sentences, and uncommon word order. Since these phenomena are orthogonal to mixing languages, their dependencies can cross language boundaries within a sentence. We exemplify two of the commonly observed cases.

6.2.1 Appositions

In appositions, two consecutive noun phrases define the same referent in different ways. In our corpus, these two noun phrases could as well be in different languages. In (23) the speaker mentions ‘someone from Berlin’ in German then refers to the same person with additional information ‘an acquaintance of my mother’ in Turkish. Following the UD guidelines, the head of the second phrase is dependent on the head of the first phrase with the relation *appos*.

Table 2 The symbols used in the glosses

Gloss	Description
1SG	First person singular agreement
2SG	Second person singular agreement
3SG	Third person singular agreement
ABL	Ablative
ACC	Accusative
AOR	Aorist
DAT	Dative
EVID	Evidential
FUT	Future
GEN	Genitive
INTJ	Interjection
LOC	Locative
OPT	Optative
P1SG	First person singular possessive marker
P2SG	Second person singular possessive marker
P3SG	Third person singular possessive marker
PASS	Passive
PASTPART	Past participle
PAST	Past tense
PL	Plural
PRES	Present tense
PRESPART	Present participle
PROG	Progressive
QUES	Question particle
WHILE	Turkish <i>-(i)ken</i> suffix meaning <i>while</i>

The description of the Universal Dependencies tags can be found in (de Marneffe et al. 2021), and their up-to-date official documentation at <https://universaldependencies.org/guidelines.html>. The additional documentation for the SAGT treebank could be found at <https://universaldependencies.org/qtd/index.html>

‘Someone from Berlin, well, an acquaintance of my mother.’

6.2.2 Dislocation

In spoken Turkish it is quite common to replace a word or phrase that does not come to mind immediately or inappropriate to say with the word *şey* ‘thing’. While it is a

noun itself, it can also replace verbs or clauses when combined with the light verb *etmek* ‘do’. The CS corpus we are collecting has many instances of such use; (24) demonstrates one case.

(24)

‘The hard part of it was this thing, Informatics.’

The speaker first uses *şey* as the nominal predicate of the copular sentence. This way the sentence is grammatically complete with the placeholder *şey* until the last word. Once the word *Informatik* ‘Informatics’ is uttered, it does not have a role in the sentence other than clarifying *şey*. UD employs the *dislocated* tag for these relations. By definition, the dislocated item is attached to the head of the placeholder. Here, the head is the placeholder itself; thus, *Informatik* is dependent on *şey*.

6.2.3 Clausal discourse elements

Spoken language contains many clauses with fixed semantics that function as discourse markers such as *you know*, *say*, *I think*. We observe similar cases in our corpus too; most frequent examples include German *weißt du* ‘you know’, *ich glaube* ‘I think’, and Turkish *bak* ‘look’. The UD policy for such cases is connecting them to the main clause with a *parataxis* tag. Some of the UD spoken treebanks (Dobrovolicj and Nivre 2016; Gerdes and Kahane 2017; Courtin et al. 2018) keep the discourse information via the subtype *parataxis:discourse*. We follow their approach and employ the same tag as exemplified in (25) with *weißt du* ‘you know’.

(25)

‘It is not food, you know, just water.’

7 Quality assurance

Manual annotation is an error-prone task. The sources of errors include unclear annotation guidelines, trivial annotator errors (e.g., choosing a wrong, orthographically similar label), and inherent subjectivity of some decisions (e.g., whether a word to be segmented is ‘lexicalised’ or not). The probability of erroneous annotations increases for complex annotation tasks with multiple layers requiring specific expertise. Detecting and correcting errors, and quality assessment of the resulting resource are important aspects of a linguistic annotation project.

The annotation project described in this paper is a rather complex project including multiple layers of annotation (from tokenisation/normalisation to syntax), requiring linguistic knowledge as well as native or near-native proficiency in multiple languages. The fact that we are annotating a spoken, non-standard form of language, and unclear, incomplete specifications of the annotation schemes that we built on also increases the difficulty of the annotation process. As a result, the prevention, detection and correction of errors is a crucial step. In this section, we first present and discuss the annotator agreement calculated on a part of the treebank, and further steps we took for detecting and correcting errors.

7.1 Annotator agreement

Inter-annotator agreement (IAA) is one of the common ways to measure the quality of the annotations. We report agreement on dependency annotations from a 114-sentence subset of the treebank which was annotated by two annotators. Following earlier literature (Berzak et al. 2016; Liu et al. 2018; Bhat et al. 2018), we report unlabelled attachment agreement (UAA), where annotators agree on the head attachment, labelled attachment agreement (LAA), where annotators agree on the head attachment and its label, and label agreement (LA). Since it is difficult to determine the probability of chance agreement for dependencies, there is no agreed method to calculate chance-corrected IAA scores on UAA and LAA. As a result, we report only raw scores for all metrics, as well as Cohen’s kappa (Cohen 1960) for label agreement. Table 1 shows the agreement scores.

CS treebanking is recent and reported IAA scores vary. On one hand, Seddah et al. (2020) do not give actual scores but imply they are low as a result of task difficulty and language variability among annotators. On the other hand Bhat et al. (2018) give agreements over 90%; perhaps having annotators with 10 years of experience helped achieve these high scores. Our scores are in the same range as that of Braggaa and van der Goot (2021). We attribute these medium-level scores to the tasks difficulty as well as to calculating them between a newly trained annotator and an experienced annotator after guideline changes due to new versions of UD treebanks. The detailed analysis of IAA was used for improving the annotation guidelines and resolving common misunderstandings or mistakes. Although we did not calculate agreement scores for after this round, we report an analysis of corrections made as an indication of improvements after this step in the following subsection.

7.2 Validation methods

Although the IAA is a useful indication of annotation quality, even a perfect IAA does not guarantee an error-free treebank. As a result, we employed a number of error detection and correction iterations. In this section we briefly discuss the methods we used.

A first step in finding potential errors is the validation script from the Universal Dependencies project,²⁸ which checks the well-formedness of the resulting CoNLL-U files, as well as violations of UD ‘universal’ guidelines, such as allowed POS, feature and dependency labels, head-initialness of coordination or modification of certain word classes. Since some constraints on allowed tags are language-specific, we used a modified version of the script that allows labels from both languages. As an additional way to find mistakes, we used the MarkBugs feature of UDApi.

Although useful, these scripts do not catch errors that do not result in violation of the general guidelines. A stream of work on error detection on treebanks is based on detecting varied annotation of the same n-grams in different sentences in a treebank (Boyd et al. 2008). Our experiments with freely available software²⁹ did not yield accurate detection of errors, presumably due to the multilingual nature of our treebank and/or the smaller size in comparison to the treebanks used in evaluating this method. We found an iterative approach where we semi-automatically checked for errors against our own annotation guidelines and wrote either regular expressions on CoNLL-U files, or more complex rules that make use of hierarchical relations. Figure 4 in Appendix 1 presents a snapshot of the actual items we had during the correction process. Our own checks include correct case markers for argument types, e.g., *Acc* for objects in German, *Acc* or *Nom* for objects in Turkish; use of clausal dependency labels (*csubj*, *ccomp*) for clausal arguments; and common mistakes of POS or dependency tags for certain lemmas.

The set of sentences annotated for the IAA calculations has 2106 tokens. The corrections on them resulted in 239 dependency changes (11.34%) of which 162 involves changing the head, 144 involves changing the dependency label (67 cases involving changing both) from the time IAA was calculated to the final version. Most common corrections include confusion between labels *advmod* and *discourse*, marking copulas as root mistakenly, and confusion between *case* and *mark*. Most of these are common confusions due to the UD annotation scheme. The number of mistakes is reduced as annotators gained more experience, and our annotation guidelines were improved with more examples of common confusions.

8 Conclusion

In this paper we present our experience with creating the SAGT Turkish–German code-switching treebank. The data source is conversation collections of bilingual Turkish–German speakers living in Germany. The recorded conversations are first

²⁸ <https://github.com/UniversalDependencies/tools/blob/master/validate.py>.

²⁹ <https://github.com/adrianeboyd/decca>.

annotated with sentence and code-switching boundaries (hence language IDs). Sentences that include intrasentential and intra-word CS are transcribed following common conventions for speech corpora. A parallel transcription layer is added to also provide a normalised version following standard orthography of both languages. The normalised transcriptions are then annotated with lemmas, POS tags, morphological features, and dependency relations.

In annotations, we follow the general UD guidelines, and earlier Turkish and German UD treebanks. When we encounter new monolingual or bilingual syntactic constructions we apply existing relations to these new conditions; if not sufficient, we introduce a subtype. Due to annotating spoken data, our sentences contain dependencies that are rare or nonexistent in monolingual Turkish and German treebanks. For those cases also, we follow general UD guidelines and other spoken UD treebanks.

Our observations suggest that annotating a code-switching treebank comes together with several interesting phenomena and their challenges. For instance, in seemingly simple case of annotating titles, there is a conflict among general guidelines, Turkish treebanks, and German treebanks. Choosing any one of the options has its own advantages or disadvantages. As another example, double case marking is an issue that has been observed first in our treebank and so far does not have an ideal solution within the UD framework. Nevertheless, UD is an evolving framework that seeks solutions to challenges. The diverse set of languages increasing at every release and the active community help achieve this goal. As the UD itself evolves and we interact with the community, our treebank is bound to improve. While the maintenance and updates of the treebank continues, our main focus from this point on is to use the treebank as a resource for computational studies.³⁰

The treebank is publicly available since the UD 2.7 data release. At the time of the writing, the latest published version is UD 2.8.³¹ The audio files of the transcriptions are also available for research purposes via a licence agreement as audio recordings are considered personal data and subject to EU data privacy regulations.

Acknowledgements We are grateful to the reviewers for their valuable comments. The treebank development is funded by DFG via project CE 326/1-1 “Computational Structural Analysis of German-Turkish Code-Switching (SAGT)”. We thank Cansu Turgut, Reha Sakızlı, Semanur Ceylan, and Sevd Ceylan for data collection and annotation. We also thank Caroline Swicegood for proofreading the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended

³⁰ In that vein, we have published our first paper on morphological analysis (Özateş and Çetinoğlu 2021).

³¹ https://github.com/UniversalDependencies/UD_Turkish_German-SAGT/tree/master.

use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix 1

The Praat annotation and CoNLL-U representation of an example sentence are given in Figs. 5 and 6 respectively.

Fig. 5 The Praat annotation of the sentence *In dem dritten Semesterda Java gelecek* ‘Java will start in the third semester’. The six annotation tiers are described in Sect. 3.2

1	In	in	ADP	_	4	case	LangID=DE
2	dem	der	DET	Case=Dat Definite=Def Gender=Neut Number=Sing PronType=Art	4	det	LangID=DE
3	dritten	dritte	NUM	NumType=Card	4	nummod	LangID=DE
4	Semesterda	Semester	NOUN	Case=Loc Number=Sing	6	obl	CSPoint=Semesterşda DeCase=Dat DeGender=Neut LangID=MIXED
5	Java	Java	PROPN	Case=Nom Gender=Neut Number=Sing	6	nsubj	LangID=DE
6	gelecek	gel	VERB	Aspect=Perf Evident=Fh Mood=Ind Number=Sing Person=3 Tense=Fut	0	root	LangID=TR SpaceAfter=No
7	.	.	PUNCT	_	6	punct	LangID=OTHER

Fig. 6 The CoNLL-U representation of the sentence *In dem dritten Semesterda Java gelecek* ‘Java will start in the third semester’. The annotated version of the sentence is given in (17). Language-specific POS tag (XPOS) and Enhanced dependency (DEPS) columns that are not annotated in the CS treebank were not included in this example

References

- Auer, P., & Wei, L. (2007). *Handbook of multilingualism and multilingual communication* (Vol. 5). Berlin: Walter de Gruyter.
- Backus, A. D. (2009). Codeswitching as one piece of the puzzle of language change: the case of Turkish Yapmak. In L. Isurin, D. Winford, & K. de Bot (Eds.), *Interdisciplinary approaches to codeswitching, number 41 in Studies in Bilingualism* (pp. 307–336). Amsterdam: John Benjamins.
- Belazi, H. M., Rubin, E. J., & Toribio, A. J. (1994). Code switching and X-bar theory: The functional head constraint. *Linguistic Inquiry*, 221–237.
- Berzak, Y., Huang, Y., Barbu, A., Korhonen, A., & Katz, B. (2016). Anchoring and agreement in syntactic annotations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (pp. 2215–2224). Association for Computational Linguistics. <https://doi.org/10.18653/v1/D16-1239>.
- Bhat, I., Bhat, R. A., Shrivastava, M., & Sharma, D. (2018). Universal Dependency parsing for Hindi-English code-switching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Long Papers)* (Vol. 1, pp. 987–998). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N18-1090>.
- Boyd, A., Dickinson, M., & Detmar Meurers, W. (2008). On detecting errors in dependency treebanks. *Research on Language and Computation*, 6(2), 113–137.
- Braggaar, A., & van der Goot, R. (2021). Challenges in annotating and parsing spoken, code-switched, Frisian-Dutch data. In *Proceedings of the Second Workshop on Domain Adaptation for NLP* (pp. 50–58). Association for Computational Linguistics.
- Çetinoğlu, Ö. (2016). A Turkish-German code-switching corpus. In *The 10th International Conference on Language Resources and Evaluation (LREC-16)*.
- Çetinoğlu, Ö. & Çöltekin, Ç. (2016). Part of speech annotation of a Turkish-German code-switching corpus. In *The 10th Linguistic Annotation Workshop (LAW-X)*.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20, 37–46.
- Çöltekin, Ç. (2010). A freely available morphological analyzer for Turkish. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)* (pp. 820–827).
- Çöltekin, Ç. (2015). A grammar-book treebank of Turkish. In Dickinson, M., Hinrichs, E., Patejuk, A., & Przepiórkowski, A. (Eds.), *Proceedings of the 14th workshop on Treebanks and Linguistic Theories (TLT 14)* (pp. 35–49).
- Çöltekin, Ç. (2016). (when) do we need inflectional groups? In *Proceedings of The First International Conference on Turkic Computational Linguistics*.

- Courtin, M., Caron, B., Gerdes, K., & Kahane, S. (2018). Establishing a language by annotating a corpus: the case of Najja, a post-creole spoken in Nigeria. In *Proceedings of the Workshop on Annotation in Digital Humanities* (pp. 7–11).
- de Marneffe, M.-C., Manning, C. D., Nivre, J., & Zeman, D. (2021). Universal dependencies. *Computational Linguistics*, 47(2), 255–308.
- Dobrovoljc, K., & Nivre, J. (2016). The Universal Dependencies treebank of spoken Slovenian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.
- Edwards, M., & Gardner-Chloros, P. (2007). Compound verbs in codeswitching: Bilinguals making do? *International Journal of Bilingualism*, 11(1), 73–91. <https://doi.org/10.1177/13670069070110010501>.
- Eisenberg, P. (2013). *Grundriss der deutschen Grammatik: Band 2: Der Satz*. New York: Springer.
- Eppler, E. M. (2005). *The syntax of German–English code-switching*. PhD thesis, University of London.
- Foth, K. A., Köhn, A., Beuck, N., & Menzel, W. (2014). Because size does matter: The Hamburg dependency treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (pp. 2326–2333). European Language Resources Association (ELRA).
- Gerdes, K., & Kahane, S. (2016). Dependency annotation choices: assessing theoretical and practical issues of Universal Dependencies. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pp. 131–140. Association for Computational Linguistics. <https://doi.org/10.18653/v1/W16-1715>.
- Gerdes, K., & Kahane, S. (2017). Trois schémas d'annotation syntaxique en dépendance pour un même corpus de français oral : le cas de la macrosyntaxe. In *Atelier sur les corpus annotés du français (ACor4French)*.
- Göksel, A., & Kerslake, C. (2005). *Turkish: A comprehensive grammar*. London: Routledge.
- Hennig, F., & Arne, K. (2017). Dependency tree transformation with tree transducers. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)* (pp. 58–66). Association for Computational Linguistics.
- Hudson, R. A. (1990). *English Word Grammar*. Oxford: Blackwell.
- Imo, W. (2016). *Grammatik*. Metzler: Eine Einführung. J.B.
- Joshi, A. K. (1982). Processing of sentences with intra-sentential code-switching. In *Proceedings of the 9th Conference on Computational Linguistics* (Vol. 1, pp. 145–150). Academia Praha.
- Kornfilt, J. (1997). *Turkish*. London: Routledge.
- Leung, H., Poiret, R., Wong, T.-S., Chen, X., Gerdes, K., & Lee, J. (2016). Developing universal dependencies for Mandarin Chinese. In *Proceedings of the 12th Workshop on Asian Language Resources (ALR12)* (pp. 20–29).
- Liu, Y., Zhu, Y., Che, W., Qin, B., Schneider, N., & Smith, N. A. (2018). Parsing tweets into Universal Dependencies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Long Papers)* (Vol. 1, pp. 965–975). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N18-1088>.
- Lynn, T., & Foster, J. (2016). Universal dependencies for Irish. In *Celtic Language Technology Workshop* (pp. 79–92).
- Lyu, D.-C., Tan, T.-P., Chng, E.-S., & Li, H. (2015). Mandarin-English code-switching speech corpus in South-East Asia: SEAME. *LRE*, 49(3), 581–600.
- McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Castelló, N. B., & Lee, J. (2013). Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Short Papers)* (Vol. 2, pp. 92–97). Association for Computational Linguistics.
- Medlock, B. (2006). An introduction to NLP-based textual anonymisation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. European Language Resources Association (ELRA).
- Myers-Scotton, C. (1993). *Duelling languages: Grammatical structure in codeswitching*. Oxford: Oxford University Press.
- Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., & Zeman, D. (2016). Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 1659–1666).

- Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F., & Zeman, D. (2020). Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 4034–4043). European Language Resources Association.
- Offazer, K., Say, B., Hakkani-Tür, D. Z., & Tür, G. (2003). Building a Turkish treebank. In Abeillé, A. (Ed.) *Treebanks: Building and Using Parsed Corpora*, chapter 15, (pp. 261–277). Springer.
- Øvrelid, L., & Hohle, P. (2016). Universal Dependencies for Norwegian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.
- Özateş, Ş. B., & Çetinoğlu, Ö. (2021). A language-aware approach to code-switched morphological tagging. In *Proceedings of the Workshop on Computational Approaches to Linguistic Code-Switching*.
- Partanen, N., Blokland, R. Lim, K., Poibeau, T., & Riebler, M. (2018). The first Komi-Zyrian Universal Dependencies treebanks. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)* (pp. 126–132). Association for Computational Linguistics.
- Popel, M., Žabokrtský, Z., & Vojtek, M. (2017). Udaapi: Universal API for Universal dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)* (pp. 96–101). Association for Computational Linguistics.
- Poplack, S. (1980). Sometimes I'll start a sentence in Spanish y termino en Espanol: Toward a typology of code-switching. *Linguistics*, 18(7–8), 581–618.
- Poplack, S. (2001). Code-switching (linguistic). *International encyclopedia of the social and behavioral sciences* (pp. 2062–2065).
- Pratap, A., Choudhury, M., & Sitaram, S. (2018). Word embeddings for code-mixed language processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3067–3072). Association for Computational Linguistics.
- Rehbein, I., Ruppenhofer, J., & Do, B.-N. (2019). tweeDe—a Universal Dependencies treebank for German tweets. In *Proceedings of the 17th Workshop on Treebanks and Linguistic Theories, TLT 2019*.
- Rehbein, I., Schalowski, S., & Wiese, H. (2014). The KiezDeutsch korpus (KiDKo) release 1.0. In *The 9th International Conference on Language Resources and Evaluation (LREC-14)*.
- Seddah, D., Essaidi, F., Fethi, A., Futeral, M., Muller, B., Suárez, P. J. O., Sagot, B., & Srivastava, A. (2020). Building a user-generated content North-African Arabizi treebank: Tackling hell. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 1139–1150). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.107>.
- Sulubacak, U., Eryiğit, G., Pamay, T. et al. (2016a). IMST: A revisited Turkish dependency treebank. In *Proceedings of TurCLing 2016, the 1st International Conference on Turkic Computational Linguistics*. EGE University Press.
- Sulubacak, U., Gokirmak, M., Tyers, F., Çöltekin, Ç., Nivre, J., & Eryiğit, G. (2016b). Universal dependencies for Turkish. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 3444–3454).
- Toribio, A. J., & Bullock, B. E. (2012). *The Cambridge handbook of linguistic code-switching*. Cambridge: Cambridge University Press.
- Türk, U., Atmaca, F., Özateş, Ş. B., Berk, G., Bedir, S. T., Köksal, A., Başaran, B. Ö., Güngör, T., & Özgür, A. (2020). Resources for Turkish dependency parsing: Introducing the BOUN treebank and the BoAT annotation tool.
- Tyers, F., Washington, J., Çöltekin, Ç., & Makazhanov, A. (2017). An assessment of Universal Dependency annotation guidelines for Turkic languages. In *5th International Conference on Turkic Language Processing (TURKLANG 2017)* (pp. 356–377).
- Underhill, R. (1976). *Turkish grammar* (Vol. 460). Cambridge: MIT press.
- Wang, L., & Liu, H. (2013). Syntactic variations in Chinese-English code-switching. *Lingua*, 123, 58–73.
- Zeman, D., Popel, M., Straka, M., Hajič, J., Nivre, J., Ginter, F., Luotolahti, J., Pyysalo, S., Petrov, S., Potthast, M., Tyers, F., Badmaeva, E., Gokirmak, M., Nedoluzhko, A., Cinková, S., Hajič jr., J., Hlaváčová, J., Kettnerová, V., & Urešová, Z. et al. (2017). CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (pp. 1–19). Association for Computational Linguistics. <https://doi.org/10.18653/v1/K17-3001>.

Zhang, Y., Riesa, J., Gillick, D., Bakalov, A., Baldrige, J., & Weiss, D. (2018). A fast, compact, accurate model for language identification of codemixed text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 328–337). Association for Computational Linguistics.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.