Enhancements to the BOUN Treebank Reflecting the Agglutinative Nature of Turkish

Büşra Marşan, Salih Furkan Akkurt, Muhammet Şen, Merve Gürbüz, Onur Güngör, Şaziye Betül Özateş, Suzan Üsküdarlı, Arzucan Özgür, Tunga Güngör, Balkız Öztürk

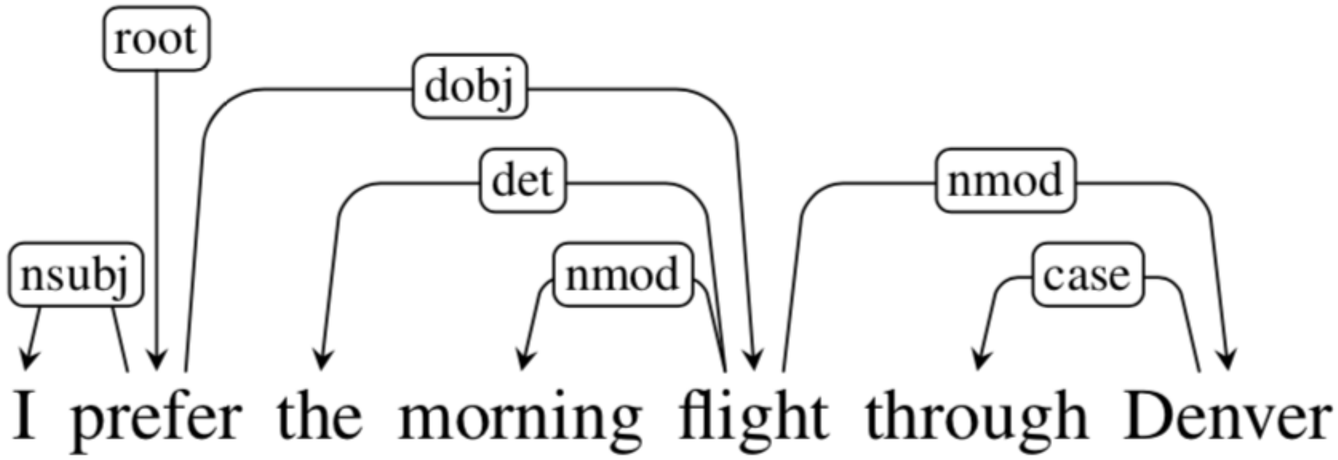
ALTNLP 2022

Contents

- A Brief Overview of Dependency Parsing
- Dependency Treebanks in Turkish
- Our Goal
- Re-annotation Process
- The BOAT Tool
- Parser Performance
- Concluding Remarks

A Brief Overview of Dependency Parsing

- Based on Dependency Grammar, introduced by Tesniére.
- Dependency relations: How head(s) and dependent(s) relate to one another
- Framework we abide by: Universal Dependencies



Dependency Treebanks in Turkish

- First attempts: IMST-UD, The Grammar Book Treebank, IWT UD
- Treebanks available at the UD repository: Tourism, ATIS, KeNet UD, Penn Treebank, FrameNet Treebank, and **BOUN Treebank**.

Dependency Treebanks in Turkish

- **BOUN Treebank** contains **9,761 sentences** and **121,214 tokens** randomly selected from **Turkish National Corpus (TNC)**.

  It covers **five** different registers: Broadsheet national newspapers, biographical texts, essays, popular culture articles, and instructional texts

Our Goal: Improving the BOUN Treebank

- Main challenges posed by Turkish: Derivation, syncretism, and null morphemes.
- Our goal was **improving the linguistic accuracy** and finding ways to **represent such phenomena** without diverging from the UD framework.
- Better linguistic accuracy is linked to **better parser and morphological analyser performance**. In addition, increases the usability of the treebank in **linguistic research**.

Re-annotation Process

- Conducted by **two linguists** who are native speakers of Turkish.
- Inter-annotator agreement:
  - 100 sentences were double-annotated.
  - LAS and UAS were calculated using **Cohen's Kappa measure.**
  - **LAS = 97.81** and **UAS: 98.61**

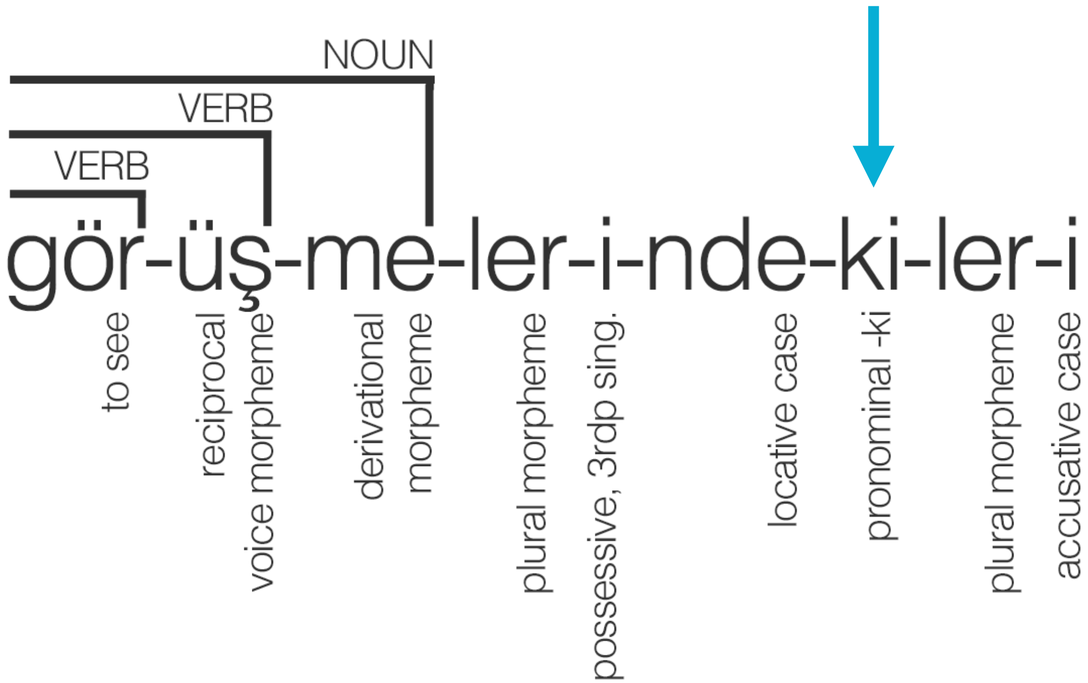
Re-annotation Process: Overcoming the Challenges

- Main challenges were **derivation** and **null morphemes**.
- UD mostly **disregards** derivation.
- There is **no agreed-upon strategy** for null morphemes.

Re-annotation Process: Overcoming the Challenges

Derivation

- UD argument: Final derivational suffix is opaque.
- This causes **loss of information**.



Re-annotation Process: Overcoming the Challenges

Derivation

- UD argument: Final derivational suffix is opaque.
- This causes **misrepresentations on a syntactic level**.

(1) *[[[kahverengi] tüy-lü] kedi]*
brown fur-ATTR cat
"a cat with brown fur"

(2) *[[kahverengi tüy]-lü kedi]*
brown fur-ATTR cat
"a cat with brown fur"

Re-annotation Process: Overcoming the Challenges

Derivation

Solution(s):

- Utilising the MISC tab
    - df= function is introduced for derivational processes triggered by a set of morphemes including -ll and -slz.
- Splitting lemmas
    - Lemmas containing -ki morpheme are splitted.

Re-annotation Process: Overcoming the Challenges

Null Morphemes

- Numerous languages including Turkish, Russian, Coptic, Marathi, and Arabic have null morphemes yet **there is no official way** within the UD framework to show these morphemes.
- Each language follows **a different strategy**.

Re-annotation Process: Overcoming the Challenges

Null Morphemes

- Turkish copula has three surface forms: i-, -y-, and ∅. For the annotation of the null morpheme, two functions for the MISC tab were introduced:
  - nullcop=3s for singular
  - nullcop=3p for plural

Re-annotation Process: Overcoming the Challenges

Copula

- “ol” copula has **six** functions:
  Intransitive verb,
  Transitive verb,
  Auxiliary verb in embedded sentences,
  Auxiliary verb following the participle,
  Light verb forming complex verbal constructions,
  Existential predicate.

- New annotation schema **distinguishes** these different uses through **new XPOS tags** and **dependency relations**.

Re-annotation Process: Changes

Lemma	var	yok	ol- (after participle)	ol- (in embedded sentences)	ol- (in light verb constructions)	ol- (as transitive or intransitive verb)	-ki (adjectivizer)	-ki (pronominal)
UPOS	NOUN	NOUN	AUX	AUX	VERB	VERB	PART	PRON
XPOS	Exist	Exist		Ptcp			Attr	Partic
Deprel	root	root	aux	cop	compound:lvc	root	dep:der	

Re-annotation Process: Statistics

- **117,732 changes** were made in the following tabs: UPOS, XPOS, Deprel, MISC, and Features.

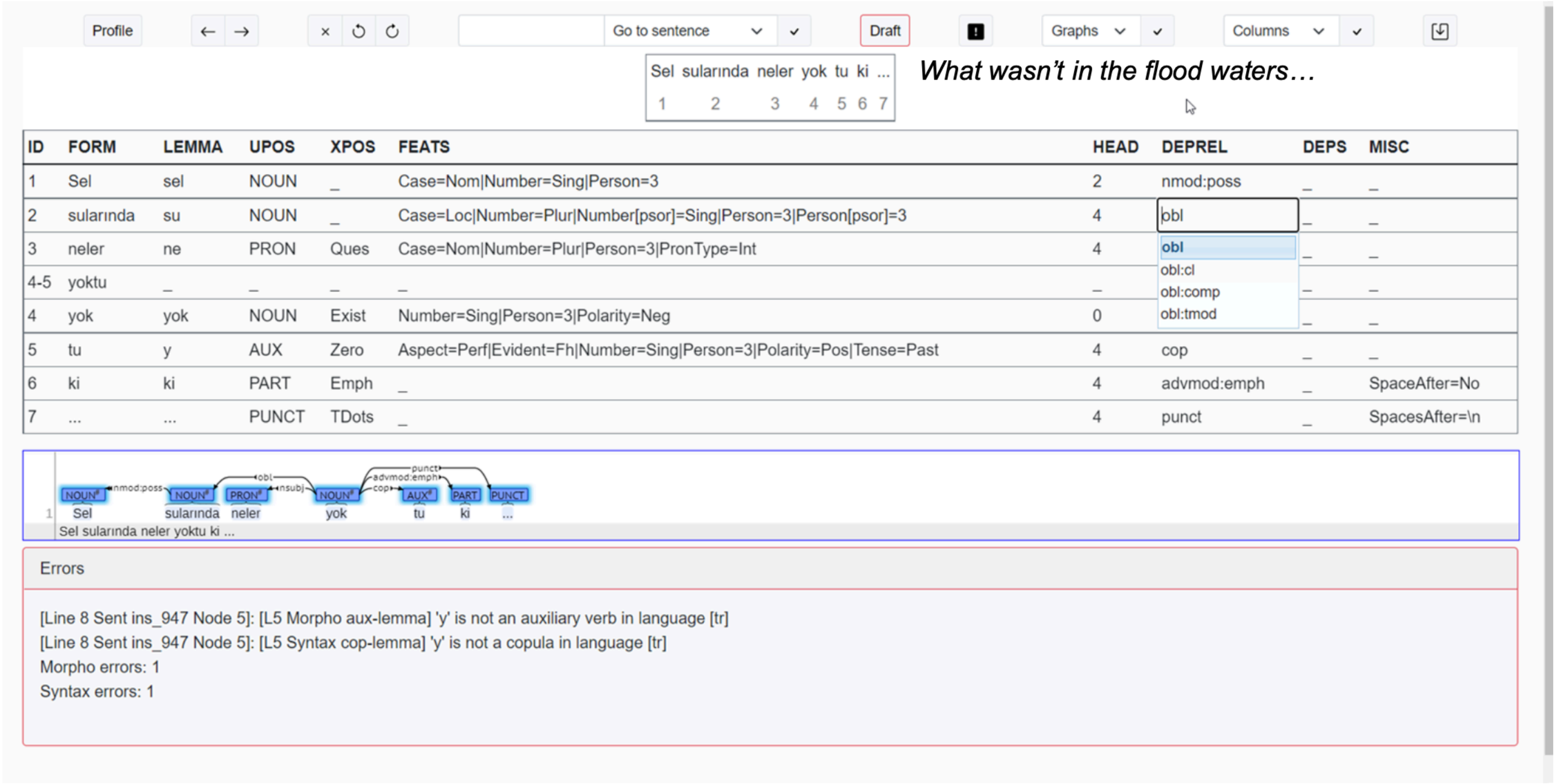
Field	UPOS	XPOS	Features	Deprel	MISC
Changes	11,396	63,829	27,098	23,32	4,973

Re-annotation Process: Statistics

- **117,732 changes** were made in the following tabs: UPOS, XPOS, Deprel, MISC, and Features.

Field	UPOS			XPOS		
Change	**Adj ->Noun**	**CConj ->Part**	**Noun ->Propn**	**Verb ->Ptcp**	**Verb ->Vnoun**	**ANum ->Indef**
Count	1,595	1,025	968	2,459	1,664	1,622

The BOAT Tool



ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS	MISC
1	Sel	sel	NOUN	_	Case=Nom|Number=Sing|Person=3	2	nmod:poss	_	_
2	sularında	su	NOUN	_	Case=Loc|Number=Plur|Number[psor]=Sing|Person=3|Person[psor]=3	4	obl	_	_
3	neler	ne	PRON	Ques	Case=Nom|Number=Plur|Person=3|PronType=Int	4		_	_
4-5	yoktu	_	_	_	_	_		_	_
4	yok	yok	NOUN	Exist	Number=Sing|Person=3|Polarity=Neg	0		_	_
5	tu	y	AUX	Zero	Aspect=Perf|Evident=Fh|Number=Sing|Person=3|Polarity=Pos|Tense=Past	4	cop	_	_
6	ki	ki	PART	Emph	_	4	advmod:emph	_	SpaceAfter=No
7	...	...	PUNCT	TDots	_	4	punct	_	SpacesAfter=\n

Parser Performance

- BiLSTM-based biaffine dependency parser proposed by Dozat and Manning was trained using the updated BOUN Treebank.

	Train	Development	Test	Entire Data
Average Arc Length	2.91	2.88	2.82	2.90
Average Token Count	12.83	12.42	12.36	12.74
Number of Sentences	7,803	982	979	9,761

Parser Performance

- **Previous version** of the BOUN Treebank:
  LAS = 70.37
  UAS = 77.36

- **Re-annotated version** of the BOUN Treebank:
  LAS = 70.26
  UAS = 77.96

Parser Performance

- Newly introduced **dependency relation labels:**
  dep:der *(1,032 instances),*
  obl:tmod *(894 instances),*
  advmod:emph *(1,860 instances),*
  compound:lev *(1,545 instances).*

- **Added complexity:**
  Newly introduced classes,
  New uses of existing tags.

- Refinements on a **morphological level**.
  BiLSTM-based biaffine dependency parser **ignores morphology**.

Concluding Remarks

- In order to offer a universal annotation style, UD tends to overlook certain particularities of languages. We aimed to **overcome this issue without significantly diverging from the framework**.
- A morphology-aware parser or a morphological analysis based downstream task can be conducted using the refined dataset.