A Universal Dependencies Treebank for Marathi

Vinit Ravishankar
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University in Prague
vinit.ravishankar@gmail.com

Abstract

This paper describes the creation of a free and open-source dependency treebank for Marathi, the first open-source treebank for Marathi following the Universal Dependencies (UD) syntactic annotation scheme. In the paper, we describe some of the syntactic and morphological phenomena in the language that required special analysis, and how they fit into the UD guidelines. We also evaluate the parsing results for three popular dependency parsers on our treebank.

1 Introduction

The Universal Dependencies (UD) project (Nivre et al., 2016) is a recent effort to attempt to arrive at 'universal' annotation standards for dependency treebanks. These annotation standards also cover POS tags and morphology, in addition to the expected dependency relations. In recent years, the UD project has been growing more popular; the CoNLL 2017 shared task on dependency parsing (Zeman et al., 2017) resulted in the development and release of a number of dependency parsing pipelines that parse raw text to UD annotated trees.

UD's treebanks cover a number of languages; however, there are, as with most language resources, several gaps in treebank availability for certain languages or families. In this paper, we describe the creation of a treebank for Marathi, an Indic language spoken primarily in the state of Maharashtra in western India.

In Section 2 of our paper, we briefly describe the grammar and political status of Marathi. Section 3 describes some prior work on Marathi NLP, including work relevant to our treebank. Section 4 describes the creation and size of our corpus. Section 5 describes some of the more interesting linguistic phenomena in Marathi and how they fit into UD guidelines. Section 6 describes our evaluation methodology and our results. We conclude with Section 7, where we discuss future avenues for expansion.

2 Marathi

Marathi is an Indic language spoken by approximately 71 million speakers, most of these in the western Indian state of Maharashtra. It is one of the 22 scheduled languages of the Indian government.[1] Due to Maharashtra's position as the state with the longest border with Dravidian language-speaking states, Marathi has adopted several features typical to the Dravidian language family, beyond those present in the south Asian sprachbund: these include clusivity, reduced relative clause construction, and a range of negative auxiliaries (Junghare, 2009). Marathi is written in the Devanagari script, with a few minor modifications and extra characters. Throughout this paper, we transliterate all examples using the International Alphabet of Sanskrit Transliteration (IAST).

Whilst not the first Indic language with a Universal Dependencies treebank, the existing Hindi and Urdu treebanks are conversions of another annotation schema (Tandon et al., 2016), that can be lossy when converting to UD. The treebank we describe is, therefore, the first (to our knowledge) manually

[1] A 'scheduled' language in this context refers to a language in which Indian public service candidates are entitled to be examined, amongst other obligations on part of the government.

Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT16), pages 190–200,
Prague, Czech Republic, January 23–24, 2018.

Keywords: dependency, UD, universal, Indic, Indo-Aryan, Marathi, annotation

annotated Universal Dependencies treebank release in an Indic language. Our motivation for choosing the UD formalism is twofold: first, we believe that the growing popularity of the framework and related conferences and shared tasks could be beneficial to work on Marathi computational linguistics. Second, the 'universal' nature of the Universal Dependencies project can only be tested by the addition of more language treebanks: the creation of a Marathi treebank, therefore, the creation of this treebank is mutually advantageous to both the project and to the state of Marathi computational linguistics.

Marathi is, compared to other Indic languages, fairly morphologically complex. Nouns tend to adopt the *three-layer* morphology described in Masica (1993): nouns first form an oblique case (often through non-transparent modifications), then take a direct case suffix, then, optionally, a postpositional suffix. Unlike in many other Indic languages, these layers are often orthographically joint in Marathi. Verbs show a wide variety of infinitives and participle forms, which are described in a later section.

Syntactically, Marathi tends to follow SOV alignment, although word order is relatively free. Marathi also shows split ergativity: the perfective aspect induces the ergative—absolutive alignment.

3 Prior work

The AnnCorra project describes a dependency annotation schema for Indian languages, based on a 'Paninian grammatical model' (Bharati et al., 2002). A Marathi treebank annotated under this schema appears to be a work in progress; this was described by Tandon and Sharma (2017), who also describe parsing strategies for Marathi and other underresourced Indian languages, based on this schema.

Whilst Marathi grammars do exist, our primary resource was Masica's pan-Indic descriptive grammar (Masica, 1993). In addition to this, Dhongade and Wali (2009) provide a fairly comprehensive grammar of Marathi; however, there is some disagreement between their grammar and Masica's. Finally, we also used a grammar by Navalkar (1868); despite being considerably dated, the grammar is quite succinct and well-written.

Several tools for Marathi exist, ranging from POS taggers (Singh et al., 2013) to morphological analysers. These tools are sometimes released under non-free licenses, or are otherwise opaque; we used a free and open-source morphological analyser (Ravishankar and Tyers, 2017) written in the Apertium formalism (Forcada et al., 2011), deeming this to be sufficient for POS tagging. All morphological disambiguation was performed manually; if incorrect, they were fixed manually.

4 Corpus

Our corpus primarily consists of stories from Wikisource. The collection of stories available is fairly large; we chose those that resembled modern spoken or written Marathi the most, as there is a significant difference between formal written Marathi, especially in the past, and written forms available today. This is reflected primarily in the use of certain morphological forms that have fallen out of use in modern spoken Marathi,[2] something that we tried to avoid for an initial treebank release. The text in our corpus, therefore, would be considered fairly standard in Pune, if a bit old-fashioned in places.

Whilst we would have liked to include news in our corpus, this was complicated: our attempts to scrape a news corpus stopped rather abruptly on the discovery that the most widely distributed Marathi newspapers were all published online as images or GIFs. A future goal is to convert these newspapers, assuming licenses permit, to text using OCR utilities.

Our final parsed corpus consisted of 3,506 tokens and 486 sentences.

4.1 Preprocessing

We ran our corpus through the Apertium morphological analyser cited above, forcing the output to be in the VISL format (Bick and Didriksen, 2015) rather than Apertium's default format. The main reason for this was that we judged it easier, ergonomically, to annotate in this format: morphological disambiguation simply involved deleting lines with inappropriate analyses, and dependency relations were added to the end of every line (representing a token). These were later converted to the required CoNLL-U format with

[2] Our dialect of reference is urban Marathi spoken primarily in the city of Pune; Marathi is fairly diverse in terms of dialects, which vary by region, caste and social class.

a script; another script converted the POS tags to UD POS tags, and the morphology to UD morphology. This conversion required some minor additional manual editing in areas where UD morphology required more specificity than our Apertium analyser provided. Appendix A has an example of a sentence in the VISL format, and the CoNLL-U equivalent. Around the final quarter of our treebank, we switched to using UD-Annotatrix (Tyers et al., 2018) for annotation, with positive results.

4.2 Word segmentation

An important issue we had to address during our creation of the treebank was that of word segmentation, also referred to here as tokenisation. A major issue we faced was the very fuzzy line between cases and postpositions in Marathi. Whilst it was clear that we would not split nouns and their cases into two tokens (despite being agglutinative and clearly separable in nature), we had problems deciding precisely what suffixes could be classed as case suffixes, and what suffixes would be classed as postpositions. There are several tests for distinguishing between the two: one is, for instance, the ability of the genitive oblique to intervene between nouns and true postpositions, whilst another is the relative morphological freedom of postpositions and their ability to form attributive adjectives. None of these tests, however, is perfect, though we eventually arrived at a closed set of cases, partially by relying on tradition and partially by consulting grammars of other Indic languages to attempt to arrive at some standardisation. Our final closed set of cases included the nominative, accusative, dative, ergative, instrumental, comitative/sociative, locative, ablative, vocative and oblique, with the oblique case being the case to which postpositions attach. We do not attach genitives to their heads: this is for consistency with Hindi, and also to avoid the verbose `[psor]` morphology that UD uses to mark possessives.

5 Annotation

Our annotation of the treebank followed the UD version 2.0 guidelines. Our justification for choosing the UD standards was the *universal* nature of the treebank collection. The inclusion of a UD Marathi treebank would benefit both UD - by adding yet another language that would test the validity of the universality of UD's annotation standards - and Marathi, by not requiring us to come up with our annotation standards and documentation.

In the following subsections, we describe some of the more interesting morphological and syntactic constructions in Marathi, and how we chose to annotate them.

5.1 Subject case

Like many other Indic languages, Marathi displays some variation in the possible cases the semantic agent of a construction can take. Part of this is due to split ergativity; ergative-absolutive alignment is triggered by the perfective aspect, whilst the imperfective follows nominative-accusative alignment.

We decided to consider all semantic agents, irrespective of case, to be the syntactic subject of the construction. This results in three standard subject cases: the nominative, for unmarked subjects in the imperfective aspect, the ergative, for subjects marked with the ergative suffix *-ne*, and the dative, for experiencer predicates.

Whilst justifying the existence of dative subjects in Marathi by UD standards is far from obvious, our decision to do so stems from the ability of the dative subject to fulfill several subjecthood tests, such as adjunct subject control. It should be noted, however, that the dative subject in Marathi does fail *other* subjecthood tests, such as verbal agreement. An example of the dative subject is the simple sentence *rātrabhar tilā jhop ālī nāhī* 'she couldn't sleep at night', glossed in Figure 1a. Note the `aux` relation with the negation 'particle', which is actually a verb: it agrees with the subject.

We decided to use the language specific relation `nsubj:own` to denote certain specific ownership constructs that had no clear parallel in other languages we examined; in these constructs, indicating ownership, a postposition (*-kaḍe*) would combine with the oblique case of the owner. This is similar to the use of the locative (*-DA*) in Turkish, or the adessive (*-llA*) in Finnish. We do not subtype `cop` as this is the standard existential use of the copula. Whilst this relation appears to be suitable for now, we are considering modifying it to `nmod:own` in a future release.

Figure 1b is a simple (truncated) sentence from the treebank that demonstrates this construction well.

(a) 'She couldn't sleep all through the night'

(b) 'Mary has a dog too'

Figure 1: Various non-nominative subject cases

5.2 Object case

Objects in Marathi also tend to adopt a number of cases. Our treebank has objects in four cases - the accusative, dative, genitive[3] and the sociative. Distinguishing between accusative objects and dative objects was interesting, as Marathi displays differential object marking: 'accusative' objects in Marathi can be glossed with an 'accusative' null suffix, or with the dative suffix *-lā* for the same verb argument structure, with the latter implying definiteness.

(1) a. *mī* *pakṣī* *baghto*
I-NOM bird-PL.ACC watch-IMPF.1MSG

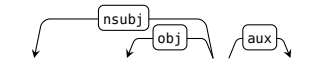
'I watch birds'

b. *mī* *pakṣīṃnā* *baghto*
I-NOM bird-PL.DAT watch-IMPF.1MSG

'I watch (some specific) birds'

Example 1a glosses the object as an accusative due to its non-definiteness, with a null morpheme, whilst Example 1b glosses it as a dative. UD's guidelines specify that a construction with only two verbal arguments should *not* use the indirect object (`iobj`) relation. Taking these things into account, we could do one of two things: either we gloss every noun corresponding to the subcategorisation frame of the governing verb and treat the accusative and dative suffixes as alternative morphological realisations of the same case, or we gloss every noun based on its morphology, thus allowing dative direct objects. We chose the latter.

The inclusion of the sociative (referred to as 'comitative' in UD) case as a direct object was another contentious issue: these objects occurred with verbs that were typically intransitive. The line between treating these arguments as core arguments of a transitive variant of the verb (that warranted the `obj` relation) and between treating them as non-core dependents of the intransitive (warranting an `obl` relation) was a thin one, and we preferred the former analysis in some instances, such as in the (slightly modified) sentence from the treebank in Figure 2: *lok kutryāṃśī bolat hote* 'people were talking to dogs'.



'People were talking with dogs'

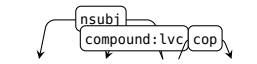
Figure 2: Sociative/comitative objects

We did not encounter examples of indirect objects in any case other than the dative.

[3]Technically the oblique as we split genitives.

5.3 Light verbs

Similar to many other Indic languages and several Indo-Iranian and Turkic languages, Marathi frequently makes use of light verb constructions (LVCs). These are a form of complex verbal predicates, *typically* noun + verb combinations that function as a semantic verb. Most of these constructions involve the verb *karṇe* 'to do/make' as the verbal head of the construction; we used the language specific relation `compound:lvc` to attach dependent nouns. A simple example of light verb constructions from our treebank is the (truncated) sedentence in Figure 3: literally 'the frog was hitting a jump', with 'jump' being the nominal part and 'to hit' being the verbal part of the light verb construction. Despite being non-finite, we chose our verb to be the head of the construction for consistency with other treebanks, particularly the Persian treebank, where LVCs are frequent (Seraji et al., 2016).



beḍūk uḍyā mārat hotā
frog jump hit was
'The frog was jumping.'

Figure 3: Light verb constructions

LVC's display varying degrees of lexicalisation. The LVC *uḍyā mārṇe* 'to hit a jump' is fairly unlexicalised: it can be both qualified with an adjective (*moṭhyā uḍyā mārṇe* 'to hit a large jump'), or modified with an adverb (*jorāt uḍyā mārṇe* 'to forcefully hit a jump'). Other constructs, like *kāljī gheṇe* 'to worry' cannot be qualified; it functions as a fully lexicalised verb. We do not take the degree of lexicalisation into account when assigning this relation.

5.4 Compound verbs

Perhaps one of the more interesting linguistic phenomena that we model in our treebank is the existence of what we refer to as 'compound verbs'. Deoskar (2006) provides an excellent description of compound verbs in Marathi; note, however, that they refer to the phenomenon as 'light verbs', as do other works on the subject (Butt, 2010; Seiss et al., 2009). The reason we use the term 'compound verb' is to prevent confusion with light verbs as described in section 5.3, which are a very distinct syntactic construct. The term 'compound verb' is also not unused in Marathi literature (Pardeshi, 2001).

Compound verbs are, essentially, a combination of two verbs, a *main* verb, very often a converb in Marathi (but a participle or an infinitive in some constructs), and a *secondary* verb, that has no real semantic value, but acts solely to modify the Aktionsart or some minor semantic meaning of the main verb (often, there is no semantic change). The set of secondary verbs is a closed set, and verbs from outside this set function as full, semantically valid verbs.

(2) a. *mī* *goṣṭa* *vāchlī*
I-NOM story.F.SG read-PERF.3FSG

'I read (the) story'

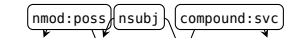
b. *mī* *goṣṭa* *vāchūn* *ṭāklī*
I-NOM story.F.SG read-CONV.PERF put-PERF.3FSG

'I finished off reading (the) story'

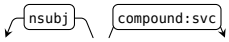
Whilst Example 2b has the same fundamental meaning as the simpler Example 2a, the addition of the vector verb results in a minor semantic shift, indicating finality, or suddenness in completion of the action denoted by the main verb. Whilst it appears that the `aux` relation would be appropriate here, Deoskar (2006) shows that the two classes (vector verbs and auxiliaries) are not the same. We, therefore, subtype another relation and use `compound:svc` to mark this relation, as in the figures 4a[4] and 4b. Despite 'serial

[4]Interestingly, dropping the compound construct would change absolutely nothing about this sentence.

verbs' being a distinct syntactic construct that have very little to do with these sorts of compound verbs, the absence of a dependency relation that better suits this phenomenon compelled us to use `compound:svc` for now.



(a) 'His eyes filled (with tears)'



(b) 'He cried (a lot, without stopping)'

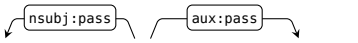
Figure 4: Compound verbs

5.5 Passive voice

Whilst the use of the passive voice is not extremely frequent in Marathi, we did come across several examples in our treebank, which led to the creation of two subtypes that are fairly common in UD: `nsubj:pass` and `aux:pass`. Marathi uses the verb *jāṇe* 'to go' as an auxiliary in the formation of certain passive constructions. The main verb is in the perfective aspect and agrees with the passive subject. An exapmle sentence from our treebank is *rājvāḍā śṛngārlā gelā* 'the palace was decorated', as in Figure 5a.

Another verbal construction common to written Marathi occured quite frequently in our treebank. This is a form of 'formal' passivisation, and uses the the auxiliary verb *yeṇe* 'to come' instead of 'to go'. The main verb, interestingly, is as infinitive in the locative case. The above sentence could be re-written as *rājvāḍā śṛngārnyāt āle* (Figure 5b) without any major change in meaning.

(a) 'The palace was decorated'

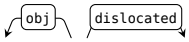


(b) 'The palace came to be decorated'

Figure 5: Two forms of passivisation

5.6 Dislocation

Dislocated pronouns to emphasise nominals or nominal clauses are fairly common in Marathi. These constructions use a demonstrative pronoun along with the clause, similar to dislocation in French. We use the `dislocated` relation to mark these, as in Figure 6.



'Look at this, the horseman'

Figure 6: Dislocation

It is important to note that 'this' in the example does not determine 'horseman', but is a standalone pronoun - fairly visibly, it does not even agree with 'horseman' in gender and number.

6 Evaluation

The pipeline that we primarily use for tokenisation and tagging is the popular UDPipe (Straka and Straková, 2017); it is a trainable pipeline consisting of a tagger, a tokeniser (MorphoDiTa) (Straková et al., 2014) and a parser (Parsito) (Straka et al., 2015). Having tagged and tokenised our text using UDPipe, we evaluate three parsers.

The first of these parsers is Parsito, included in UDPipe itself. It (like many modern parsers) uses a neural network to learn transitions for parsing dependencies. We evaluate UDPipe twice - once using the

	Precision	Recall	F_1 score
Multiwords	99.09	45.31	61.88
Words	94.90	90.18	92.48
Sentences	92.24	92.72	92.44

Table 1: Tokeniser results on raw text.

	UPOS	Feats	All tags	Lemma
Gold standard	78.82	65.99	62.67	74.40
Tokenised	74.11	64.73	61.87	75.37

Table 2: Tagger F_1 scores evaluated with both gold standard and automatic tokenisation.

default settings, and again using external word embeddings trained on the Marathi wiki. We used pre-trained fastText embeddings of dimension 300 (Bojanowski et al., 2016); we believed that these would perform better than embeddings generated by other tools, as fastText also takes into account subword units to build word embeddings, which can have better results for more morphologically complex languages.

The second is the newer BIST parser (Kiperwasser and Goldberg, 2016). Similar to UDPipe, it uses neural networks for parsing: sentences are processed using bidirectional LSTMs. Unlike UDPipe, however, it also offers an implementation that uses a graph-based parsing strategy. Whilst BIST also allows us to use custom word embeddings, we did not do so for infrastructural reasons: using custom embeddings results in exponential model size blowup. We intend to rectify these issues and evaluate BIST with embeddings in the future.

Finally, our third parser is the much older MaltParser (Nivre et al., 2007). Unlike the others, MaltParser does not use a neural network for learning transitions. Given that our treebank is still fairly small, we were interested in comparing the performance of the two approaches: neural networks famously require substantial amounts of data, and despite neural parsers showing clearly better results averaged across all treebanks in competetive evaluations, we wanted to compare their performance on our treebank.

Whilst our primary evaluation is on end-to-end parsing, we also perform a secondary evaluation given gold-standard tokenisation and POS tags. We evaluated both labelled (LAS) and unlabelled (UAS) attachment scores; we also evaluated the *weighted* LAS, which underweights the contribution of correctly labelling certain relations (like `case` and `punct`) to the final score. Evaluation was carried out using the same script that was officially used for the CoNLL 2017 shared task. Each evaluation involved training 10 models for use in 10-fold cross-validation.

BIST parser required some held-out data to be used as a dev set; we used 45 (fixed) sentences for this data, and ran 10-fold CV on the remainder. We ran all parsers with the default parameters, except for BIST parser, where we raised the number of training epochs to 50.

6.1 Results

	Raw text			Gold standard		
	UAS	LAS	(w)LAS	UAS	LAS	(w)LAS
UDPipe	63.00	51.79	46.14	77.74	68.88	64.61
BIST	**67.60**	**54.18**	**47.25**	68.70	55.05	47.99
MaltParser	62.02	49.45	44.01	**80.75**	70.35	65.16
UDPipe[+emb]	59.77	48.20	42.63	79.48	**71.94**	**68.47**

Table 3: Unlabelled, labelled and weighted labelled attachment scores for our parsers, evaluated on a raw text pipeline and on gold-standard tokenisation and POS tags.

Table 1 refers to our tokeniser's results. The poor performance of the tokeniser on multiword tokens

stands out; the relatively high frequency of multiword tokens due to orthographically joined postpositions is likely one of the reasons. Table 2 is the performance of two taggers: one on gold-standard tokenised data, and the other on data tokenised by UDPipe in the previous step.

Finally, we present our dependency parsing results in Table 3.

6.2 Discussion

As expected, our results for gold standard tokenisation and POS tags are significantly better than our results for parsing raw text. What we expected a lot less is the drastic differences in the performance of different parsers, and the performance of different parsers in different situations.

Whilst BIST has the best scores for parsing raw text, this advantage quickly vanishes as it does not improve much in performance on gold standard text at all, and drops to being the worst parser amongst the lot. Interestingly, the results bore out our intuition that MaltParser would be competitive despite its age: whilst not the *best* parser based on the more important LAS anywhere, it does have the best UAS for gold standard tokenisation and POS tags, and is fairly close to the best LAS scores.

Another interesting result worth noting is UDPipe's performance on raw text with word embeddings included; whilst these embeddings intuitively ought to improve (or at least not worsen) results, they do result in a noticeable parsing performance drop on raw text. Gold standard text parses much better, giving us our best LAS scores. We propose that this might occur due to word embeddings trained on external corpora being unable to deal with poorly segmented multiwords: the small size of the treebank does not explain the significant difference between raw text and gold standard POS-tagged text.

7 Future work

Obviously, our most important short-term goal is to increase the size of our treebank, aiming for a release of 10,000 manually parsed tokens. This was the treebank size expected from a surprise language in the CoNLL-2017 shared task. Another short-term goal is to generate data sets for easier evaluation of Marathi word embeddings (Abdou et al., 2018). Apart from this, we have several medium-term goals.

UD have some rudimentary support for language-family specific documentation. As Marathi is the only Indic treebank (that we know of) directly annotated according to UD specifications, we intend to use it as a starting point for writing documentation for Indic languages, contrasting with Marathi wherever possible, and expanding where not. A manual conversion of UD Hindi to fit these standards would be a place to start.

Finally, we also intend to add *enhanced* dependency relations: this has been done for some languages already (Schuster and Manning, 2016), and would be an interesting addition.

Acknowledgements

Work on this paper was partially funded through the stipend provided by the Erasmus Mundus Language and Communication Technologies program. We thank Dan Zeman, Francis M. Tyers and Memduh Gökırmak for their valuable insight on annotation standards. We also thank our anonymous reviewers for their comments.

References

Mostafa Abdou, Artur Kulmizev, and Vinit Ravishankar. 2018. MGAD: Multilingual Generation of Analogy Datasets. In *Proceedings of Language Resources and Evaluation Conference (LREC'18) [to appear]*.

Akshar Bharati, Rajeev Sangal, Vineet Chaitanya, Amba Kulkarni, Dipti Misra Sharma, and KV Ramakrishnamacharyulu. 2002. Anncorra: building tree-banks in Indian languages. In *Proceedings of the 3rd workshop on Asian language resources and international standardization-Volume 12*. Association for Computational Linguistics, pages 1–8.

Eckhard Bick and Tino Didriksen. 2015. Cg-3 – beyond classical constraint grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA*. Linköping University Electronic Press, Linköpings universitet, pages 31–39.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* .

Miriam Butt. 2010. The light verb jungle: still hacking away .

Tejaswini Deoskar. 2006. Marathi light verbs. In *Proceedings from the Annual Meeting of the Chicago Linguistic Society*. Chicago Linguistic Society, volume 42, pages 183–198. http://www.ingentaconnect.com/content/cls/pcls/2006/00000042/00000002/art00012.

R. Dhongade and K. Wali. 2009. *Marathi*. London Oriental and African language library. John Benjamins Publishing Company. https://books.google.co.in/books?id=zVVOvi5C8uIC.

M. L. Forcada, M. Ginestí-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, J. A. Pérez-Ortiz, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F. M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation* 25(2):127–144.

Indira-Y Junghare. 2009. Syntactic convergence: Marathi and Dravidian. *von Kopp, B.: Texts and the Art of Translation. The Contribution of Comparative* 2:163.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and Accurate Dependency Parsing using Bidirectional LSTM Feature Representations. *TACL* 4:313–327. https://transacl.org/ojs/index.php/tacl/article/view/885.

C.P. Masica. 1993. *The Indo-Aryan Languages*. Cambridge Language Surveys. Cambridge University Press. https://books.google.cz/books?id=Itp2twGR6tsC.

G.R. Navalkar. 1868. *The Student's Manual of Marathi Grammar, Etc. [By Gan(a) Pat(i) Ráv(a) Raghunath(a).]*.. https://books.google.cz/books?id=VJYhMwEACAAJ.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Chris Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Dan Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of Language Resources and Evaluation Conference (LREC'16)*.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13(2):95–135.

Prashant Pardeshi. 2001. The Explicator Compound Verb in Marathi: Definitional. *Linguistics* 38:68–85. http://www.lib.kobe-u.ac.jp/repository/80010010.pdf.

Vinit Ravishankar and Francis M Tyers. 2017. Finite-State Morphological Analysis for Marathi. In *Proceedings of the 13th International Conference on Finite State Methods and Natural Language Processing (FSMNLP 2017)*. pages 50–55.

Sebastian Schuster and Christopher D Manning. 2016. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks .

Melanie Seiss, Miriam Butt, and Tracy Holloway King. 2009. On the difference between auxiliaries, serial verbs and light verbs. In *Proceedings of the LFG09 Conference*. CSLI Publications, pages 501–519. http://web.stanford.edu/group/cslipublications/cslipublicationsLFG/14/papers/lfg09seiss.pdf.

Mojgan Seraji, Filip Ginter, and Joakim Nivre. 2016. Universal Dependencies for Persian. In *Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Jyoti Singh, Nisheeth Joshi, and Iti Mathur. 2013. Development of Marathi Part of Speech Tagger using Statistical Approach. *CoRR* abs/1310.0575. http://arxiv.org/abs/1310.0575.

Milan Straka, Jan Hajič, Jana Straková, and Jan Hajič jr. 2015. Parsing Universal Dependency Treebanks using neural networks and search-based oracle. In *Proceedings of Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT 14)*.

Milan Straka and Jana Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, Vancouver, Canada, pages 88–99. http://www.aclweb.org/anthology/K/K17/K17-3009.pdf.

Jana Straková, Milan Straka, and Jan Hajič. 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Baltimore, Maryland, pages 13–18. http://www.aclweb.org/anthology/P/P14/P14-5003.pdf.

Juhi Tandon, Himani Chaudhary, Riyaz Ahmad Bhat, and Dipti Misra Sharma. 2016. Conversion from pāṇinian kārakas to universal dependencies for hindi dependency treebank. *LAW X* page 141.

Juhi Tandon and Dipti Misra Sharma. 2017. Unity in diversity: A unified parsing strategy for major Indian languages. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*. pages 255–265.

Francis M. Tyers, Mariya Shejanova, and Jonathan North Washington. 2018. UD Annotatrix: An annotation tool for universal dependencies. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*. page *this volume*.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 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*. Association for Computational Linguistics, pages 1–19. https://doi.org/10.18653/v1/K17-3001.

A Formats

```
"<">"
	""" qt @punct #1->5
"<माझी>"
	"मी" prn p1 mf sg @nmod:poss #2->4
		"चा" gen f sg @case #3->2
"<जमीन>"
	"जमीन" n f sg nom @obj #4->5
"<विकणार>"
	"विकणे" vblex pros mfn sp @root #5->0
"<नाही>"
	"नाही" vaux neg p1 sg @aux #6->5
"<.>"
	"." sent @punct #7->5
"<">"
	""" qt @punct #8->5
```

Figure 7: An example of the VISL format. The sentence is *mājhī jamīn vikṇār nāhī* ‘I will not sell my land’.

```
# sent_id = 355
# text = "माझी जमीन विकणार नाही."
1   "   "   PUNCT   _   _   5   punct   _   SpaceAfter=No
2-3 माझी    _   _   _   _   _   _   _   _
2   _   मी   PRON    _   Number=Sing|Person=1    4   nmod:poss   _   SpaceAfter=No
3   _   चा   ADP _   Gender=Fem|Number=Sing  2   case    _   _
4   जमीन जमीन NOUN    _   Case=Acc|Gender=Fem|Number=Sing 5   obj _   _
5   विकणार विकणे VERB    _   Aspect=Prosp|VerbForm=Fin   0   root    _   _
6   नाही नाही   AUX _   Number=Sing|Person=1|Polarity=Neg|VerbForm=Fin  5   aux _   SpaceAfter=No
7   .   .   PUNCT   _   _   5   punct   _   SpaceAfter=No
8   "   "   PUNCT   _   _   5   punct   _   _
```

Figure 8: The same sentence in the CoNLL-U format.