ISSN 2347 - 3983
Volume 9. No. 8, August 2021
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter12982021.pdf
https://doi.org/10.30534/ijeter/2021/12982021

Bootstrapping Dependency Treebank of Urdu Noisy Text

Amber Baig[1], Mutee U Rahman[2], Sehrish Abrejo[3], Sirajuddin Qureshi[4], Saima Tunio[5], Shadia S. Baloch[6]
[1]Department of Computer Science, Isra University, Hyderabad, Pakistan, amberbaig@gmail.com
[2]Department of Computer Science, Isra University, Hyderabad, Pakistan, muteeurahman@gmail.com
[3]Department of Computer Science, Isra University, Hyderabad, Pakistan, sehrish-abrejo@hotmail.com
[4]Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China, siraj.qureshi@gmail.com
[5]Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China, saima.tunio@gmail.com
[6]Department of Computer Science, Isra University, Hyderabad, Pakistan, shadiasaadbaloch@gmail.com

ABSTRACT

This paper describes how bootstrapping was used to extend the development of the Urdu Noisy Text dependency treebank. To overcome the bottleneck of manually annotating corpus for a new domain of user-generated text, MaltParser, an opensource, data-driven dependency parser, is used to bootstrap the treebank in semi-automatic manner for corpus annotation after being trained on 500 tweet Urdu Noisy Text Dependency Treebank. Total four bootstrapping iterations were performed. At the end of each iteration, 300 Urdu tweets were automatically tagged, and the performance of parser model was evaluated against the development set. 75 automatically tagged tweets were randomly selected out of pre-tagged 300 tweets for manual correction, which were then added in the training set for parser retraining. Finally, at the end of last iteration, parser performance was evaluated against test set. The final supervised bootstrapping model obtains a LA of 72.1%, UAS of 75.7% and LAS of 64.9%, which is a significant improvement over baseline score of 69.8% LA, 74% UAS, and 62.9% LAS

Key words: bootstrapping, dependency parsing, data-driven approaches, low-resourced languages, treebank, Urdu tweets

1. INTRODUCTION

The necessity of hand-annotated resources, particularly treebanks, is widely recognized in computational linguistics. Treebanking is an important stage in the development of linguistic resources for a language [7], especially for data-driven parsing and many advanced applications like machine translation [17].

Because their creation is costly and time-consuming [5], especially when the domain under consideration is user-generated text found on social media websites such as Twitter, developing these annotations at a reasonable cost is still crucial for low-resource languages [20]. Millions of tweets are posted every day, resulting in noisy and informal content that can be a useful corpus for applications like language technologies, data analysis, sentiment analysis, event detection, and opinion mining [12], [19], [1], to name a few.

For the past two decades, the issue of annotation costs and ways to reduce reliance on annotated corpora remained a recurrent theme in the NLP community [13]. Researchers need reliable strategies for speeding up the annotation process without biassing the gold standard [8]. Bootstrapping is a novel approach that attempts to iteratively generating treebanks in an efficient and cost-effective manner [23]. During bootstrapping process, a trained parser is used to pre-parse raw text, which is then manually corrected by human annotators [9]. To retrain the parser, this corrected data is added to the training set. As a result, the size of the annotated data grows quickly while parser performance steadily improves.

The process of bootstrapping a dependency treebank of Urdu tweets using a data-driven dependency parser is described in this paper. Urdu is a widely spoken language in South Asia, but it is still regarded as a language with limited resources in terms of language technology [18]. Extending NLP tools and resources to social media data can shed light on a variety of scientific questions, including theoretical and contrastive linguistics, linguistic typology, and NLP [11]. The rest of the paper is arranged as follows: The tools and treebank used in this study are described in Section 2. The bootstrapping experiments and their learning setup are described in Section 3, and the results of the experiments are described in Section 4. The error analysis of the parser output is presented in Section 5, and the conclusions are presented in Section 6.

2. MATERIALS AND TOOLS

2.1 Parser

MaltParser [16] is a transition-based data-driven dependency parser, that is used for training a parsing model using a treebank as input and parsing new data by means of this trained model. To induce classifiers from training data, MaltParser uses two different built-in learning libraries: liblinear and libsvm. Liblinear library supports linear classification [24], and libsvm is for Support Vector Machines [23].

Different parsing algorithms are implemented in MaltParser Version 1.9.1. It contains parsing algorithms mainly from three different families such as Nivre, Covington, and Stack, and includes Planar and 2-Planar.

2.2 Evaluation Metrics

MaltEval [14] is used to evaluate parser performance, with Labelled Attachment Score (LAS), Unlabelled Attachment Score (UAS), and Label Accuracy (LA) as evaluation metrics. These three measures are essentially token-level accuracies, which account for all test data tokens and give each token in the evaluation equal weightage.

Equations (1), (2), and (3) show the formula for calculating LAS, UAS, and LA:

$$LAS = \frac{number\ of\ correct\ head\ \&\ dependency\ labels}{total\ tokens} \tag{1}$$

$$UAS = \frac{number\ of\ correct\ head\ labels}{total\ tokens} \tag{2}$$

$$LA = \frac{number\ of\ correct\ labels}{total\ tokens} \tag{3}$$

2.3 UNTDT Treebank

UNTDT (Urdu Noisy Text Dependency Treebank) [4], a manually annotated dependency treebank of 500 Urdu tweets is used as a gold-standard corpus for parser training in this study. The treebank is annotated at morphological and syntactic levels by adopting Universal Dependencies [15] framework to the particularities of social media text. Refer to [4] for full review of the treebank. This treebank was validated using 10-fold cross validation. Best average accuracy score reported by authors was 74% UAS, 62.9% LAS and 69.8% LA. This score is used as a baseline score in present study.

3. BOOTSTRAPPING

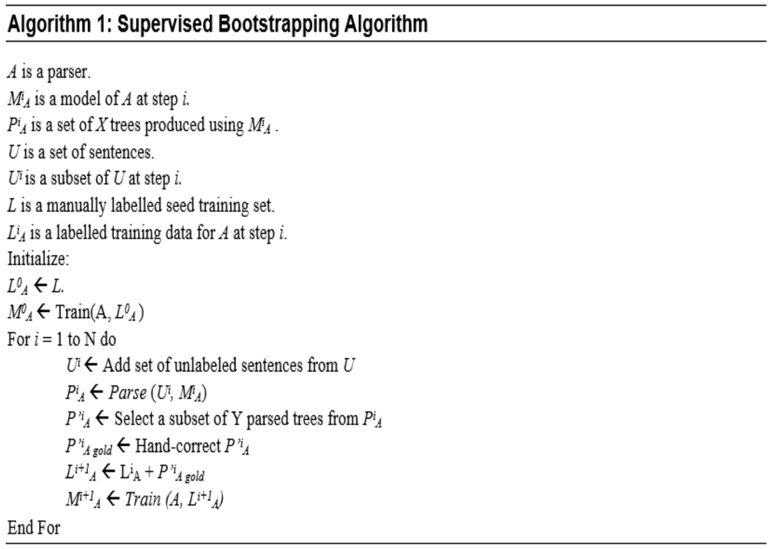
The process of semi-automatically creating annotated training data from large amounts of unannotated data is known as bootstrapping [2]. The sentences are parsed using a pre-trained parser. As a result, instead of annotating from scratch, the annotator(s) can correct the parser output. After that, the newly parsed data is added to the training data, and the process is repeated until all the data has been parsed. In some cases, bootstrapping eliminates the need for manual annotation. The supervised bootstrapping approach, on the other hand, necessitates manual correction to ensure that gold standard trees are added to the Treebank at each iteration.

The following sections describe the entire bootstrapping process:

3.1 Method

For bootstrapping, the method used by [6], [10] and [21] was followed. The pseudocode for this approach is given in Fig 1.

The algorithm requires only a single dependency parser, A. The parser A is first trained on the existing manually labeled data, L to create a model Mi$_A$. The set of gold standard POS-tagged sentences (U) is divided into 4 sets, each set contains 300 sentences U^i. For each of the four iterations (i = 1...4), U^i is parsed to produce P^i_A. A subset Y is then selected from P^i_A for manual correction to produce $P'^i_{A\ gold}$. Every time, manually corrected sentence set ($P'^i_{A\ gold}$) is added in the training set L^i_A for making a large training set of L^{i+1}_A. Induction of a new parsing model (M^{i+1}_A) is performed by training the parser with the new training set. This practice was repeated and as a result, Treebank size grew, and the parsing quality improved gradually.



```
Algorithm 1: Supervised Bootstrapping Algorithm

$A$ is a parser.
$M^i_A$ is a model of $A$ at step $i$.
$P^i_A$ is a set of $X$ trees produced using $M^i_A$ .
$U$ is a set of sentences.
$U^i$ is a subset of $U$ at step $i$.
$L$ is a manually labelled seed training set.
$L^i_A$ is a labelled training data for $A$ at step $i$.
Initialize:
$L^0_A \leftarrow L$.
$M^0_A \leftarrow$ Train(A, $L^0_A$)
For $i$ = 1 to N do
    $U^i \leftarrow$ Add set of unlabeled sentences from $U$
    $P^i_A \leftarrow Parse\ (U^i, M^i_A)$
    $P'^i_A \leftarrow$ Select a subset of Y parsed trees from $P^i_A$
    $P'^i_{A\ gold} \leftarrow$ Hand-correct $P'^i_A$
    $L^{i+1}_A \leftarrow L^i_A + P'^i_{A\ gold}$
    $M^{i+1}_A \leftarrow Train\ (A, L^{i+1}_A)$
End For
```

Figure 1: Supervised Bootstrapping Algorithm [10]

3.2 Experiments

From the pre-processed corpus of raw 4500 Urdu tweets collected by [3], 1200 tweets were randomly selected for this study. A model of of UDPipe [22], which is a neural network based trainable pipeline system was trained on UNTDT for performing tokenization, lemmatization, POS tagging and morphological analysis for Urdu tweets. UDPipe model not only performs above mentioned tasks, but also converts the tweets in CONLL-U format which is form of data required by MaltParser for input. These tweets were than divided into 4 blocks of 300 tweets each to be utilized for bootstrapping the parser.

For experiments, UNTDT is divided in training set of 300 tweets, development set and test set of 100 tweets each. MaltParser is trained on 300 tweets test set using baseline parser model settings. The algorithm shown in Fig V- 1 is repeated for four iterations. At each iteration, 75 (Y) randomly sampled tweets were manually corrected from a set of 300 (X) raw tweets pre-parsed by the parser. The size of the final training set is 600 trees (300 + (4*75)). All the manual corrections were performed by single annotator. At the end of each iteration, the accuracy of parser model is evaluated on the development set and the next batch of 300 tweets is parsed using the newly induced model. After the final iteration, the accuracy of final induced model is tested with the test set. At the end of this experiment, there are 800 (20, 951 tokens) gold standard tweets in the treebank.

4. RESULTS

Figures 2, 3 and 4 show the results of supervised bootstrapping experiments. Fig 2 shows the labelled accuracy, Fig 3 shows unlabeled attachment score and Fig 4 depicts the labeled attachment score over the four training iterations. The highest scoring models occur on the fourth iteration, reaching LA 72.1%, UAS 75.7% and LAS 64.9%. However, steadily improvement of evaluation scores is observed during all the iterations. All results are calculated on the development set.

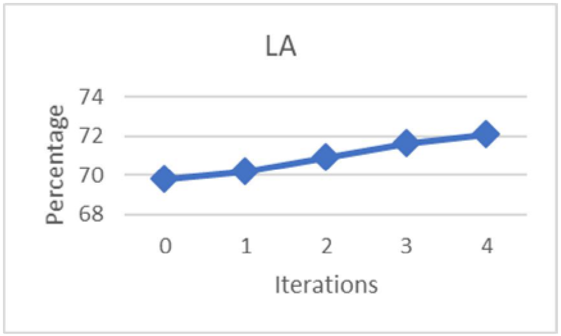


Figure 2: Label Accuracy on Development Set

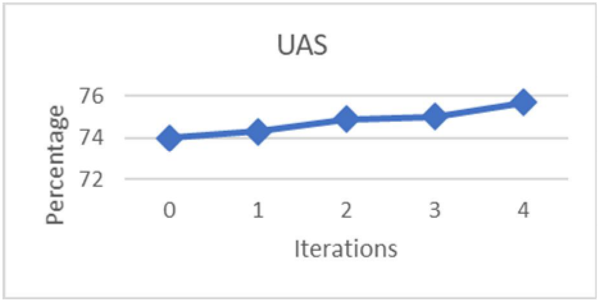


Figure 3: Unlabeled Attachment Score on Development Set

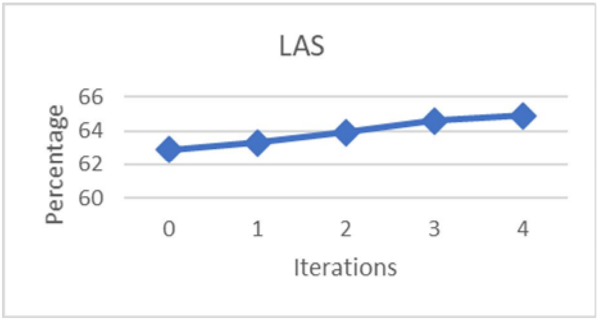


Figure 4: Labeled Attachment Score on Development Set

The reported accuracy of baseline model is LA of 69.8%, UAS of 74% and LAS of 62.9% [4], whereas the final supervised bootstrapping model obtains a LA of 72.1%, UAS of 75.7% and LAS of 64.9%. Overall, the results showed that adding automatically parsed manually corrected training data to the baseline model is beneficial. This iterative parsing corpus method allowed to benefit from the repetition of elements in the data. As an outcome of exposure to repetitive syntactic structures, at each iteration, the parser gradually became better. Recurrently encountered and learned constructs were annotated more accurately by the parser, leaving the manual rectification of only formerly un-encountered, difficult, or infrequent parses each time. At each iteration, the learning procedure became speedier with the addition of freshly parsed data to the training set.

5. ERROR ANALYSIS

An error analysis of bootstrapped parser's output is performed to understand difficulties a dependency parser faces while annotating Urdu tweets. Since the description of UD relations is beyond the scope of this paper, the readers are advised to refer to [15] to accurately comprehend the error analysis.

In bootstrapped models, MaltParser often confused between vocative and nsubj relation. Possible cause of this confusion could be part-of-speech tag proper noun (PROPN) of both vocatives and nominal subjects. Although, this confusion seemed to be reduced to certain extant from third iteration, but it was not resolved completely. There were few error cases of goeswith labeled as compound, while in certain cases vocatives were wrongly parsed as compound, list as discourse or dislocated or vice versa.

Errors related to the verb's argument structure were amongst the most frequent error types. The absence of post-positions, ambiguous post-positions, coordination, etc. were noted as some of the reasons for these errors. The other major source of errors was non-projectivity. This was the main cause for errors in relative clause constructions and tweets with clausal complements. In general, MaltParser encountered problems with different modifiers and clausal complements (advcl, advmod, acl and xcomp). It is often challenging to distinguish these types because the distinction depends only on the complement verb or noun form. In case of adverbial clause modifiers (advcl) and adjectival modifiers (advmod), the problems may be partially due to attachment problems and partially due to the problems in differentiating between complements and modifiers.

Another source of error is confusion between apposition (appos) and nominal modifiers (nmod). The possible reasons may be owing to semantic prerequisite of an apposition for having its head word's referent. Similarly, MaltParser often made mistakes between indirect object (iobj) and oblique (obl). Particularly, the confusion of subjects (nsubj) and objects (obj), adverbial clause (advcl) and adverbial modifier (advmod) was also present. The morphological and semantic proximity of these relations is possibly the cause of errors.
In some cases, adverbial modifiers (advmod) were marked as nominal modifiers (nmod) by MaltParser, whereas in some instances, adjectival modifiers (amod) of nouns were incorrectly marked as the clausal modifiers (acl).

Additional problematic category for MaltParser was conj. With coordination, it is problematic for a parser to resolve the sentence element to coordinate with, since it not only contains one but many heads that can be used to syntactically replace the entire construction. It is not a simple problem of assigning the correct label, but a deeper problem that relates to the whole construction. The number of children of a coordinating conjunction can be more than 2. Consequently, these children can be spread across the entire sentence, leading to long distance dependencies. Likewise, there are no robust cues to identify these children (e.g. commas are not always present). Children can be sub-trees creating long distance dependencies. Similarly, complex interactions of different conjunctions leading to long distance dependencies were additional source of error in bootstrapped models of MaltParser.

5. CONCLUSION

Twitter dependency parsing is a domain of natural language processing that is still in its early stages of development for Urdu. As a result, there is a scarcity of manually annotated corpus for statistical parsers. To overcome the manual annotation bottleneck, this paper described a bootstrapping process for developing a dependency treebank for Urdu Noisy Text.

A trained parser is used to parse the subsequent set of tweets for manual rectification using a simple bootstrapping algorithm. As a result, the size of the treebank has increased from 500 to 800 annotated tweets. These parsing experiments resulted in a significant increase in accuracy over baseline, with LA rising from 69.8 percent to 72.1 percent, UAS rising from 74 percent to 75.7 percent, and LAS rising from 62.9 percent to 64.9 percent.

As future directions of this work, it would be beneficial to investigate methods such as self-training, co-training, active learning, or unsupervised approaches for increasing the treebank size to improve the parsing results obtained in this work.

ACKNOWLEDGEMENT

This paper is produced from PhD thesis of the first author entitled "Dependency Treebanking and Parsing of Urdu Noisy Text", which is submitted at Isra University, Hyderabad, Pakistan. All faculty members of Department of Computer Science, Isra University are acknowledged for their help and support throughout the course of this study.

REFERENCES

1. N. F. Abd Yusof, C. Lin, X. Han, and M. H. Barawi. **Split Over-Training for Unsupervised Purchase Intention Identification**, *International Journal of Advanced Trends in Computer Science and Engineering,* Vol. 9, pp. 3921-3928, 2020.
2. F. Albogamy, A. Ramsay, and H. Ahmed. **Arabic tweets treebanking and parsing: A bootstrapping approach**, in *Proceedings of the Third Arabic Natural Language Processing Workshop*, 2017, pp. 94-99.
3. A. Baig, M. U. Rahman, H. Kazi, and A. Baloch. **Developing a POS Tagged Corpus of Urdu Tweets**, *Computers,* Vol. 9, p. 90, 2020.
4. A. Baig, M. U. Rahman, A. S. Shah, and S. Abbasi. **Universal Dependencies for Urdu Noisy Text**, *International Journal of Advanced Trends in Computer Science and Engineering,* Vol. 10, pp. 1751-1757, 2021.
5. C. Callison-Burch. **Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk**, in *Proceedings of the 2009 conference on empirical methods in natural language processing*, 2009, pp. 286-295.
6. J. Judge, A. Cahill, and J. Van Genabith. **Questionbank: Creating a corpus of parse-annotated questions**, in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics,* 2006, pp. 497-504.
7. P. Kolachina and A. Ranta. **Bootstrapping UD treebanks for delexicalized parsing**, in *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, 2019, pp. 15-24.
8. T. Lingren, L. Deleger, K. Molnar, H. Zhai, J. Meinzen-Derr, M. Kaiser, *et al.*, **Evaluating the impact of pre-annotation on annotation speed and potential bias: natural language processing gold standard development for clinical named entity recognition in clinical trial announcements**, *Journal of the American Medical Informatics Association,* vol. 21, pp. 406-413, 2014.
9. H. Loftsson. **Correcting a PoS-tagged corpus using three complementary methods**, in *Proceedings of the*

12th Conference of the European Chapter of the ACL (EACL 2009), 2009, pp. 523-531.

10. T. Lynn, O. Cetinoglu, J. Foster, E. Uí Dhonnchadha, M. Dras, and J. van Genabith. **Irish treebanking and parsing: A preliminary evaluation**, in *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)* 2012, pp. 1939-1946.
11. A. Miletic, M. Bras, M. Vergez-Couret, L. Esher, C. Poujade, and J. Sibille. **Building a Universal Dependencies Treebank for Occitan**, in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 2932-2939.
12. R. U. Mustafa, M. S. Nawaz, M. I. U. Lali, T. Zia, and W. Mehmood**. Predicting the cricket match outcome using crowd opinions on social networks: A comparative study of machine learning methods**, *Malaysian Journal of Computer Science*, vol. 30, pp. 63-76, 2017.
13. G. Ngai and D. Yarowsky. **Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking**, presented at the 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, 2001.
14. J. Nilsson and J. Nivre. **MaltEval: An evaluation and visualization tool for dependency parsing**, in *LREC*, 2008.
15. J. Nivre, M.-C. De Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, *et al.*, **Universal dependencies v1: A multilingual treebank collection**, in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 165 9-1666.
16. J. Nivre, J. Hall, and J. Nilsson. **Maltparser: A data-driven parser-generator for dependency parsing**, in *LREC*, 2006, pp. 2216-2219.
17. M. S. Rasooli, M. Kouhestani, and A. Moloodi. **Development of a Persian syntactic dependency treebank**, in Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, pp. 306-314.
18. A. A. Raza, A. Habib, J. Ashraf, and M. Javed. **A review on Urdu language parsing**, *Int. J. Adv. Comput. Sci. Appl*, vol. 8, pp. 93-97, 2017.
19. M. Sanguinetti, C. Bosco, A. Lavelli, A. Mazzei, O. Antonelli, and F. Tamburini. **PoSTWITA-UD: An Italian Twitter treebank in universal dependencies**, in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
20. D. Seddah, F. Essaidi, A. Fethi, M. Futeral, B. Muller, P. J. O. Suárez, *et al.*, **Building a user-generated content North-African Arabizi treebank: Tackling hell**, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1139-1150.
21. M. Seraji, B. Megyesi, and J. Nivre. **Bootstrapping a Persian dependency treebank**, *Linguistic Issues in Language Technology*, vol. 7, pp. 1-10, 2012.
22. M. Straka and J. Straková. **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*, 2017, pp. 88-99.
23. F. Zarei, A. Basirat, H. Faili, and M. Mirain. **A bootstrapping method for development of Treebank**, *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 29, pp. 19-42, 2017.
24. C.-C. Chang and C.-J. Lin. **LIBSVM: a library for support vector machines**, *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, pp. 1-27, 2011.
25. R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. **LIBLINEAR: A library for large linear classification**, *the Journal of machine Learning research*, vol. 9, pp. 1871-1874, 2008.