ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1491032021.pdf
https://doi.org/10.30534/ijatcse/2021/1501032021

Towards Silver Standard Dependency Treebank of Urdu Tweets

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

ABSTRACT

Manually annotated corpus is a perquisite for several natural language processing applications including parsing. Nevertheless, annotated corpus is not always available for resource-poor languages, especially when domain under consideration is noisy user-generated data found on social media platforms such as Twitter. To overcome this deficiency of hand-annotated corpus, researchers have focused their attention on semi-automatic corpus annotation methods. This paper describes the experiments carried out using semi-automatic methods like self-training and co-training in an attempt for creating silver-standard dependency treebank of Urdu tweets. Six iterations of each approach were performed using same experimental conditions using MaltParser and Parsito parser, both statistical data driven parsers. For self-training experiments, the best performing MaltParser model was trained on 1250 Urdu tweets, with an accuracy of 70.2% LA, 74.4% UAS, 63% LAS. Whereas the best performing Parsito model was also trained on 1250 Urdu tweets with an accuracy of 70.8% LA, 74.8% UAS, 63.4% LAS. For co-training experiments, best performing MaltParser model was trained on 1500 Urdu tweets, with an accuracy of 70.5% LA, 74.4% UAS, 63.2% LAS. The best performing Parsito model was also trained on 1500 Urdu tweets with an accuracy of 70.5% LA, 74.3% UAS, 63% LAS. Although, there was not much difference between the results of both approaches, co-training results were slightly better for both parsers and is used for generating a silver-standard dependency treebank of 4500 Urdu tweets.

Key words: co-training, dependency parsing, manual annotation, silver-standard, self-training, tweets, Universal Dependencies, Urdu.

1. INTRODUCTION

In computational linguistics, the necessity of hand-annotated resources, particularly treebanks, is well acknowledged. Treebanking is a critical phase in the linguistic resource development for a language [14], particularly for data-driven parsing and many advanced applications such as machine translation [28]. Since their development is costly and time-consuming [5], specifically 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 critical for low-resource languages [32].

Users of Twitter post millions of tweets every day, resulting in a noisy and informal corpus that can be beneficial for applications such as data analysis, language technologies, sentiment analysis, opinion mining, and event detection [19], [30], [1]. However, for many languages of the world, annotated corpora for social media domains are not always available or exist in small quantity.

Because manually generating corpora takes time and effort, some recent studies have investigated ways to automatically annotate corpora with annotations without involving human annotators [23]. The term “silver standard” describes annotation quality amongst a manually annotated gold-standard and the uncorrected automatic processing output [11]. As a result, interest in creating silver standard treebanks grew in recent years [12] to overcome scarcity of hand-annotated data for new language domains such as social media.

Semi-supervised approaches like self-training [17] and co-training [31] are used to automatically parse huge amount of unannotated data with existing parsers. This automatically parsed data can be used as additional training resource in addition to gold standard trees for training improved parser models [17], [38]. Hence, permitting parser learning from its own or from annotations of other parsers.

This paper reports on experiments carried out using self-training and co-training by utilizing state-of-the-art statistical dependency parsers, MaltParser and Parsito in an attempt for creating a silver-standard dependency treebank for Urdu tweets, using relatively small gold-standard treebank, Urdu Noisy Text Dependency Treebank (UNTDT) [3]. 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 [40].

The paper is outlined as follows: Section 2 enlists previous work related to self-training and co-training. In Section 3, materials and methods of this study are described. Results and their discussion are presented in Section 4 and Section 5 respectively. Finally, conclusions are presented in Section 6.

2. RELATED WORK

First application of self-training on PCFG parser was by [7]. However, this attempt didn't yield fruitful results. The author found that during the self-training process, the parser errors are amplified rather than eliminated.

[33] implemented and evaluated self-training in multiple settings, where 500 sentence training data was utilized to parse 30 sentences in each iteration. Only modest improvements were achieved after numerous self-training iterations due to the use of small number of additional sentences.

Solid results of self-training with a 1.1% f-score improvement were reported by [17] with Charniak-parser [8], a two-stage parser which includes a discriminative reranker and a lexicalized context-free parser. Later, same method was applied by [18] for out-of-domain text with good accuracy results.

In 2012 shared task SANCL, majority of phrase structure systems used self-training [25]. The improvement of top ranked system by employing Self-training suggested self-training to be a well-established method for improving out-of-domain constituency parsing accuracy [15]. Though, in SANCL 2012 shared task, no dependency-based system used self-training.

[42] introduced one of several effective methods of dependency parsing using self-training. For Chinese, the unlabelled attachment score was enhanced around 1% point by them. They added parsed sentences with high ratio of dependency edges with short distance spanning, i.e., the head and dependent are in close proximity with the purpose of observing whether higher accuracy is achieved by short dependency edges in comparison to longer edges.

A multi-iterative self-training method was applied to Hindi by [13] for improving training domain parsing accuracy. In every iteration, additional 1,000 sentences were added to a small initial 2,972 sentence training set, selected based on their parsing score. After 23 self-training iterations, their labelled and unlabelled attachment scores increased by 0.7% and 0.4% respectively over the baseline score.

For Dutch parsing, [26] used a modified Alpino parser [36], for producing dependency trees by applying single and multiple iterations of self-training. She found that in some cases, self-training would only bring a minor improvement but deteriorated with the addition of additional unlabelled data.

For out-of-domain parsing of Italian, self-training was used by [27] in combination with similarity-based sentence selection and dependency triplets statistics. They found unstable consequences of self-training and will not bring improvement. Self-training was applied to dependency parsing of nine languages by [6] and [4]. Only negative results were reported by [6] for their self-training evaluations of dependency parsing. Likewise, [4] observed positive effects on Swedish only. [16] used self-training in bootstrapping fashion to semi-automate dependency treebank development for Irish. 1945 sentences were annotated using Mate parser in iterative manner where 323 sentences were labelled in a single iteration. She performed six self-training iterations and reported 70.2% LAS and 79.1% UAS for Irish.

Co-training was first used in the syntactic analysis domain by [31], who applied it to a phrase structure parser.

The author used a subgroup of 9695 sentences from annotated Wall Street Journal data as well as a larger set of unlabelled data (roughly 30K sentences) as the initial training set. The maximum possible n sentences from two views were added to the training set of the subsequent repetition in each co-training iteration. After 12 iterations, substantial increase in both precision (by 7.79%) and recall (by 10.52%) was achieved by the parser in co-training experiments.

In dependency parsing domain adaption track of the CoNLL 2007 shared task, co-training was employed successfully by [29]. Two out-of-domain trained models were used by them for parsing domain-specific unlabelled data. Next, identical using output of both the models was sample identified using selection and lastly, these uncorrected parsed trees were added to the actual out-of-domain annotated training set.

To evaluate a state-of-the-art neural network parser, [37] employed normal agreement-based co-training and tri-training. The annotations of additional training data were agreed by the Berkeley constituency parser [24] and a traditional transition-based parser (zPar) [39] in their work. This extended training data was then used to retrain the zPar parser and neural network parser. Gain of the neural network parser from the tri-training was about 0.3% and is 1% higher than the state-of-the-art accuracy. Although, only negative effects were reported in their zPar parser co-training evaluation. [16] experimented with simple co-training, agreement-based co-training, and disagreement-based co-training in bootstrapping fashion to semi-automate dependency treebank development for Irish. 1945 sentences were annotated using Mate and Malt parsers in iterative manner where 323 sentences were labelled in a single iteration. She performed six self-training iterations of each co-training variant. According to her findings, these experiments did yield performance improvement over her baseline score, but the gain was not that much.

3. MATERIALS AND METHODS

3.1 Parsers

This study uses MaltParser [22] and Parsito [34]. MaltParser is a greedy transition-based dependency parser with a liblinear model at its core. Parsito is also a transition-based dependency parser inspired by the Stanford neural dependency parser [9]. It improved the original parser by adding two major features: a search-based oracle that is like a dynamic oracle, and morphological feature set which is provided by the UD corpus. Parsito is now part of the UDPipe [35] parsing pipeline, which handles tokenization, morphological analysis, and POS tagging.

3.2 Evaluation Metrics

Labelled Attachment Score (LAS), Unlabelled Attachment Score (UAS), and Label Accuracy (LA) are used to evaluate parser performance using MaltEval [20]. These three metrics are essentially token-level accuracies, which take into account all test data tokens and assign equal weight to each token in the evaluation.

The formula for calculating LAS, UAS, and LA is shown in Equations (1), (2), and (3):

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

$$UAS = \frac{number\ of\ correct\ head\ labels}{total\ tokens} \quad (2)$$

$$LA = \frac{number\ of\ correct\ labels}{total\ tokens} \quad (3)$$

3.3 Corpus

The gold-standard corpus for parser training in this study is UNTDT (Urdu Noisy Text Dependency Treebank) [3], a manually annotated dependency treebank of 500 Urdu tweets. By applying the Universal Dependencies (UD) framework [15] to the particularities of social media text, the treebank is annotated at the morphological and syntactic levels. For a complete review of the treebank, see [3]. 10-fold cross validation was used to validate this treebank. Authors reported the best average accuracy scores of 74 percent UAS, 62.9 percent LAS, and 69.8 percent LA. In the current study, this score serves as a baseline.

3.4 Experiments

The pre-processed corpus of raw 4500 Urdu tweets collected by [2] was utilized in this study. A model of UDPipe [35] which is, a neural network-based trainable pipeline system, was trained on UNTDT for performing tokenization, lemmatization, POS tagging, and morphological analysis of these pre-processed Urdu tweets. The UDPipe model not only completes the tasks listed above, but it also converts tweets into the CONLL-U format, which is the type of data required by MaltParser and Parsito for input. It should be noted here that the output of UDPipe was not manually corrected and was used as is in the self-training and co-training experiments. Out of these 4500 Urdu tweets, 1500 tweets were randomly selected to be used for self-training and co-training experiments. For parser training, UNTDT was segmented in 300 tweets training set, 100 tweets each development and test set.

For self-training experiments, algorithm used by [16] is adopted and is shown in Fig 1. A single dependency parser, A is required by this algorithm. First, the parser A is trained on the existing manually labeled data, L to create a model MiA. Set of UDPipe processed 1500 randomly selected Urdu tweets (U) is divided into 6 groups, where each group contains 250 sentences U^i. For each of the six repetitions (i = 1...6), U^i is parsed to produce P^i_A. Every time, freshly parsed sentence set (P^i_A) is appended in the training set L^i_A to form a large L^{i+1}_A training set. Then, by using the new training set for training, new parsing model (M^{i+1}_A) is then obtained.

Self-training 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:
$L^0_A \leftarrow L$.
$M^0_A \leftarrow \text{Train}(A, L^0_A)$
For *i* = 1 to N do
    $U^i \leftarrow$ Add set of unlabeled sentences from *U*
    $P^i_A \leftarrow Parse\ (U^i, M^i_A)$
    $L^{i+1}_A \leftarrow L^i_A + P^i_A$
    $M^{i+1}_A \leftarrow Train\ (A, L^{i+1}_A)$
End For

Figure 1: Self-training Algorithm

Although self-training experiments require a single dependency parser, here MaltParser is utilized along with Parsito parser incorporated in UDPipe and the result of both parsers are compared.

Both MaltParser and Parsito parser are trained on same 300 gold standard tweets used in [3]. MaltParser is trained using baseline parser model settings of [3] whereas Parsito is trained using link2 algorithm. The algorithm shown in Fig. 1 was repeated for six iterations in two separate experiments. In first experiment, MaltParser is used whereas in second experiment, Parsito replaced MaltParser. At each iteration, 250 randomly sampled tweets were parsed and added to gold standard corpus of 300 sentences without manual correction to induce a new parser model. After each iteration, the accuracy of parser model is assessed on the development set and the next batch of 250 tweets is parsed using the newly induced model. However, it should be noted here that the sentences parsed during self-training were not added to the original gold standard treebank as they were not manually validated. Therefore, the size of the treebank remained same i-e. 500 gold standard tweets at the end of the self-training experiments.

Co-training Algorithm

A and *B* are two different parsers.
M^i_A and M^i_B are models of *A* and *B* at step *i*.
P^i_A and P^i_B are a sets of trees produced using M^i_A and M^i_B.
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 and L^i_B are a labelled training data for *A* and *B* at step *i*.
Initialize:
$L^0_A \leftarrow L^0_B \leftarrow L$.
$M^0_A \leftarrow \text{Train}(A, L^0_A)$
$M^0_B \leftarrow \text{Train}(B, L^0_B)$
For *i* = 1 to N do
    $U^i \leftarrow$ Add set of unlabeled sentences from *U*
    $P^i_A \leftarrow Parse\ (U^i, M^i_A)$
    $P^i_B \leftarrow Parse\ (U^i, M^i_B)$
    $L^{i+1}_A \leftarrow L^i_A + P^i_B$
    $L^{i+1}_B \leftarrow L^i_B + P^i_A$
    $M^{i+1}_A \leftarrow Train\ (A, L^{i+1}_A)$
    $M^{i+1}_B \leftarrow Train\ (B, L^{i+1}_B)$
End For

Figure 2: Co-training Algorithm

For co-training experiments, the algorithm used by [33] is adopted and is presented in Fig. 2. The algorithm requires two dependency parsers, A and B. The parsers A and B are first trained on the existing manually labeled data, L to create models M^i_A and M^i_B. Set of UDPipe processed 1500 randomly selected Urdu tweets (U) used in self-training experiments were used in co-training experiments as well by dividing them into 6 groups, where each group contains 250 tweets U^i. For each of the six repetitions (i = 1...6), U^i is parsed to produce P^i_A and P^i_B. This time, the newly parsed tweet set P^i_B is added to the training set L^i_A for making a large training set L^{i+1}_A. Conversely, the newly parsed tweet set P^i_A is added to the training set L^{i+1}_B for making a large training set L^{i+1}_B. Two new parsing models (M^{i+1}_A and M^{i+1}_B) are then obtained by training A and B with their new training sets respectively.

In co-training experiments, both MaltParser and Parsito parser are again utilized with the same parser settings as that of self-training experiments and are trained on 300 gold standard tweets.

The algorithm shown in Fig. 2 was repeated for six iterations. At each iteration, 250 randomly sampled tweets were parsed using both Malt and Parsito. The output of Malt is added to the training set of Parsito without manual correction to induce a new Parsito parser model whereas parsed output of Parsito is added to the training set of Malt for induction of new model, again without manual correction. At the end of each repetitions, both models are retrained with on updated training data and the accuracy of newly induced parser models is then evaluated on the development set and the next batch of 250 tweets is parsed using the newly induced models. However, it should be noted here that the tweets parsed during co-training are not added to the original gold standard treebank as they are not manually validated. Therefore, the size of the treebank remains same i-e. 500 gold standard tweets at the end of the co-training experiments. At the end of final iteration, the accuracy of both Malt and Parsito's final induced models is tested with the test set.

5. RESULTS

The self-training experimental results of are shown in Fig 3 and Fig 4. The 3rd iteration produced best performing MaltParser model, trained on 1250 tweets, with an accuracy of 70.2% LA, 74.4% UAS, 63% LAS. However, after 3rd iteration, performance of the model dropped and no significant improvement over baseline score was observed.

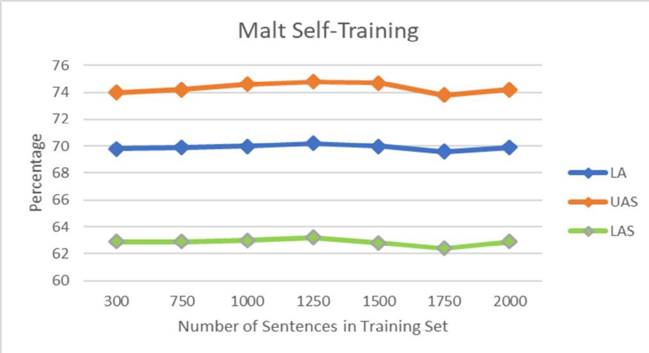

Figure 3: Self-Training Results of MaltParser on Development Set

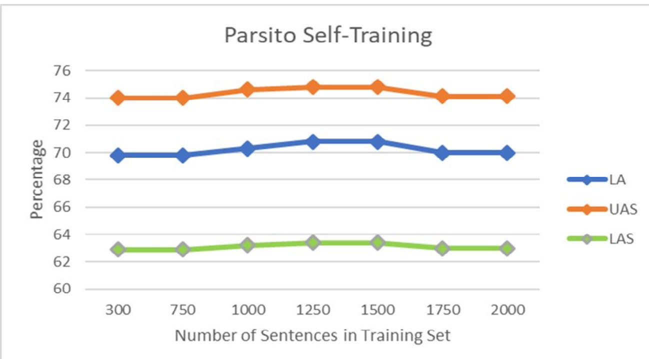

Figure 4: Self-Training Results of Parsito parser on Development Set

The best performing Parsito model was also trained on 1250 tweets, at the 3rd iteration with an accuracy of 70.8% LA, 74.8% UAS, 63.4% LAS. However, performance of Parsito model also dropped slightly on 5th iteration, although not as much as Malt's model and slight improvement over baseline score has also been observed. Although, this improvement was not statistically significant.

The co-training experimental results of are presented in Fig 5 and Fig 6. Calculation of statistical significance of the results is performed using McNemar's test (p <= 0:05) provided by MaltEval.

The 4th iteration produced best performing MaltParser model trained on 1500 tweets, with an accuracy of 70.5% LA, 74.4% UAS, 63.2% LAS, though, the improvement is not statistically significant. However, after 4th iteration, performance of the model dropped and no significant improvement over baseline score has been observed.

The best performing Parsito model was also trained on 1500 tweets, at the 4th iteration with an accuracy of 70.5% LA, 74.3% UAS, 63% LAS. However, performance of Parsito model also dropped from 4th iteration, and no significant improvement over baseline score has also been observed.

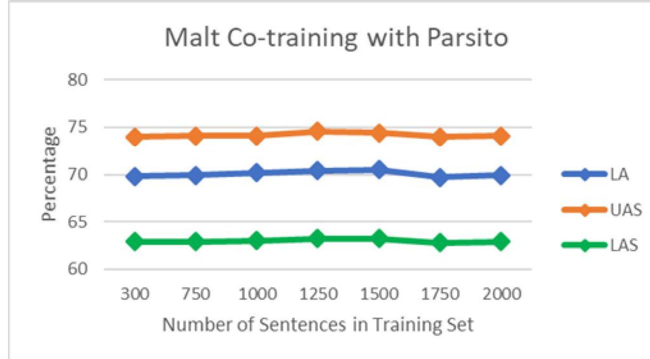

Figure 5: Co-Training: Results of MaltParser on Development Set

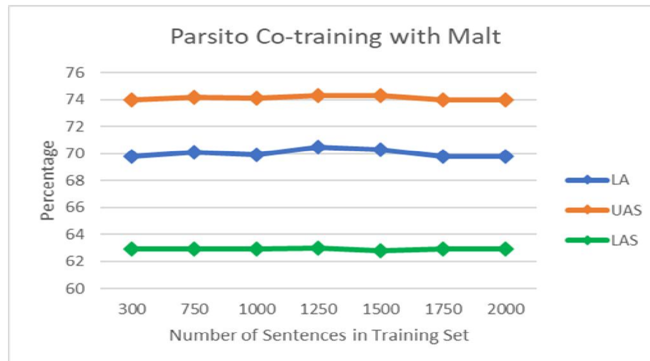

Figure 6: Co-Training: Results of Parsito parser on Development Set

5. DISCUSSION

To understand the difficulties a dependency parser faces while annotating Urdu tweets, an error analysis of parser output is performed. Because the description of UD relations is outside the scope of this paper, readers are advised to consult [15] for a thorough understanding of the error analysis.

In semi-supervised models, the errors discussed in [41] were observed along with some new trends. For example, there were several error cases of advcl labeled as xcomp or conj, whereas nummod was mistakenly parsed to nmod or compound, acl to advcl or advmod, and obl to advcl. The reasons for these parsing errors were due to POS errors because automatically tagged corpus was utilized in both self-training and co-training experiments.

MaltParser showed an obvious and substantial difference in this regard, with performance drop for long tweets with automatic POS tags. While Parsito's output is unexpectedly consistent with automatic POS tags. When parsing the data using automatic POS tags, Parsito surpasses MaltParser for punctuation, conjunctions, and determiners. This in turn resulted in improving MaltParser's performance in co-training experiments. However, using MaltParser models for training Parsito for co-training experiments didn't resulted in any significant improvement in parsing performance of Parsito. Overall, MaltParser was the better performing model, although its edge over Parsito is rather small. Significant improvement due to the usage of automatically tagged POS data was not seen. Same self-training and co-training experiments with gold standard POS tagged data may be performed to compare the results of both approaches. Compared to full dependency Treebank annotation, the cost of POS annotation is considerably less. The experiments performed in [3] and [41], confirmed the link amongst POS accuracy and parser accuracy. Using an accurate POS tagger and larger unlabelled data, co-training specifically might be helpful in developing a practically high-end treebank for this domain with comparatively minor cost of manual annotation.

While comparing the performance of a traditional dependency parser with neural network model, MaltParser was the highest performing model in the semi-supervised experiments, although its performance gain over Parsito is rather small i.e for LA, UAS and LAS, 0.6%, 0.4% and 0.4% respectively. This is because UDPipe needs a lot of training data since it uses neural networks, but the treebank used in this study is moderately small.

6. CONCLUSION

Due to the expensive nature of manual annotation, both in terms of time and effort, interest in creating silver standard treebanks grew in recent years. The term "silver standard" describes annotation quality amongst a manually annotated gold-standard and the uncorrected automatic processing output.

The semi-supervised learning experiments performed in this study showed that addition of automatically parsed data in the training set does not improve accuracy of parsing, neither have a significant negative effect. This led to the creation of a silver standard treebank for Urdu tweet corpus (4500 tweets, 127,337 tokens) using best scoring co-training parser model. Nevertheless, it must be noted that contrary to the gold-standard treebank, which is presumed to be annotated correctly, the silver standard treebank is annotated automatically and therefore can be faulty. For building a parser, this is not a good enough resource. The purpose of creating a silver standard treebank in this study is not to train a parser model out of it, but to have a resource readily available which can be used to accelerate the development of a gold standard treebank with much less effort and time as compared to manual annotation. Since co-training experiments showed slightly better results as compared to self-training, further co-training experiments like agreement-based co-training and disagreement-based co-training experiments can be explored as future directions of this work.

ACKNOWLEDGEMENT

The basis for this paper is first author's PhD thesis, "Dependency Treebanking and Parsing of Urdu Noisy Text", which is submitted to Isra University in Hyderabad, Pakistan. Isra University's Department of Computer Science faculty members are thanked for their assistance and support throughout the 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. A. Baig, M. U. Rahman, H. Kazi, and A. Baloch. **Developing a POS Tagged Corpus of Urdu Tweets**, *Computers*, Vol. 9, p. 90, 2020.
3. 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.
4. A. Björkelund, O. Cetinoglu, A. Falenska, R. Farkas, T. Müller, W. Seeker, *et al.* **The IMS-Wroc law-Szeged-CIS entry at the SPMRL 2014 shared task: Reranking and morphosyntax meet unlabeled data**, *Proceedings of the Shared Task on Statistical Parsing of Morphologically Rich Languages, Dublin, Ireland. Dublin City University*, 2014.
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. C. Cerisara. **Semi-supervised experiments at LORIA for the SPMRL 2014 Shared Task**, in *Proc. of the Shared Task on Statistical Parsing of Morphologically Rich Languages*, 2014.
7. E. Charniak. **Statistical parsing with a context-free grammar and word statistics**, *AAAI/IAAI*, vol. 2005, p. 18, 1997.
8. E. Charniak and M. Johnson. **Coarse-to-fine n-best parsing and maxent discriminative reranking**, in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005, pp. 173-180.
9. D. Chen and C. D. Manning. **A fast and accurate dependency parser using neural networks**, in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 740-750.

10. W. Chen, Y. Wu, and H. Isahara. **Learning reliable information for dependency parsing adaptation**, in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 2008, pp. 113-120.
11. K. Eckart and M. Gärtner. **Creating Silver Standard Annotations for a Corpus of Non-Standard Data**, in *13th Conference on Natural Language Processing (KONVENS 2016)*, 2016, pp. 90-96.
12. M. Filannino and M. Di Bari. **Gold standard vs. silver standard: the case of dependency parsing for Italian**, *CLiC it*, pp. 141-145, 2015.
13. R. Goutam and B. R. Ambati. **Exploring self training for Hindi dependency parsing**, in *Proceedings of 5th International Joint Conference on Natural Language Processing*, 2011, pp. 1452-1456.
14. 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.
15. J. Le Roux, J. Foster, J. Wagner, R. Kaljahi, and A. Bryl. **DCU-Paris13 systems for the SANCL 2012 shared task**, in *NAACL 2012*, 2012, pp. 1-4.
16. T. Lynn. **Irish dependency treebanking and parsing**, PhD thesis, Dublin City University and Macquarie University, 2016.
17. D. McClosky, E. Charniak, and M. Johnson. **Effective self-training for parsing**, in *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, 2006, pp. 152-159.
18. D. McClosky, E. Charniak, and M. Johnson. **Reranking and self-training for parser adaptation**, in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006, pp. 337-344.
19. 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.
20. J. Nilsson and J. Nivre. **MaltEval: An evaluation and visualization tool for dependency parsing**, in *LREC*, 2008.
21. 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. 1659-1666.
22. J. Nivre, J. Hall, and J. Nilsson. **Maltparser: A data-driven parser-generator for dependency parsing**, in *LREC*, 2006, pp. 2216-2219.
23. A. Oellrich, N. Collier, D. Smedley, and T. Groza. **Generation of silver standard concept annotations from biomedical texts with special relevance to phenotypes**, *PloS one*, vol. 10, p. e0116040, 2015.
24. S. Petrov and D. Klein. **Improved inference for unlexicalized parsing**, in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 2007, pp. 404-411.
25. S. Petrov and R. McDonald. **Overview of the 2012 shared task on parsing the web**, in *Proceeding of the First Workshop on Syntactic Analysis of Non-Canonical Language*, 2012.
26. B. Plank. ***Domain adaptation for parsing***: Citeseer, 2011.
27. B. Plank and A. Søgaard. **Experiments in newswire-to-law adaptation of graph-based dependency parsers**, in *International Workshop on Evaluation of Natural Language and Speech Tool for Italian*, 2012, pp. 70-76.
28. 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.
29. K. Sagae and J.-i. Tsujii. **Dependency parsing and domain adaptation with data driven LR models and parser ensembles**, *Trends in Parsing Technology*, pp. 57-68, 2010.
30. 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.
31. A. Sarkar. **Applying co-training methods to statistical parsing**, in *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, Pennsylvania, USA, 2001, pp. 1-8.
32. 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.
33. M. Steedman, M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, *et al* **Bootstrapping statistical parsers from small datasets**, in *10th Conference of the European Chapter of the Association for Computational Linguistics*, 2003, pp. 331-338.
34. M. Straka, J. Hajic, J. Straková, and J. Hajic Jr. **Parsing universal dependency treebanks using neural networks and search-based oracle**, in *International workshop on treebanks and linguistic theories (tlt14)*, 2015, pp. 208-220.

35. 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.
36. G. Van Noord. **At last parsing is now operational**, in *Actes de la 13ème conférence sur le Traitement Automatique des Langues Naturelles. Conférences invitées*, 2006, pp. 20-42.
37. D. Weiss, C. Alberti, M. Collins, and S. Petrov. **Structured training for neural network transition-based parsing**, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 323–333, 2015.
38. D. Zeman and Z. Žabokrtský. **Improving parsing accuracy by combining diverse dependency parsers**, in *Proceedings of the Ninth International Workshop on Parsing Technology*, 2005, pp. 171-178.
39. Y. Zhang and J. Nivre. **Transition-based dependency parsing with rich non-local features**, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 188-193.
40. 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.
41. A. Baig, M. U. Rahman, S. Abrejo, S. Qureshi, S. Tunio and S. S. Baloch. **Bootstrapping Dependency Treebank of Urdu Noisy Text**, *International Journal of Emerging Trends in Engineering Research*, Vol. 9, No. 8, 2021.
42. W. Chen, Y. Wu, and H. Isahara. **Learning reliable information for dependency parsing adaptation**, in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 2008, pp. 113-120.