International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 05 | May 2019 www.irjet.net p-ISSN: 2395-0072

An Analysis of Recent Advancements on the Dependency Parser

Patel Hemalkumar Ratilal[1]

[1]*Student, School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu, India*

Abstract - *Dependency Grammar (DG) is a class of current syntactic hypotheses that are all in light of the dependency connection as opposed to the constituency relation. Dependency is the thought that linguistic units, e.g. words, are associated with each other by coordinated connections. The finite verb is taken to be the structural focal point of condition structure. All other syntactic units are either straightforwardly or in a roundabout way associated with the verb as far as the coordinated connections, which are called dependencies. Dependency grammar has been a great help in NLP (Natural Language Processing).Dependency parser and parsing techniques have an upper hand when it comes to the understanding of the NLP. This paper reviews the recent advancement in the dependency parser and its applications.*

Key Words: Dependency Grammar , Dependency Parser, Parse Tree, NLP, Semantic Dependencies, Morphological Dependencies, Prosodic Dependencies

1. INTRODUCTION

Dependency Grammar is different from the conventional grammar in terms of syntactic structure. While Context Free Grammar (CFG) is a generative grammar, Dependency Grammar is completely descriptive. It only depends on the dependency relation that exists between lexical items present in the sentence. This is a great virtue of the DG which makes it a suitable grammar for the free order languages. Even though it is used for the free order languages, some frameworks are there which follows the DG, such as Link Grammar, Operator Grammar, Lexicase, Word Grammar, Meaning Text Theory etc. There are various ways to represent the dependencies which are purely based upon the type of the dependency, mostly semantic, morphological, prosodic and syntactic. Semantic Dependencies are based upon the predicates and arguments, or triples. Morphological Dependencies are constructed on the words and sometimes parts of words. When a word or its part affects another word, then it becomes the morphological. Nature of the clitics is recognised by the Prosodic Dependencies. Clitic generates its own word by merging with its host. DG has its own style of approaching the sentences in linear order. DGs force to focus on the content by being minimal than the conventional Constituency-based Grammars. The Syntactic Function is considered primitive in DGs. These functions are annotated as labels in the Parse Tree.

2. Different Directions of Development

M. Kumar and M. Dua [1] used the ability of the Paninian Grammar to provide the great solution for the languages with the word free order and applied that to Stanford parser which in fact provides the better dependencies for the languages having fixed order. English is one such language which supports Stanford Parser. Previous results showed that Paninian Grammar is could be applied to the other Indian Languages with much higher efficiency.

Syntax along with the semantics was caught and handled by VerbNet. They have exhibited the issues that experienced while doing an adjustment and proposed the answer for beating these issues. They utilized Hindi Dependency parser for confirmation of comes about. With this adjustment of Stanford Parser hand in hand syntax tree of English and Hindi can be made.

M. Savic´, G. Rakic´, Z. Budimac and M. Ivanovic [2] displayed SNEIPL, a novel way to deal with the extraction of programming systems that depends on a dialect autonomous, enhanced solid linguistic structure tree representation of the source code. The appropriateness of the methodology is exhibited by the extraction of programming systems speaking to genuine, medium to extensive programming frameworks written in various languages which have a place to various programming paradigms. To examine the fulfilment and rightness of the methodology, class coordinated effort systems (CCNs) extricated from certifiable Java programming frameworks are contrasted with CCNs acquired by different devices. Dependency finder was used to fetch the dependencies having the entity levels from the Java bytecode. In addition to that, they used the Doxygen, which is efficient in extracting the dependency using the fuzzy parsing, which in turn is complete to independent to the language being used. They compared these results with that of SNEIPL. They demonstrated that the SNEIPL can form highly precise networks better than the Dependency Finder and Doxygen.

S. N, S. M. Jasmine and S. Joseph [3] proposed an machine learning approaches like ADTree and Naïve Bayes Classifier along with the dependency parsing to extract the hypernym and meronym relations in the given sentences. The approach utilized the path between nouns in the syntax tree generated.

W. Z. YUAN, Z. MIAO, W. ZHU and W. LIU [4] developed a cross breed strategy for identifying ASR blunder in talked turns is produced. The mistaken content is locally dissected first by neighboring co-event relations utilizing ngram model. They used the ability of Dependency Parser to find the long distance dependency relations to analyze the text globally. Their tests demonstrate that we can utilize data from a Dependency parsing stage together with n-gram model not just to identify incorrect ASR speculations that can cause understanding blunders, additionally dependably find mistakes and here and there right them as the speculations are being handled. They also showed that building Chinese language parser has much more complexity than normal languages.

M. Shen, D. Kawahara and S. Kurohashi [5] they demonstrated another approach for Dependency Parsing that can use complex subtree representations by applying productive subtree choice methods. Chinese Treebank along with the Penn Treebank were analyzed using this approach to show the viability of the novel method. Their framework accomplishes the best execution among known administered frameworks assessed on these datasets, progressing the benchmark precision from 91.88% to 93.42% for English, what's more, from 87.39% to 89.25% for Chinese. Other semi-supervised parsing methods such as word-clustering along with the subtree feature integration show no overlap with their novel approach which is a clear advantage of this method.

C. Lee, G. G. Lee and M. Jang [6] brought the novel model for dependency structure called Dependency Structure Language Model to counter the weakness of models like unigram and bigram in TDT (Topic Detection and Tracking). Models like unigram and bigram lack the ability to capture the semantics in records. In addition to that, dependencies having the long distance between them are not supported by those models. The novel model they proposed utilized the Chow Expansion Theory along with the dependency parse tree produced by a dependency parser. Long-distance dependencies can be easily taken care by the dependency structure language model. They concluded that Dependency Structure Language Model beat the conventional language model. In addition to that, they showed that the proposed model has advantages over the bigram in TDT. However, the proposed model has high computational cost, which needs to be taken care in future.

W. Chen, M. Zhang and Y. Zhang [7] demonstrated an approach to tackle the problem of feature sparseness in dependency parsing. Their method leads to learning of feature embeddings automatically. In their methodology, the component embeddings are gathered from a lot of auto-parsed information. To begin with, the sentences in crude information are parsed by a gauge framework, what's more, they acquired Dependency Trees. They proposed two learning methods to induce feature embeddings using the representation of every model feature by utilizing neighboring features on Dependency Trees. They extracted the bunch of features for Dependencies that are based on Graph by utilizing feature embeddings. The new parsers are highly capable of utilizing settled hand-composed features along with the hidden class representation of features. The new parsers accomplish critical execution changes over a solid pattern.

S. Pyysalo, F. Ginter, T. Pahikkala, J. Boberg, J. J¨arvinen and T. Salakoski [8] demonstrated an assessment of Link Grammar along with the Connexor Machinese Syntax (CMS), two noteworthy and highly researched Dependency Parsers, on a custom hand-explained corpus comprising of sentences with respect to the interactions between protein and protein. To express the protein-protein correspondence, interaction graph was used. They did the performance benchmark of the both the parsers in with measurements like a number of full parses along with the individual dependencies. They concluded that both the parsers lack the ability to parse the biomedical English. While CMS significantly beat the Link Grammar, their analysis implied that there are some ways in which the Link Grammar could perform better in the specific domain. They did the deep evaluation of Link Grammar, addressing issues including panic mode and sampling.

W. Chen, M. Zhang, Y. Zhang and X. Duan [9] proposed the highly accurate framework to address the sparseness problem. The framework was used to train the discriminative models.

It works on the concept of frequency bucketing on big data processed through automation. In addition to that, the feature clusters are utilized to define the meta features. The proposed approach is highly efficient when it comes to dealing with dependency parsing along with the part of speech tagging. The meta-parser is proved highly competitive with the best available parsers and most accurate in Chinese data for the Dependency Parsing Task. The tagger showed the reduced error rate of 5% than the state-of-the-art baseline systems when processed on the Chinese and English language data. In addition to that, meta features are proven efficient while dealing with the unknown features.

J. Yu and W. Chen [10] proposed an automatic identification method to address the recognition problem for co-ordinate structures in Chinese Dependency. They introduced the new features related to the word pairs collected into the Dependency Parser. In the first step, they influenced two hand-made standards to remove exceptionally precise direction word sets as seed words. The second step is to use seed words to extricate coordinate structures in the corpus for further utilization of coordinate word pair extraction. Exploratory results demonstrated that the extricated coordinate word sets can essentially enhance the precision on coordinate structure Dependency Analysis.

R. Goutam [11] utilized the Partial Parsers to investigate the impact of applying bootstrapping strategies such as self-preparing along with the co-preparing on Hindi Dependency Parsers. They went through distinctive criteria for selecting very sure Partial Parses that are going to be useful for bootstrapping. Our outcomes appeared that co-preparing between Partial Parser, Malt and MST utilising understanding check as determination criteria played out the best and gave huge change in execution improvement at baseline. Utilising co-training, we could enhance the exactness of the low-performing Partial Parser to that tantamount to the cutting edge precision for programmed Hindi Dependency Parsing.

W. Wang and M. P.Harper [12] showed the assessment of a language model based on Statistical Constraint Dependency Grammar (CDG) parser. Grammatical structure of a sentence is represented by dependency relations between words utilized in the sentences. CDG applies some restrictions to decide that structure. This Language Model outperforms the conventional CDG based Language Model by showing reduced WER(Word Error Rate) compared to the latter. In addition to that, other Language Model based on state of the art parser were lacking in the ability of reduced the WER(Word Error Rate).The reason of the being powerful is its ability to tightly integrate with various sources.

S. Majidi and G. Crane [13] did the detailed study on the error rate of the different annotators for the ancient Greek Language. They compared the error rate of the two type of annotators, Student Annotators and Dependency Parser Annotators. They found out the error rate almost similar for both type. Both students and parser find the same difficulty. The errors of automatic parsers and that of the student overlap at some stage. We can consolidate these different tasks. Teaching students an Ancient Greek Language with increasing the accuracy of the linguistic information.

M. Wang, H. Xia, D. Sun, Z. Chen, M. Wang and A. L [14] proposed a novel content digging technique for effectively recovering and extricating protein phosphorylation data from writing. They utilized the Natural Language Processing (NLP) techniques to change every sentence into Dependency Parse Tree. This provides the edge to the Dependency Parse Trees in reflecting inborn relationships of phosphorylation-related watchwords. This helped to extract the detailed insight about substrates, kinases and phosphorylation locales. Contrasted and other existing methodologies, the proposed technique exhibits essentially enhanced execution, recommending it is a capable bioinformatics way to deal with recovering phosphorylation insight from a vast literature. Their method has the ability to simplify the complex sentences which give that the ability to outperform any existing method. Even information extracted from the Dependency Trees are precise and highly accurate.

Weichselbraun and N. S¨usstrunk [15] proposed a way to optimize the existing methods by focusing on the feature extraction along with the parsing constraint optimization. In addition to that, this approach could be applied to the existing methods such as MDParser (Volokh, 2013) and Stanford parser (Chen and Manning, 2014). They compared them by applying them for the English Universal Dependencies Corpus. They measured the performance by two factors - Unlabeled Attachment Score-UAS (correctly assigned head) and Labeled Attachment Score-LAS (fraction of heads and types which have been correctly recognized).

The results show that Syntactic Parser outperforms rest. After applying the optimization throughput was increased four times of the original throughput of MDParser. In addition to that, this performance gain was not acquired at the cost of the accuracy. Not only that, Syntactic Parser provided better accuracy in some cases. Syntactic Parser also showed the very good performance in other languages other than English like French, German in LAS outperforming MaltParser and MST.

Rahul.C, Dinunath.K, R. Ravindran and K.P.Soman [16] proposed their work for English to Malayalam Statistical Machine Translation by specifying rule based reordering along with the morphological information. There are two ways which have been demonstrated extremely successful – rearranging the English Based Sentences to Malayalam Syntax – Word separation by utilizing the root suffix method on both the languages. The phrase-based system is beaten by the proposed method. Their method leads to the conclusion that the Indian Languages - which are not rich in morphology and parsing tools should be treated with this method. Their method reduced the need of such tools for the Indian Languages following SOV (Subject – Object – Verb) format.

L. D. Caro and M. Grella [17] proposed a novel algorithm for Sentiment Analysis (SA) along with the model which is context-based. Their algorithm depends on upon the rules of propagation for the Dependency Tree at the syntactic level. Their model tunes the user's opinions on the basis of the context. They implemented their ideas through SentiVis utilizing Data Visualization ability of the SentiVis for the Sentiment Analysis. Separated sentiments were arranged in the 2D format for further processing. 2D format for useful for calculating the strength with variability. Not only that this 2D table could be used according to the user's wish such as finding fruitful facts along with the removing the false positives and negatives.

B. V. S. Kumari and R. R. Rao [18] gained the insight about some of the popular parsers including MaltParser, TurboParser, MSTParser, Easy First Parser and ZPar by different settings of features. They showed how different parser gets affected by different settings. They also did the benchmark on the performance of the different parsers. Telugu Dependency Treebank from

the ICON 2010 tools contest was used to test the reports. The state-of-the-art performance was 91.8% in UAS and 70.0% in LAS. They pointed out that while building the datasets we need to keep in mind that a good mixture of long and short sentences along with the short and long distance dependencies should be there in the test language. Telugu is one such language. They also concluded that a parser should be given the ability to add/remove some of the features so that parser could learn from the datasets and develop on its own.

H. Mohamed, N. Omar, M. J. A. Aziz and S. A. Rahman [19] examined the Dependency Grammar needed to develop the Malay Language Grammar Parser. They proposed the possibilities for generating Probabilistic Dependency Malay Parser utilizing annotated relation based on the English Language. They defined the rules of the inter-relation of the Malay and English. They projected the Tree Structure which was utilized to train the stochastic analyzer. That training produced the tree lattices having Malay Dependency Trees. Those lattices also had the probabilities so that information could be gained and tested were performed using the decoders.

P. N. Pelja, R. P, S. G. M and R. Binu [20] proposed and automatic AMR tool to address the issue of time-consuming tasks such as abstract meaning representation. Their experiments proves that their method performs up to the point with 80% accuracy on simple sentences. Although they did not perform the tests on the complex sentences, they claim that their method could be developed to support and handle the complex sentences of the English Language.

Table 1: Comparative Analysis

Ref. NO.	Problem Addressed	Methodology	Merits	Demerits
1	Using Paninian Grammar for annotation scheme for English	Use of Karaka relations mapping	English-Hindi parallel treebank with Paninian Grammar	Max accuracy achieved so far is still less than 75%
2	Extraction of Software Networks	SNEIPL – a novel approach based on language-independent and concrete syntax tree (eCST) of source code	Highly precise networks better than Dependency Finder and Doxygen, Language-independent, more precise than available tools	It can only fetch software networks and cannot do further reverse engineering
3	Extraction the relationships between entities exist in the text	Use of the Dependency parsing techniques along with the machine learning techniques	Better than WordNet, Uses probability, F-score with ADTree classifier (75%) for hypernym relation, Naïve Bayes classifier (70%) for meronym relation.	F-Score for hypernym is less than expected
4	Detection the Automatic Speech Recognition (ASR) errors in Chinese Language	Using Dependency Parsing and n-gram model to detect the error	More efficient than only n-gram model	Max efficiency achieved is still not considerable and needs some improvement, it can't correct the error
5	Highly accurate syntactic analysis	Novel feature set, feature back-off strategy	No overlap with semi-supervised parsing methods, efficient than previous methods	
6	Weakness of unigram and	Novel approach	Better handling of	High computational

	bigram models for topic detection and tracking	called Dependency Structure Language Model using Chow expansion theory and linguistic parser	long-distance dependencies, more efficient and effective than previous methods	cost $O(n^3)$
7	Solving feature sparseness in Dependency Parsing	Parsing of sentences with baseline system, feature embedding, inferring feature embedding	Improved performance when tested on English and Chinese Languages	Results are unknown for other languages like Japanese
8	Information Extraction in biomedical protein- protein interactions	Tests were done on a custom hand annotated corpus for Link Grammar and Connexor Machinese Syntax	Link Grammar and Connexor Machinese Syntax evaluated, failure sources identified and improvements proposed	Results are unknown for machine annotated corpus
9	Higher sparseness in Discriminative methods used in NLP	Frequency bucketing	High accuracy on English and Chinese data	Smaller data was used for tests
10	Identification of coordinate word pairs in Chinese Language	Utilizing seeds to find new rules for unlabelled data, generating candidate coordinate word pairs and use them to Dependency Parser model	Simple and effective approach	No re-use of final word pairs as seed words
11	Parsing languages having small treebank	Using Partial Parser joined with Dependency Parser for self-training, using Malt Parser and MST Parser with Partial Parser for co-training	Performance improvement, accurate for Hindi Parsing	Results are unknown for Telugu and Bangla.
12	Evaluation of Statistical Constraint Based Dependency Parser	The 20k open vocabulary DARPA Hub1 WSJ CSR is used to evaluate SCDG parser-based LM	Reduction in Word Error Rate and more accuracy than other Constraint based parsers	Accuracy is still not considerable
13	Comparing human and machine annotators for Greek Language	Comparisons of errors using Spearman's Rank correlation	Overlap of mistakes show that we can combine two different methods	Test done only on the Greek Language
14	Text mining for protein phosphorylation	Simplification of complex sentences using Dependency Parsing, Pattern derivation from Dependency Tree	Efficient, accurate, can be applied to large texts in biology literature	No test done for entire text of literature
15	Optimizing Dependency Parsing Throughput	Replacing infrequent word	Four fold improvement with	Not good UAS score for other languages

		with 'unknown' label, Feature Encoding in 64 bits	no cost of accuracy, Best UAS for English	for
16	Poor Statistical Machine Translation from Malayalam to English	Rearrangement of English sentences to Malayalam	Good for Languages which follow SOV order (Indian Languages) , reduces the use of parsing tools	Only for languages which follow SOV order
17	Performing Sentiment Analysis using Dependency Parsing	Using Dependency Parsing for sentiment propagation rules, Context based model to tune sentiments, Using SentVis for implementation	Detection of outliers, Efficient on the simple sentences of English	Conditional statements cannot be analysed,no results available for other languages
18	Exploration of effect of statistical parser for Telugu parsing	Using MSTParser, Malt Parser, ZPar ,TurboParser and Easy-First Parser with different feature settings	Test language was Telugu – morphologically rich language	Small size of Telugu Treebank
19	Generating Parser for knowledge acquisition in Malay Language	Using existing parsers for English along with English-Malay parallel corpus	New possibilities are developed for Malay Parsers	Model is only proposed , not created in practice
20	Automation of the AMR (Abstract Meaning Representation) for information retrieval	Developed an editor for AMR, Use of Dependency Parser	80% accuracy on simple sentences of the English	Tests on the complex sentences are unknown, dependent on the English

3. SCOPE OF RESEARCH

AMR automation is yet not made for the complex sentences [20]. Construction of the Chinese Language Parser having high accuracy is still a poorly addressed question. Chinese Language is very complex and different from the other languages. Dependency Re-ranking System Proposed above [5] is still not combined with the other methods. The effect of the methods proposed [11] on the Bangla and Telugu languages which have scarce resources has not been studied. In addition to that, more advanced strategies need to be used as a selection criteria. The alternate clustering algorithm could be combined with method proposed [9]. Enhancing of the parser [9] by utilising semi-supervised methods is yet not done. Parsers for Malayalam are not still much efficient [16]. Morphologically rich languages are difficult to parse and very interesting work is going on them. Different parsers having complementary features are combined to increase the accuracy.

REFERENCES

[1] M. Kumar and M. Dua, "Adapting Stanford Parser's Dependencies to Paninian Grammar's Karaka relations using VerbNet," Procedia Computer Science, no. 58, pp. 363-370, 2015.

[2] M. Savic´, G. Rakic´, Z. Budimac and M. Ivanovic, "A language-independent approach to the extraction of dependencies between source code entities," Information and Software Technology, no. 56, pp. 1268-1288, 2014.

[3] S. N, S. M. Jasminea and S. Joseph, "Automatic Extraction of Hypernym & Meronym Relations in English Sentences Using Dependency Parser," Procedia Computer Science, no. 93, pp. 539-546, 2016.

[4] W. Z. YUAN, Z. MIAO, W. ZHU and W. LIU, "Detecting errors in Chinese spoken dialog system using ngram and dependency parsing," ICWMMN Proceedings, 2008.

[5] M. Shen, D. Kawahara and S. Kurohashi, "Dependency Parse Reranking with Rich Subtree Features," IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, vol. 22, no. 7, 2014.

[6] C. Lee, G. G. Lee and M. Jang, "Dependency structure language model for topic detection and tracking," Information Processing and Management, no. 43, pp. 1249-1259, 2007.

[7] W. Chen, M. Zhang and Y. Zhang, "Distributed Feature Representations for Dependency Parsing," IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, vol. 23, no. 3, 2015.

[8] S. Pyysalo, F. Ginter, T. Pahikkala, J. Boberg, J. J¨arvinen and T. Salakoski, "Evaluation of two dependency parsers on biomedical corpus targeted at protein—protein interactions," International Journal of Medical Informatics, no. 75, pp. 430-442, 2006.

[9] W. Chen, M. Zhang, Y. Zhang and X. Duan, "Exploiting meta features for dependency parsing and part-of-speech tagging," Artificial Intelligence, no. 230, p. 173–191, 2016.

[10] J. Yu and W. Chen, "Extracting Coordinate Word Pairs for Dependency Parsing," Collaborative Innovation Center of Novel Software Technology and Industrialization, Suzhou, China.

[11] R. Goutam, "Exploring self-training and co-training for Hindi Dependency Parsing using partial parses," in International Conference on Asian Language Processing, Hyderabad, India, 2012.

[12] W. Wang and M. P.Harper, LANGUAGE MODELING USING A STATISTICAL DEPENDENCY GRAMMAR PARSER, IEEE, 2003.

[13] S. Majidi and G. Crane, "Human and Machine Error Analysis on Dependency Parsing of Ancient Greek Texts," in IEEE, 2014.

[14] M. Wang, H. Xia, D. Sun, Z. Chen, M. Wang and A. Li, "Literature mining of protein phosphorylation using dependency parse trees," Methods, no. 67, pp. 386-393, 2014.

[15] A. Weichselbraun and N. S¨usstrunk, Optimizing Dependency Parsing Throughput, Chur, Switzerland: Department of Information, University of Applied Sciences Chur.

[16] Rahul.C, Dinunath.K, R. Ravindran and K.P.Soman, "Rule Based Reordering and Morphological Processing For English-Malayalam Statistical Machine Translation," in IEEE, Coimbatore, 2009.

[17] L. D. Caro and M. Grella, "Sentiment analysis via dependency parsing," Computer Standards & Interfaces , no. 35, p. 442–453, 2013.

[18] B. V. S. Kumari and R. R. Rao, "Telugu dependency parsing using different statistical parsers," Journal of King Saud University – Computer and Information Sciences, 2015.

[19] H. Mohamed, N. Omar, M. J. A. Aziz and S. A. Rahman, "Statistical Malay Dependency Parser for Knowledge Acquisition Based on Word Dependency Relation," Procedia - Social and Behavioral Sciences , no. 27, p. 188 – 193, 2011.

[20] P. N. Pelja, R. P, S. G. M and R. Binu, "Automatic AMR Generation for Simple Sentences Using Dependency Parser," Procedia Technology, no. 24, pp. 1528-1533, 2016.