King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Telugu dependency parsing using different statistical parsers

B. Venkata Seshu Kumari [a,*], Ramisetty Rajeshwara Rao [b,1]

[a] *JNTUH, Hyderabad, Telangana, India*
[b] *Computer Science & Engineering, JNTU Kakinada, Andhra Pradesh, India*

Received 14 September 2014; revised 21 November 2014; accepted 24 December 2014
Available online 3 November 2015

KEYWORDS

Dependency parsing;
Telugu;
MSTParser;
MaltParser;
TurboParser;
ZPar

Abstract In this paper we explore different statistical dependency parsers for parsing Telugu. We consider five popular dependency parsers namely, MaltParser, MSTParser, TurboParser, ZPar and Easy-First Parser. We experiment with different parser and feature settings and show the impact of different settings. We also provide a detailed analysis of the performance of all the parsers on major dependency labels. We report our results on test data of Telugu dependency treebank provided in the ICON 2010 tools contest on Indian languages dependency parsing. We obtain state-of-the art performance of 91.8% in unlabeled attachment score and 70.0% in labeled attachment score. To the best of our knowledge ours is the only work which explored all the five popular dependency parsers and compared the performance under different feature settings for Telugu.

1. Introduction

Dependency parsing is the task of uncovering the dependency tree of a sentence, which consists of labeled links representing dependency relationships between words. Parsing is useful in major NLP applications like Machine Translation, Dialogue Systems, Question Answering, etc. This led to the development of grammar-driven, data-driven and hybrid parsers. Due to the availability of annotated corpora in recent years, data driven parsing has achieved considerable success. The availability of phrase structure treebank for English (Marcus et al., 1993) has seen the development of many efficient parsers.

Unlike English, many Indian (Hindi, Bangla, Telugu, etc.) languages are free-word-order and are also morphologically rich. It has been suggested that free-word-order languages can be handled better using the dependency based framework than the constituency based one Bharati et al. (1995). Due to the availability of dependency treebanks, there are several recent attempts at building dependency parsers. Two CoNLL shared tasks (Buchholz and Marsi, 2006; Nivre et al., 2007a) were held aiming at building state-of-the-art dependency parsers for different languages. Recently in two ICON tools contests (Husain, 2009; Husain et al., 2010), and Coling 2012 Hindi parsing shared task (Bharati et al., 2012), rule-based, constraint based, statistical and hybrid approaches were explored towards building dependency parsers for three Indian

[*] Corresponding author. Tel.: +91 9989308242.
E-mail addresses: venkataseshukumari@gmail.com (B.V.S. Kumari), rajaraob4u@gmail.com (R.R. Rao).
[1] Tel.: +91 9959559456.
Peer review under responsibility of King Saud University.

languages namely, Telugu, Hindi and Bangla. In all these efforts, state-of-the-art accuracies are obtained by the popular data-driven parsers namely, MaltParser (Nivre et al., 2007b) and MSTParser (McDonald, 2006).

Among Indian languages, though there has been significant amount of work on dependency parsing Hindi, there is very little work on parsing Telugu. Most of the work in ICON 2010 tools contest for parsing Telugu used MaltParser. In this paper, we consider five popular dependency parsers, MaltParser, MSTParser, TurboParser, ZPar and Easy-First Parser. We provide related work in Section 2 and details of dependency parsing, Telugu language and the Telugu dependency treebank in Section 3. In Section 4, we experiment with different parser and feature settings and show the impact of different settings. Section 5 provides a detailed analysis of the performance of all the parsers on major dependency labels. We conclude with possible future directions in Section 6. We obtain state-of-the art performance of 91.8% in unlabeled attachment score and 70.0% in labeled attachment score. To the best of our knowledge ours is the only work which explored all the five popular dependency parsers and compared the performance under different feature settings for Telugu.

2. Related work

There has been significant amount of work on dependency parsing in the recent past. Though majority of the work is done on English language, there has been increasing interest in parsing other languages. CoNLL 2006 and 2007 Shared tasks (Buchholz and Marsi, 2006; Nivre et al., 2007a) introduced the task of multi-lingual dependency parsing. Different approaches were explored in these two shared tasks for parsing eighteen different languages: Arabic, Basque, Catalan, Chinese, Czech, Danish, Dutch, English, German, Greek, Hungarian, Italian, Japanese, Portuguese, Slovene, Spanish, Swedish, and Turkish. Three metrices: labeled attachment score (LAS), unlabeled attachment score (UAS) and label accuracy (LA) were used for evaluation. LAS is the percentage of tokens with both correct dependency head and correct dependency label. UAS is the percentage of tokens with correct dependency head and LA is the percentage of tokens with correct dependency label. Different techniques like data-driven vs. hybrid; single step vs. two-stage; transition based vs. graph based were explored. In all these efforts, state-of-the-art accuracies are obtained by MaltParser (Nivre et al., 2007b), a transition based parser and MSTParser (McDonald, 2006) a graph based parser.

Following CoNLL shared tasks, there were two ICON tools contests (Husain, 2009; Husain et al., 2010) aimed at parsing three Indian languages: Hindi, Telugu and Bangla. Different rule-based, constraint based, statistical and hybrid approaches were explored towards building dependency parsers. Kesidi et al. (2010) used a constraint based approach. The scoring function for ranking the base parsers is inspired by a graph based parsing model and labeling. Nivre (2009), Ambati et al. (2009) and Kosaraju et al. (2010) used MaltParser and explored the effectiveness of local morphosyntactic features, chunk features and automatic semantic information. Parser settings in terms of different algorithms and features were also explored. Ambati et al. (2009) explored the usefulness of MSTParser for parsing Indian languages. Zeman (2009) combined various well known dependency parsers forming a super parser by using a voting method.

Recently in Coling 2012 workshop on Machine Translation and Parsing in Indian Languages, Hindi parsing shared task was held with the latest Hindi dependency treebank (Bharati et al., 2012). In this shared task, in addition to experimenting with individual parsers, there were efforts at combining different parsers. McDonald and Nivre (2007) showed that MaltParser and MSTParser make different kinds of errors and combining both the parsers can result in better parsing performance. Following this idea, Kumari and Rao (2012) combined the output of MaltParser and MSTParser in an intuitive manner to extract pros of both the parsers. Kukkadapu et al. (2012) explored voting and blending techniques for parsing Hindi using MaltParser, MSTParser and TurboParser.

In this work, we explore five popular dependency parsers namely MaltParser (Nivre et al., 2007b), MSTParser (McDonald, 2006), TurboParser (Martins et al., 2009), ZPar (Zhang and Clark, 2011) and Easy-First Parser (Goldberg and Elhadad, 2010). MaltParser is a transition based parser whereas MSTParser is a graph based parser. TurboParser is also a graph based parser but uses integer linear programming technique for parsing in contrast to MSTParser which uses maximum spanning tree algorithms. Zpar is a shift-reduce parser similar to MaltParser but uses beam search unlike greedy search used by MaltParser. MaltParser and Zpar parse a sentence from left to right. But, Easy-First Parser use non-directional strategy for parsing where easier dependencies are resolved first and use them as features while resolving harder dependencies.

In addition to standard English Penn treebank data (Marcus et al., 1993), all these parsers were explored for CoNLL shared task data. Though average number of tokens in test data is around 5000 tokens, number of tokens in the training data varied from around 30 thousand tokens (Slovene) to 1.2 million tokens (Czech). Apart from the amount of training data, morphological richness and free word order nature posed greater challenges for the parsers. It has been observed that parsing performance is least for morphologically rich and/or free word order languages like Arabic, Turkish, etc (Buchholz and Marsi, 2006; Nivre et al., 2007a).

MaltParser and MSTParser are the two parsers which are widely explored in the dependency parsing literature. Though MaltParser is explored extensively for Telugu in ICON shared tasks, there is very little work on experimenting with MSTParser for Telugu. Kukkadapu et al. (2012) adapted TurboParser for Hindi but there is no work on adapting it for Telugu. There has been some work on exploring Zpar and Easy-First Parser for languages other than English (Zhang and Nivre, 2012; Goldberg and Elhadad, 2010). There is no work on adapting these parsers for Indian languages in general and Telugu in particular. So, ours is the first work on exploring TurboParser, Zpar and Easy-First Parser for Telugu. Also, most of the papers compare MaltParser, MSTParser and one of the TurboParser or Zpar or Easy-First parsers but not all of them. To the best of our knowledge, ours is the only work which explored all the five popular dependency parsers and compared their performance for any language in general and Telugu in particular.