11-712: NLP Lab Report

Rajarshi Das

Tuesday 29th April, 2014

Abstract

This is a report on the development of an open source dependency parser for the language, Bengali. Presently I have reported some basic information about the language.

The goal of this project is to design, implement and evaluate a dependency parser for the language, Bengali (also my native language). This language is characterized by a rich system of inflections, derivation and compound formation (Chakroborty, 2003; Saha et al., 2004) which makes analysis and generation of Bengali, a challenging task (Ghosh et al., 2009).

1 Basic Information about Bengali

According to (Lewis, 2013), Bengali is an eastern Indo-Aryan Language and is native to the region of eastern south Asia. It is the official language of Bangladesh and is also spoken in the Indian state of West Bengal and parts of Tripura and Assam.

Bengali follows the SOV order in terms of ordering of subject, object and verb (Dasgupta, 2003). It makes use of postpositions instead of prepositions. Determiners follow the noun while numerals, adjectives and possessors precede the noun. It exhibits no case or number agreement and no grammatical gender phenomena (Dasgupta, 2003). Nouns and pronouns are declined into four cases - nominative, objective, genitive and locative (Bhattacharya, 2001)

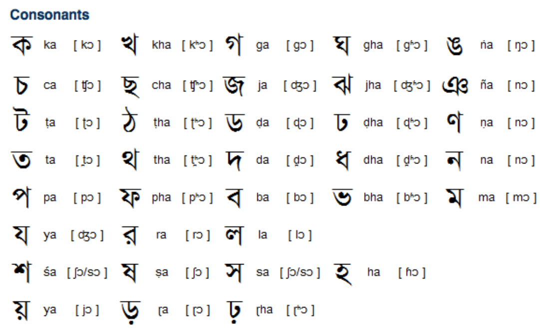
Bengali is written using the Bengali script. It has 11 vowel graphemes and 39 graphemes representing consonants and other modifiers. The script is written and read horizontally from left to right. Figure 1 and 2 show the vowels (and its various diacritics) and consonants in the Bengali script (Image source: Internet).

Figure 1: Vowels and vowel diacritics in Bengali script.

2 Past work on Bengali dependency parsing

Some work has been done in building dependency parsers for Bengali. (Ghosh et al., 2009) have used a statistical CRF based model followed by a rule based post processing technique. (Nivre,

Figure 2: Consonants in Bengali script.



2009), (Ambati et al., 2009) used a transition based dependency parsing model based on MaltParser (Nivre and Hall, 2005). (De et al., 2009) uses a hybrid approach where they simplify the complex and compound sentential structures and then recombine the parses of the simpler structure by satisfying the demands of the verb groups. (Abhilash and Mannem, 2010) use a bidirectional parser with perceptron learning with rich context as features. (Kosaraju et al., 2010) used Maltparser and explored the effectiveness of local morphosyntactic features chunk features and automatic semantic information. (Attardi et al., 2010) used a transition based dependency shift reduce parser which used a Multi layer Perceptron classifier. They were all tested on the same dataset as a part of a shared task held at ICON 2009 and 2010. (Hussain, 2009; Hussain et al., 2010). In the 2009 contest, (Ambati et al., 2009) system performed the best and in 2010, best score of Unlabeled Attachment Accuracy was achieved by (Attardi et al., 2010) and the best scores for Label Accuracy and Labeled Attachment was achieved by (Kosaraju et al., 2010).

3 Existing useful resources for the task

Microsoft Research India has a POS tagged dataset for several Indian languages including Bengali. The bengali dataset has 899 POS tagged sentences. Also I have been able to gain access to the annotated dataset which was used in the shared task held at ICON 2009 and 2010. Although I am aware that I cannot use the annotated dataset, I am hopeful that it will provide important insights for annotation.

4 Attested phenomena in the language

As mentioned earlier Bengali, like many Indian Languages is a free word order language. There has been an annotation effort for dependency parsing in Bengali in the past as a part of the shared task held at ICON 2009 and 2010. The data was annotated using the computational Paninian Grammar (Bharati et al., 1995). The paninian grammatical model treats a sentence as a series of modifier-modified elements starting from a primary modified (the root of the tree - generally the main verb) (Bharati et al., 2009). Also in (Bharati et al., 2009) and (Begum et al., 2010), they have catalogued in detail all the annotation rules. I am planning to follow the same rules just to be consistent, so that my annotations can be reused by researchers. Although the Paninan theory was formulated by Panini (a grammarian from Ancient India) 2500 years ago for the language Sanskrit, it is basically a dependency grammar (Kiparsky and Staal, 1969; Shastri, 1973). The framework is inspired by a inflectionally rich language such as Sanskrit and gives a strong framework for annotating for other Indian Languages. Also, although (Bharati et al., 2009) has been written as a guideline for

Corpus	#sentences	avg. #tokens / sentence
Test A	84	11.96
Test B	97	10.34
Train	571	9.56

Table 1: Corpus Statstics

annotating Hindi treebank, similar rules should apply to Bengali, because of the similarity in the languages.

5 Initial Design

For the test corpora, I have chosen a dataset of transcribed text of a speech corpus (Das et al., 2011). The text is a short story and are basically conversations between people. Some statistics about the corpus are shown in Table 1. Since both Test A and Test B are from the same story, we shouldn't expect much difference in results between the two corpuses. The primary reason I selected this dataset is because the data was POS tagged and that helped me a lot while annotating the data.The text is in ITRANS format (http://en.wikipedia.org/wiki/ITRANS) instead of the native bengali script. ITRANS is an ASCII transliteration scheme for Indic scripts. There are free software packages available which converts between this format and the native bengali script. Many of the other NLP tools which I used also uses data in this format so it was easy to integrate. The training set is also a collection of short stories in bengali. This dataset was also manually POS tagged.

Next, I am going to present some of my annotation decisions. Since I don't have much experience in linguistics, some of my annotation decisions might not be correct, but for the purpose of this project I have tried to remain consistent all throughout. Here are some of the decisions.

1. Multi-word name or proper nouns: In this case, I have made the last word of the name as the root of the chunk and the tree is a linear chain with the first word being the leaf. For example, the proper noun, Mr. Ramesh Singh would become Mr ← Ramesh ← Singh.

2. In Bengali, sometimes adverbs are repeated in order to stress something. In this case we again form a linear chain as above. This time the root is the first occurrence of the adverb. For example.
   Aami Satyi Satyi Bolchi
   (I) (truly) (truly) (saying)....
   In this case the two adverbs jointly form a chunk with the root as the second adverb.

3. Negative particles: In many cases, negative words normally group with the verb to change the sentence. In Bengali, the negative word usually is after the verb. I have annotated this chunk with the verb as the parent and the negative word as the child. For example.

   <u>Karte habe</u> na.
   (do) (no)

   In this example, the last word (na) is a negative word and is the child of the verb to the left.

Model	Accuracy (%)
Basic Model	**57.11**
Standard Model	55.59
Full Model	55.27

Table 2: First round of evaluation on Corpus A. Training size of just 500 tokens

4. In many cases, Bengali has a lot of multi verb expression (verbs occurring together to express the same thing). In such cases I have also annotated the dependency as a linear chain with the head as the first occurring verb.

5. In Bengali, adjectives generally precede the noun. I have made the adjective as the child of the noun which it is modifying.

6. Similar rule as above for adverbs with verbs. Sometimes the adverb and the corresponding verb are separated by few words. But in all cases the adverb is a child of the verb as a root.

7. Conjunctions, which occur between two noun phrases, I have made it as the head of the two noun phrase chunks. For example,

   Tamera saÑge Chele bA meYe ....
   (Tom) (with) (son) (or) (daughter)

   In this case the conjunction 'bA' is the head of 'Chele' and 'meYe'.

8. For sentences which start with a conjunction, I have made the conjunction as a direct child of the root of the parse tree.

9. Bengali is characterized by 'postpositions' in contrast to prepositions in English. Postpositions are made as the direct child of the nouns and pronouns it modifies (usually the noun/pronoun it follows)

10. The root of the sentence is usually the main verb.

For the strategy of implementation, I am thinking of doing a mix of semi-supervised and rule based methods.

6 First Round of evaluation - System Analysis of corpus A

I have been able to annotate around **2500** tokens in total. Separating 2000 tokens for test data, I had around 500 tokens left for training. I used the Turbo Parser (Martins et al., 2010) to train the parser. The features which I used to train the parser are POS tags (coarse and fine). The coarse POS tags are basically the first character of the POS tags.

The unlabeled attachment scores after training the basic, standard and full model are listed in table 2.

Model	Accuracy (%)
Basic Model + 4023 tokens	**62.32**
Standard Model + 4023 tokens	60.48
Full Model + 4023 tokens	60.91
Basic Model + 6033 tokens	**63.63**
Standard Model + 6033 tokens	60.48
Full Model + 6033 tokens	63.19

Table 3: Second round of evaluation on Corpus A.

7 Lessons learned and Revised Design

As we can see, the unlabeled accuracies are not that high. This is primarily because of the small size of the training set. Also for the same reason, the basic model is performing better than the standard and the full model. For the next steps, I plan to annotate a lot more data to get meaningful results so that I can think of incorporating other features. Some of the features which I am planning to incorporate are morphological features and also do some unsupervised clustering. But right now, the priority is to annotate more training data.

7.1 Performance on more training data

In the few weeks, I have been able to annotate more data for training. Similar features as above were used (coarse and full POS tags). I did annotation in couple of batches. In the first batch, I annotated a total of around **4000** tokens and at the second round of effort, I was able to annotate upto **6000** tokens. Table 3 shows the unlabeled attachment scores.

The unlabeled accuracy scores are respectable and much better than the initial round of evaluation with just 500 tokens. It is clear that the parser was able to learn better with more training examples. Also the difference in performance of the three models has decreased suggesting that the size of the training set is meaningful to train a standard or full model. Infact for 5300 tokens, the full model performs better than the standard model. Also the difference in accuracies for the two training sets is not much suggesting that we have to use new features to have a better parser. I am planning to add some morphological features to see if there is an increase in the accuracy.

As planned earlier, I have incorporated some morphological features of the language. I have used the Bengali morphological analyzer (Sarkar) made available by the researchers at Indian Institute of Technology Kharagpur. On manual analysis, the performance of the morphological analyzer looked decent. Although there were many morphological features produced as output by the morphological analyzer, I used the root/lemma of a given word as one of the feature. Table 4 shows the unlabeled accuracy on test corpus A.

Morphological features indeed helped!. The accuracy of the parser increased a bit on adding information about the root of each word. This was interesting to observe. Although the performance of the standard model remained the same, the accuracy of both the basic and full model increased.

Model	Accuracy (%)
Basic Model + lemma	**64.17**
Standard Model + lemma	60.15
Full Model + lemma	62.11
Basic Model + l + n + p	60.91
Standard Model + l + n + p	63.30
Full Model + l + n + p	**64.28**

Table 4: Unlabeled accuracy on Test Corpus A. Morphological features used for training.

Model	Accuracy (%)
Basic Model + 4023 tokens	58.36
Standard Model + 4023 tokens	58.14
Full Model + 4023 tokens	**59.25**
Basic Model + 6033 tokens	60.91
Standard Model + 6033 tokens	60.47
Full Model + 6033 tokens	**63.01**

Table 5: Unlabeled accuracy on Test Corpus B.

8 System Analysis of Corpus B

Table 5 shows the unlabeled accuracy for test corpus B on both the training datasets. On the larger training set, the system performs rather poorly with the highest accuracy of 59.25% achieved by the full model. As the parser is trained on the larger corpus, the accuracy increased (60.91% by the full model) though it still performs poorly as compared to test corpus A.

Model	Accuracy (%)
Basic Model + lemma	61.68
Standard Model + lemma	62.13
Full Model + lemma	**63.68**
Basic Model + l + n + p	60.47
Standard Model + l + n + p	60.58
Full Model + l + n + p	**60.91**

Table 6: Unlabeled accuracy on Test Corpus B. Morphological features used for training.

References

A Abhilash and Prashant Mannem. Bi directional dependency parser for indian languages. In *Proceedings of ICON 2010 tools contest on Indian Language Dependency Parsing*, Kharagpur, India, 2010.

B R Ambati, P. Gadde, and Jindal K. Experiments in indian language dependency parsing. In *Proceedings of ICON 2009 tools contest on Indian Language Dependency Parsing*, Hyderabad, India, 2009.

G Attardi, S.D. Rossi, and M Simi. Dependency parsing of indian languages with desr. In *Proceedings of ICON 2010 tools contest on Indian Language Dependency Parsing*, Kharagpur, India, 2010.

R Begum, S Husain, A Dhwaj, D. M. Sharma, Bai L., and Sangal R. Dependency annotation scheme for indian languages. In *Proceedings of the third international joint conference on Natural Language Processing*, Hyderabad, India, 2010.

A Bharati, V Chitanya, and R. Sangal. Natural language processing: A panninian perspective. Prentice-Hall of India, New Delhi, 1995.

A Bharati, M. Sharma, S. Husain, Bai L., R. Begam, and Sangal R. Anncora: Treebanks for indian languages, guidelines for annotating hindi treebank. 2009.

Tanmoy Bhattacharya. Bengali. In Jane Garry and Carl Rubino, editors, *Facts About the World's Languages: An Encyclopedia of the World's Major Languages: Past and Present*. H.W. Wilson Press, New York/Dublin, 2001.

Bamondeb Chakroborty. Uchchotoro bangla byakaron. *Akshay Malancha*, 2003.

Biswajit Das, Sandipan Mandal, and Pabitra Mitra. Bengali speech corpus for continuous automatic speech recognition system. In *Proceedings of peech Database and Assessments (Oriental COCOSDA)*, 2011.

Probal Dasgupta. Bangla. In George Cardona and Dhanesh Jain, editors, *The Indo-Aryan Languages*, pages 351–390. Routledge, London, 2003.

S De, A Dhar, P Bhaskar, and U Garain. Structure simplification and demand satisfaction. In *Proceedings of ICON 2009 tools contest on Indian Language Dependency Parsing*, Hyderabad, India, 2009.

Aniruddha Ghosh, A Das, P Bhaskar, and Sivaji Bandyopadhyay. Dependency parser for bengali: the ju system at icon 2009. *NLP Tool Contest ICON*, 2009.

S. Hussain. Dependency parser for indian languages. In *Proceedings of ICON 2009 tools contest on Indian Language Dependency Parsing*, Hyderabad, India, 2009.

S Hussain, Prashant Mannem, Bharat Ambati, and Phani Gadde. The icon 2010 tools contest on indian language dependency parsing. In *Proceedings of ICON 2010 tools contest on Indian Language Dependency Parsing*, Kharagpur, India, 2010.

P. Kiparsky and J. F. Staal. Synctactic and relations in panini. In *Foundations of language 5*, 1969.

P Kosaraju, S.R. Kesidi, Ainavolu V.B.R., and Kukkadapu P. Experiments on indian language dependency parsing. In *Proceedings of ICON 2010 tools contest on Indian Language Dependency Parsing*, Kharagpur, India, 2010.

M. Paul Lewis. Ethnologue: Languages of the world, 2013. URL `http://www.ethnologue.com/`.

André F. T. Martins, Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. Turbo parsers: Dependency parsing by approximate variational inference. In *Proceedings EMNLP*, pages 34–44, 2010.

Joakim Nivre. Parsing indian languages with maltparser, 2009.

Joakim Nivre and Johan Hall. Maltparser: A language-independent system for data-driven dependency parsing. In *In Proc. of the Fourth Workshop on Treebanks and Linguistic Theories*, pages

13–95, 2005.

Goutam Kumar Saha, Amiya Baran Saha, and Sudipto Debnath. Computer assisted bangla words pos tagging. In *Proceedings of the International Symposium on Machine Translation NLP and TSS (iTRANS)*, 2004.

Sudeshna Sarkar. Morphological analyzer engine for bengali. URL `http://nltr.org/snltr-software/`.

C. Shastri. Vyakarana chandrodaya. In *Foundations of language 5*, 1973.