11-712: NLP Lab Report

Rajarshi Das

Friday 2nd May, 2014

Abstract

This is a report on the development of an open source dependency parser for the language, Bengali. I have been able to annotate a total of around 7500 tokens (around 5500 tokens for train set and 1000 tokens for each test set). The genre of the text were short stories. I have used a supervised learning approach with morphosyntactic features. The system achieves the highest accuracy of **64.39%** on test corpus A and **66.56%** on test corpus B.

Caveat: I have used a parts-of-speech (POS) tagged dataset for my experiments. To the best of my knowledge, the data was tagged manually and hence would be highly accurate when compared to the output of an automatic POS tagger. The results of the system would be different from what I have reported, if an automatic POS tagger is used instead.

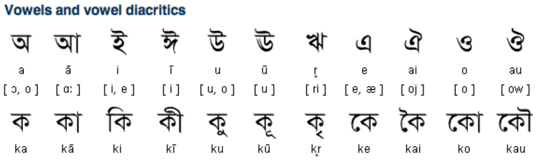
The goal of this project is to design, implement and evaluate a dependency parser for the language, Bengali (also my native language). This language is characterized by a rich system of inflections, derivation and compound formation (Chakroborty, 2003; Saha et al., 2004) which makes analysis and generation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



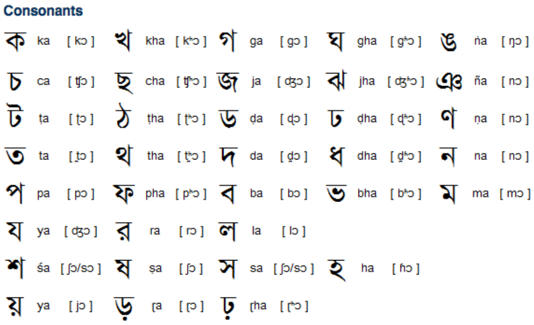
Vowels and vowel diacritics

অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
a	ā	i	ī	u	ū	r̥	e	ai	o	au
[ɔ, o]	[ɑː]	[i, e]	[i]	[u, o]	[u]	[ri]	[e, æ]	[oj]	[o]	[ow]
ক	কা	কি	কী	কু	কূ	কৃ	কে	কৈ	কো	কৌ
ka	kā	ki	kī	ku	kū	kr̥	ke	kai	ko	kau

2 Past work on syntax of Bengali

Bengali is a well studied language. Several work on analyzing morphology (Dasgupta et al.), parts-of-speech tagging (Dandapat et al., 2007) already exists for the language in the literature. Some work

Figure 2: Consonants in Bengali script.



Consonants

ক	ka	[kɔ]	খ	kha	[kʰɔ]	গ	ga	[gɔ]	ঘ	gha	[gʱɔ]	ঙ	ṅa	[ŋɔ]
চ	ca	[tʃɔ]	ছ	cha	[tʃʰɔ]	জ	ja	[dʒɔ]	ঝ	jha	[dʒʱɔ]	ঞ	ña	[nɔ]
ট	ṭa	[ʈɔ]	ঠ	ṭha	[ʈʰɔ]	ড	ḍa	[ɖɔ]	ঢ	ḍha	[ɖʱɔ]	ণ	ṇa	[nɔ]
ত	ta	[t̪ɔ]	থ	tha	[t̪ʰɔ]	দ	da	[d̪ɔ]	ধ	dha	[d̪ʱɔ]	ন	na	[nɔ]
প	pa	[pɔ]	ফ	pha	[pʰɔ]	ব	ba	[bɔ]	ভ	bha	[bʱɔ]	ম	ma	[mɔ]
য	ya	[dʒɔ]	র	ra	[rɔ]	ল	la	[lɔ]						
শ	śa	[ʃɔ/sɔ]	ষ	ṣa	[ʃɔ]	স	sa	[ʃɔ/sɔ]	হ	ha	[ɦɔ]			
য়	ya	[jɔ]	ড়	ṛa	[ɽɔ]	ঢ়	ṛha	[ɽʰɔ]						

has been done in building dependency parsers for Bengali. During the year 2009 and 2010, there was a shared task on dependency parsing of Indian languages at ICON (International Conference of Natural Language Procesing). Bengali was one of the languages of the shared task. (Ghosh et al., 2009) have used a statistical CRF based model followed by a rule based post processing technique. (Nivre, 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

For this task, I am planning to use a dataset which was manually POS tagged. This data set was primarily used to train a statistical POS tagger. Since we are required to annotate for dependency parsing ourselves, I have decided to annotate using the manually POS tagged dataset because I think that would make the task of annotation easier. I collected the dataset from the POS tagger which is available at (`http://nltr.org/snltr-software/`). The dataset is a collection of short stories (with a total of around 40,000 tokens). I am also planning to use one of the freely available morphological analyzer if I plan to incorporate morphological features.

4 Attested phenomena in the language

Bengali, like many Indian Languages is a free word order language. 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).

Corpus	#sentences	avg. #tokens / sentence
Test A	84	11.96
Test B	97	10.34
Train	571	10.56

Table 1: Corpus Statstics

5 Initial Design

For the test corpora, I have chosen one of the short story from the dataset. The short story is a conversations between few people and is characterized by lots of direct speeches and short sentences. The training corpus were also a collection of different short stories. Table 1 shows some statistics of the test and the train set. Since both Test A and Test B are from the same short story, there are a lot of over lapping words and therefore I expect the system's performance to be similar on both the test corpus. Also, 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.

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. For doing the annotation, I have used the tool DG Annotator by Giuseppe Attardi (`http://medialab.di.unipi.it/Project/QA/Parser/DgAnnotator/`)

Annotation decisions

1. Multi-word names: For multi-word proper nouns, 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. Sometimes adverbs/adjectives 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.

   Karte habe na.
   (do) (no)

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

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 direct child of the verb as a root.

7. Conjunctions which occur between two noun phrases have been made 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. Common nouns which occur together have also been annotated as a linear chain with the first noun as the head of the chunk. (This is a decision which I think is not grammatically sound, since I have faced some sentences where this decision was not applicable. In such cases, I have annotated the sentences in the way I felt it was correct, but I couldn't come up with a general rule for doing so)

11. The root of the sentence is usually the main verb of the sentence or a conjunction which connects two shorter sentences.

After annotation is complete, I am planning to do supervised learning to see how the system performs. I also plan to try some unsupervised approaches such as Brown Clustering (Brown et al., 1992)

6 System Analysis of corpus A

6.1 First round of evaluation

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 first alphabet of the POS tags are the coarse POS tags.

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

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

6.2 Performance on more training data and features

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 **3500** tokens and at the second round of effort, I was able to annotate upto **5500** 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. On the smaller set of training examples, I got the best accuracy of **62.32 %**. The performance of the full and the standard model were comparable. On the larger training corpus, the highest accuracy was **63.63%** by the basic model. Also the accuracy of the full model increased to **63.19%**, suggesting that with more training examples, it is overfitting less to the training data. On just adding more training data, the accuracy increased quite a lot. I am hoping that on Test corpus B, it is going to perform equally well or better since both the corpuses and similar and the avg. length of sentences in test corpus B is shorter as compared to corpus A. Next, I plan to try adding some more morphological features.

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 (eye-balling), the performance of the morphological analyzer looked decent. Although there were many morphological features produced as output by the morphological analyzer, I first added the *root/lemma* of a given word as one of the feature. Table 4 (upper-half) shows the unlabeled accuracy on test corpus A. From now all the results which I will report, will all be on the larger training corpora (5463 tokens) and will have the POS (coarse and fine) tags as the feature by default. On adding lemma of the word, the performance improved!. The highest accuracy of the parser on corpus A, increased to **64.17%** from 63.6% by the basic model. I am a little worried that the performance of the Full model, since the performance decreased by 1%. But since the over all performance of the parser increased, I think this feature helped.

Next I added, the *number* and *person* features with lemma. Table 4 (lower-half) shows the unlabeled accuracy on test corpus A. Interestingly, the performance of the basic model decreased to **60.91%**. Also the accuracy of the standard and full model increased. The overall performance increased to **64.28%**. After performing this experiment, in retrospect I realized that Bengali is a language which doesn't have any number/gender agreement. That might be one of the reason the performance of the basic model decreased so much. It does have personal pronouns and may be thats why the full and the standard model is able to learn from it.

In the next experiment, I threw away the number and the person features and added the *tense* of the verbs. So the features now were POS tags, lemma and tense. Table 5 (upper-half) reports

Model	Accuracy (%)
Basic Model + 3641 tokens	**62.32**
Standard Model + 3641 tokens	60.48
Full Model + 3641 tokens	60.91
Basic Model + 5463 tokens	**63.63**
Standard Model + 5463 tokens	60.48
Full Model + 5463 tokens	63.19

Table 3: Second round of evaluation on Corpus A.

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.

the unlabeled accuracy. The performance of the basic model decreased (although not as bad as the previous experiment). The performance of the standard and the full model were moderate.

After these experiments, I wasn't sure which features were working the best. (I am still skeptical about the performance of the number and person features). So i thought I will include all features upto now and perform the experiments. Table 5 (lower-half) reports the unlabeled accuracy. The basic model performs the best with an accuracy of **64.39%**. This is also the highest accuracy received over all.

Unsupervised methods I thought it might be a good idea to do some sort of unsupervised clustering of the words. Basically that would enable me to have a coarser representation of the words. I used brown clustering (Brown et al., 1992) which clusters words together having the same context. I used Percy Liang's implementation of brown clustering (https://github.com/percyliang/brown-cluster). I experimented with 100 clusters on my whole unannotated dataset (42,203 tokens). The feature which I used was the cluster number of the words. The additional features which I used are POS, lemma, tenses. Unfortunately I didn't get any improvements. I report the results in table 6. There wasn't any improvements in the result (although there wasn't much decrease, which I think is because of the morph features). Since the clustering approach did not look promising, I did not try any other combinations.

7 Lessons learned and Revised Design

The primary lessons learned from experimenting with corpus A are that more training examples improved the performance of the parser consistently. I would annotate more if I get more time, because I think it will still improve the accuracy. Also with more data the full model performs equally good as the basic model (probably will start performing better on more training data). Also adding morphological features was a good idea. I am still not sure which morphological

Model	Accuracy (%)
Basic Model + l + tense	**63.74**
Standard Model + l + tense	62.43
Full Model + l + tense	62.32
Basic Model + l + n + p + t	**64.39**
Standard Model + l + n + p + t	62.11
Full Model + l + n + p + t	63.19

Table 5: Unlabeled accuracy on Test Corpus A. Morphological features used are lemma and tenses of the verbs.

Model	Accuracy (%)
Basic Model + l + t + brown	**63.95**
Standard Model + l + t + brown	61.89
Full Model + l + t + brown	62.43

Table 6: Accuracy on corpus A using brown clustering and some morph features (lemma, tense)

feature is having the most difference. Since the test corpus B is similar to corpus A, I would expect similar results in them too.

8 System Analysis of Corpus B

I ran the same bunch of experiments on corpus B. I will report the results and my analysis for each experiments here. The first experiment was by just adding the POS tags. The results are reported in table 7 .On the smaller training corpus (3641 tokens), the highest accuracy achieved is **63.34%** by the full model (compared to 62.32% on corpus A). Its nice to see the full model outperforming the other models. On the larger training corpus, the results improve even further. The highest accuracy achieved is **65.45%** again by the full model. This is significantly higher than what was achieved on the full model (63.63%).

Next I experimented with the various morph features. On adding just the *lemma* of the tokens, the accuracy of all the models increased!. The highest accuracy achieved here was **66.45%** by the full model and accuracy of the basic and the standard model also increased (64.34 from 63.68 and 64.78 from 62.79) respectively. This confirms my hypothesis that morphological feature lemma was performing well.

The next experiment was to add *number* and *person* features. The results are reported in table 8 (lower-half). It is clear that adding the number and person features is decreasing performance. It supports my intuition and although this wasn't clear from the results of corpus A, it is pretty clear now. Because of the absence of number feature in bengali, the parser is not able to learn anything new or probably learning wrong things.

Like for test corpus A, I eliminated the number and person features and incorporated the tense feature. The results are reported in table 9 (upper-half). This improved the performance even more. The accuracy of the full model further increased to **66.56%**. This means the morph features

Model	Accuracy (%)
Basic Model + 3641 tokens	61.35
Standard Model + 3641 tokens	61.90
Full Model + 3641 tokens	**63.34**
Basic Model + 5463 tokens	63.68
Standard Model + 5463 tokens	62.79
Full Model + 5463 tokens	**65.45**

Table 7: Unlabeled accuracy on Test Corpus B.

Model	Accuracy (%)
Basic Model + lemma	64.34
Standard Model + lemma	64.78
Full Model + lemma	**66.45**
Basic Model + l + n + p	63.23
Standard Model + l + n + p	63.01
Full Model + l + n + p	**63.79**

Table 8: Unlabeled accuracy on Test Corpus B. Morphological features used for training.

lemma and *tense* helps!.

On experimenting with every feature, the performance of the corpus decreased!. Clearly the number and person feature isn't working well. The results are reported in 9 (lower-half).

9 Final Revision

Since I was annotating a linguistic task for the first time, I realized that some of the decisions which I made might not be correct and was not consistent with some of my later decisions during annotations of the training phase. So I thought I will go through the test corpus and re-annotate to be consistent with my annotations in the training set. I am reporting the results separately because according to the guidelines we were supposed to annotate the test corpus first. I report the best results which I get with each experiment for corpus A in table 10 and for corpus B in table 11.

Revising the annotations was a good idea and it is clear that it increased the accuracy. The highest accuracy achieved on Corpus A increased to **66.23%** (from 64%). Also this was achieved by just using the lemma and POS tags as the features (this time it outperformed the number and person feature combination). Also accuracy for test corpus B also increased to **67.66 %** (from 66.56%). This was achieved by the full model with lemma, POS and tense as the feature. Over all, the performances follow my prior intuition.

10 Conclusion and Future work

In this work, I have annotated dependency parse tree for around 7500 tokens (2000 for test and 5500 for training) for the language Bengali. The genre of the data was mostly short stories. I have come up with my annotation rules and have reported them. I have tried to be consistent with them,

Model	Accuracy (%)
Basic Model + l + tense	64.12
Standard Model + l + tense	65.67
Full Model + l + tense	**66.56**
Basic Model + l + n + p + t	**63.46**
Standard Model + l + n + p + t	62.35
Full Model + l + n + p + t	62.90

Table 9: Unlabeled accuracy on Test Corpus B. In the upper half of the tables, morphological features lemma and tenses are used. In the lower-half all morphological features are used.

Model	Accuracy (%)
Basic Model + POS	65.36
Basic Model + POS + lemma	**66.23**
Full Model + l + n + p	65.80
Basic Model + l + tense	65.58
Full Model + + l + n + p + t	65.36

Table 10: Accuracy on Corpus A after revised annotation on test corpus

Model	Accuracy (%)
Full Model + POS	66.33
Full Model + POS + lemma	67.22
Full Model + l + n + p	64.23
Full Model + l + tense	**67.66**
Basic Model + + l + n + p + t	64.12

Table 11: Accuracy on Corpus B after revised annotation on test corpus

but some of these rules might not be grammatically sound. For parsing, I have used supervised approaches and have experimented using various combinations of morphological features and POS tags. The most helpful features were POS tags and *lemma* of the tokens. The highest accuracy achieved on corpus A is **64.39%** and that on corpus B is **66.56%**. On revising the annotations for test corpus, the accuracy increased to **66.23%** for corpus A and **67.66%** for corpus B.

The over all accuracy achieved by my system isn't very high and I think it can be improved by adding more training data. Also I am not very confident about the quality of my annotations and I think the performance would significantly improve if the system receives high quality annotations. Also as a future work, I would like to implement a variant of semi-supervised learning. Annotating data is very labor-intensive task but there is no dearth of unannotated data. So if we could use use the un-annotated data to improve our system, that would be great. It would be interesting to see the results of self training (testing our model on unannotated dataset and incorporating them as training data). These are some of the future extensions I would be interested in doing for improving my system.

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.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, 1992.

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

Sandipan Dandapat, Sudeshna Sarkar, and Anupam Basu. Automatic part-of-speech tagging for bengali: An approach for morphologically rich languages in a poor resource scenario. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 221–224, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1557769.1557833`.

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.

Sajib Dasgupta, Naira Khan, Asif Iqbal Sarkar, Dewan Shahriar, Hossain Pavel, and Mumit Khan. Morphological analysis of inflecting compound words in bangla.

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.