11-712: NLP Lab Report

Pradeep Prabakar Ravindran

April 26, 2013 (Final Due date)

Abstract

First draft of a report documenting my experience developing a dependency parser for Tamil.

TamilDep is a tool that performs dependency parsing of the Tamil Language. This tool would soon be open-sourced and become publicly available at https://github.com/pradeep-gnr/TamilDep. Along with the code for a dependency parser, a corpora of tamil sentences annotated with their dependency parse trees is also expected to be released. Since there are very few resources publicly available for Tamil, I hope that this software would prove useful to researchers working on Tamil natural language processing.

1 Basic Information about Tamil

Tamil is a Dravidian language that is widely spoken in the Indian state of Tamil Nadu and the north eastern part of Sri Lanka. Reports estimate that there are about 70 million native Tamil Speakers all over the world (Wikipedia (2013)). In terms of history, Tamil is also one of the oldest languages in the world with works of Tamil literature dating back to almost 2000 years ago (Wikipedia (2013)).

Grammatically, Tamil has a lot of characteristics that make it pretty different from the Germanic languages. For example, Tamil almost always follows a subject-object-verb oriented form (Ramasamy and Žabokrtskỳ (2011)). For instance, consider the following sentence.

Avan	angu	selgiran
(He)	(there)	(going)

The sentence approximately translates to 'He is going there'. Sentences where inflected verb forms occurring at the end of the sentence are relatively common in Tamil. Also, similar to languages like Czech, Tamil is relatively a free word order language. For example, the sentence that was discussed above could be expressed in multiple forms such as

angu	avan	selgiran
(There)	(he)	(going)

selgiran	avan	angu
(going)	(he)	(there)

All of these sentences are perfectly valid in terms of grammatical structure. Another key characteristic of Tamil sentences is that words could be agglutinative and sometimes it is very difficult to even define what a word is. There are a number of ways in which suffixes could be added to the noun and verb stems to produce highly inflected forms.

For instance, 'has gotten over' which is a phrase in English could be expressed as a single word in Tamil. Also, there could be multiple such words that would approximately translate to the sample phrase. Some such examples are discussed below.

```
1) Nadanthathu
2) Nadanthathuvittathu
3) Nadanthayittru
```

All these words express the phrase 'has gotten over' and they are all highly inflected forms of the root word 'Nada' which roughly translates to 'happening' or 'proceeding'. Since these type of verb phrases are relatively common in Tamil, identifying the 'root' of the head words of such phrases would be a challenge from a dependency parsing perspective.

2 Past Work on the Syntax and NLP tools for Tamil

One of the most infleuntial books on Tamil grammar written in English was by Thomas Lehmann : A grammar of modern Tamil (Lehmann et al. (1989)). The book gives a very detailed analysis of several of the morphological and syntactic phenomenon that characterize written Tamil. While languages like English do not differ significantly from their spoken and written form, there are significant differences among the spoken and written forms in Tamil. Tamil has the property that sentences could be semantically valid even when certain words are omitted and spoken Tamil usually consists of a lot of missing words. Written Tamil however is more morphologically richer than its spoken version. Lehmann also notes that Tamil is relatively a head final language and that the phrasal head appears in several inflected forms at the end of the phrase (Lehmann et al. (1989)). Also, another key characteristic of Tamil is that the verbs agree with gender, tense etc. One of the key differences in tamil syntax when compared to other languages is that the language exhibits a restricted free word order. Hence there are a multitude of semantically and syntactically valid sentences that could be formed using the same set of words. From a parsing perspective - this property poses a lot of challenges because unlike English, writing context free grammar rules to characterize Tamil Syntax is a non-trivial problem. (Srinivasan CJ (2007)) discuss the challenges they face because of this issue while building a spoken dialog system for Tamil and turn to dependency parsing to aid semantic annotation of sentences.

Unlike other Indian languages like Hindi, Telugu and Bengali - Tamil does not have a lot of annotated corpora for NLP tasks. The Tamil dependency treebank ((Ramasamy and Žabokrtsk<span>ỳ</span> (2011))) which was released recently is the only publicly available dependency corpora that I could find online. The authors have also released an annotation manual that describes the rules that were used in labeling dependency edges among words. The entire corpus was annotated in the Prague dependency treebank (PDT) (Böhmová et al. (2003)) format and a detailed description of this corpora and the annotation methodology that was used will be discussed in part 3. Although this corpus is quite small (about 3000 words), the annotations are quite diverse and cover various types of sentences in Tamil. The same group also used this corpus to build a rule based and corpus based dependency parser and report a 75 % accuracy when sentences are labeled with POS tags.

There was also an earlier paper (Dhanalakshmi and Rajendran (2010)) that used morphological analysis along with various custom heuristics for syntactic parsing of Tamil to identify phrasal consituents. Besides the work done by Ramaswamy et al (Ramasamy and Žabokrtskỳ (2011)), I could find only one related paper that describes a set of Natural language tools that were created specifically for Tamil. They have also released an open source Morphological analyzer for Tamil but unfortunately there was no dependency tree annotated corpora or software that was available for download. To the best of my knowledge, TamilDep would be the first open source dependency parser written specifically for Tamil.

3 Available Resources for Tamil

First of all to build a good annotated corpus, we need good quality sentences in the target language. The source text must be of high quality both in grammar and content. The content must also be diverse in the number of topics covered to make the parser more robust while handling different types of inputs. Tamil has a separate Wikipedia portal where there are quite a significant of documents that have been written exclusively in Tamil. There are also several other corpora that have been publicly released for various Tamil NLP tasks such as English-Tamil machine translation (Ramasamy et al. (2012)), Tamil Wordnet etc (Rajendran et al.).

The English-Tamil parallel corpora released by (Ramasamy and Žabokrtskỳ (2011)) consists of a total of around 169871 sentences that have been crawled from various news articles discussing multiple topics. From this corpus, I have extracted random sentences and constructed corpora A and B. Corpora A consists of about 70 sentences with 1126 tokens and corpora B contains 70 sentences with 1185 tokens. Additionally I have also created corpora C, which could be used as training data if I decide to pursue a supervised approach to dependency parsing. Corpora C consists of about 210 sentences and 3500 tokens. The data has been uploaded in the corpora folder in the root repository.

To aid the dependency annotation process, I decided to extend on the work done by (Ramasamy and Žabokrtskỳ (2011)) In their earlier paper on dependency parsing experiments for Tamil, (Ramasamy and Žabokrtskỳ (2011)) describe their experiences creating a detailed annotation corpora for Tamil. The annotations are in the Prague treebank format that has a three level layer based annotation at the syntactic and lexical level. They have also publicly released their annotation manual and rules to guide dependency annotation for Tamil. This manual is very comprehensive and documents several phenomena that is unique to Tamil and issues that a person needs to consider while performing the annotation. Because of the highly inflected nature of the language, morphological analysis must also be performed prior to syntactic annotation. The manual also discusses strategies for morphological analysis and outlines the rules in detail so that it could be easily extended into a morphological analyzer. I decided to use the custom tag set that was used by (Ramasamy et al. (2011)) in the hope that future annotated corpora produced for Tamil would conform to a particular standard and the research community can evaluate their methodologies on different corpora. Also, this would encourage the development of supervised models which tend to work well when more data is available.

4 Survey of Phenomena in Tamil

In this section, I shall describe some of the interesting phenomena that Tamil part of speech tags exhibit. Lehmann (Lehmann et al. (1989)) notes that Tamil has about eight part of speech tags (Nouns, Verbs, Postpositions, Adjectives, Adverbs, Quantifiers, Determiners, and Conjunctions). Each of these part of POS tags exhibit some interesting behavior that help characterize Tamil language.

4.1 Nouns

Lehmann states that nouns in Tamil are those words that can take case suffixes and also the suffixes (aay, aaka). The suffixes (aay, aaka) are adverbial suffixes that could be applied to several nouns. Lehmann also notes that not all nouns could take these suffixes and these nouns are called as defective nouns. Nouns in Tamil could also be inflected in a number of ways according to case and number. An inflected noun is usually of the form noun + (number) + case. The most common forms of suffixes that could inflect Tamil nouns are given below.

1. plural suffix
2. oblique suffix
3. euphonic suffix
4. case suffix

Noun stems in Tamil are the stems of nouns as they would be listed in a dictionary. Tamil noun stems could both be simple and complex. Complex nouns are however formed by combining a root and a derivational suffix. For example, paal (milk) is a simple noun having no suffixes while pati + ppu (study) is an example of a complex noun that has a verbal root (padi) and a suffix attached to it. Both these types are relatively common in Tamil. Tamil also exhibits a class of noun stem called as oblique stem that has no meaning when it exists by itself but can combine with case suffixes and post positions to produce various noun forms.

4.2 Pronouns

Tamil has the interesting phenomena that verbs with gender and case markings can stand out as separate sentences even when there is no explicit subject. As I discussed in the earlier sections, verb phrases in Tamil inflected with pronoun markers are relatively frequent in Tamil. For example, the verb 'walk' can be inflected as pronouns

1. Nadanthan - (He walked)
2. Nadandhal - (She walked)
3. Nadandhadhu - (It walked)

4.3 Postpositions

Postpositions are very important in Tamil and occur very frequently after nouns resulting in complex inflected phrases. Lehmann notes that postpositions could be inflected or un-inflected nouns or they could also be non-finite verb forms. Lehmann also discusses 8 different scenarios in which postpositions occur and I have described some of them below.

1. Occurring after nouns (Nominative)- (kaadu + varai - (till the forest)))
2. After nouns (accusative case) - (padippai + patri - (About your education))
3. Occurring after nouns (dative case) - (Idharku + paatilaga (Instead of this))

4.4 Adjectives

Lehmann notes that adjectives in Tamil usually are of two distinct types: simple and derived. Some examples of simple and derived adjectives are given below.

1. ketta paiyyan (bad boy)
2. vayathana aasiriyar (old teacher) - Here the adjective 'vayathana' consists of a noun stem 'vayathu' (age) and the suffix 'aana'.

   Derived adjectives are usually obtained by adding suffixes like 'aana' etc to nouns.

5 Initial Design

To perform the annotation, I have tried to follow the rules that were discussed in the annotation manual by Ramaswamy et al (Ramasamy et al. (2011)). The manual consists of several example sentences that cover a wide range of phenomena which was useful for me in making some decisions. One interesting phenomena, was that several Tamil sentences contained english words in between. To make my parser more robust, I have also considered such sentences. In most of the examples that I annotated, the head word mostly occured at the end of the sentence. The annotated corpora A anb B have been uploaded in the repository. Each of them contain around 1000 tokens. Currently, I have just added the head information for each term in CONLL format. I have also included the text of the string in addition to the annotations. But as of now, I have not extracted any features such as POS tags and morphological information to the output. I could find only one open source POS tagger for Tamil (Dhanalakshmi and Rajendran (2010)) and I was unable to get the software to work. But by the next deadline, I think that I would be able to use the POS tag features as input for my model. Also, I am aware that features based on morphological analysis would be immensely useful for highly inflected languages such as Tamil, but again I was unable to get the morphological analyzer that was built by the same group (Dhanalakshmi and Rajendran (2010)) to work. I hope to resolve some dependency issues in the software to extract POS tags and morphological features before building my model. My idea is to build a supervised dependency parser by annotating some additional training data. I feel that supervised approaches would work well because I think more training data would help capture the wider range of morphological phenomena that occurs in Tamil words. I am currently annotating more data for the supervised model and I plan to use MaltParser for training the model. My eventual goal would be to build a stable data driven supervised dependency parser guided by a rich feature set such as POS tags and morphological annotations.

6 System Analysis on Corpus A

For the baseline system, I have used a supervised learning approach where I had annotated additional 2000 tokens as training data. I found a Tamil chunker tool released by the Language Technologies Research centre at IIIT Hyderabad that performs various pre-processing tasks for Tamil (`http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php`). For example, it can automatically convert various english letters occuring in Tamil to an equivalent Tamil script. Also, it has an inbuilt POS tagger and a morphological analyzer in addition to a tokenizer. For the baseline, I just used the part of speech tags as features in the MALT parser. This baseline would give me a vague idea of how good/bad my approach is. Although, I have just used limited training data for the baseline - I am planning to annotate more training data and use roughly about

7000 tokens for training.

I used the standard settings prescribed in the Malt Parser and the unlabelled attachment score accuracy that I got was **43%**. The performance of the model is relatively poor and it was something that I had anticipated. The training data that I used was relatively small for the testing data that I used. In the next iteration, I plan to make the following changes.

1. Annotate more training data of about 7000 tokens.
2. Revise POS annotations and try to formulate a sparser POS representation because the existing POS tag set that I am using is quite large.
3. Add morphological features to MALT Parser.
4. Experiment with various parameter settings in MALTParser.

7 Lessons Learned and Revised Design

8 System Analysis on Corpus B

9 Final Revisions

10 Future Work

References

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. The prague dependency treebank. In *Treebanks*, pages 103–127. Springer, 2003.

V Dhanalakshmi and S Rajendran. Natural language processing tools for tamil grammar learning and teaching. *International journal of Computer Applications (0975-8887)*, 8(14), 2010.

T. Lehmann, Pondicherry Institute of Linguistics, and Culture. *A Grammar of Modern Tamil*. PILC publication. Pondicherry Institute of Linguistics and Culture, 1989. URL `http://books.google.com/books?id=THlkAAAAMAAJ`.

S Rajendran. Tamil wordnet.

Loganathan Ramasamy and Zdeněk Žabokrtskỳ. Tamil dependency parsing: results using rule based and corpus based approaches. In *Computational Linguistics and Intelligent Text Processing*, pages 82–95. Springer, 2011.

Loganathan Ramasamy, Ondřej Bojar, and Zdeněk Žabokrtský. Tamil dependency treebank (tamiltb) 0.1. annotation manual. Technical Report TR-2011-42, Institute of Formal and Applied Linguistics (FAL MFF UK), Faculty of Mathematics and Physics, Charles University, 2011.

Loganathan Ramasamy, Ondřej Bojar, and Zdeněk Žabokrtský. Morphological processing for english-tamil statistical machine translation. In *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012)*, pages 113–122, 2012.

Loganathan R Santhosh Kumar C Srinivasan CJ, Udhaykumar N. Robust dependency parser for natural language dialog systems in tamil. *5th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 1–6, 2007.

Wikipedia. Tamil language. 2013. URL `http://en.wikipedia.org/wiki/Tamil_language`.