MALAYALAM POS TAGGER AND CHUNKER

DEEPTHY ANTONY
CB.EN.P2CEN08007
OUTLINE

- Introduction
- POS Tagging
- AMRITA Tagset
- Corpus Development
- SVMTool
- Chunking
- Chunk Tagset
- Yamcha
- Results and GUI
- Conclusion
INTRODUCTION

• Part-of-speech (POS) tagging is the process of assigning POS tags to each and every word in a sentence.

• It is like assigning the grammatical category such as Noun, Verb, Adjective, Adverb etc.

• The next process after POS tagging is chunking, which divides sentences into non recursive inseparable Phrases. i.e. only one head in a phrase.
There are many tools available for POS tagging and Chunking.

We have used SVM based Tools for both Malayalam POS tagging and Chunking.

- POS Tagging
  - SVMTOOL
- Chunking
  - YAMCHA
What is PoS tagging?

- PoS tagging is the process of identifying and marking the words in the sentence with its Part of Speech (PoS) category.
PoS Tagger

- **INPUT**: a string of words (sentence)

- **OUTPUT**: a single best tag for each word (POS Tagged sentence)
Example

- Assign POS tags to words in a sentence.

She is a beautiful girl

അവള്‍ ഒരു സുന്ദരിയായ പെണ്കുട്ടി ആ കുന്നു

PRP  DET  ADJ  NN  VF
Why PoS Tagging?

- Needed for languages which have more than one PoS category for a word
- Most language processing and language understanding systems require sentences to be disambiguated from PoS categories
- Ex: Machine Translation, Dialog Systems, Speech synthesis and etc.
Lexical Ambiguity in PoS Tagging

- Assign POS tags to words in a sentence.

what can it do

அதே மிக்க கேடுக

PRP QW VINT
PRP QW NN
PoS Application

Part-of-speech (POS) tagging is important for many Applications

- Word sense disambiguation
- Parsing
- Language modeling
- Q&A and Information extraction
- Text-to-speech
- Machine translation
- Semantic tagging
- Dialogue tagging
- Information Retrieval....
AMRITA TAGSET

- Every natural language is different from each other. So there is a need for a separate tagset.

- We have used <AMRITA tagset> to tag our corpus.

- It is based on the 'category' of the word, does not consider the grammatical features of the word.
The tagset consists of 29 tags, where there are 5 tags for nouns, 1 tag for pronoun, 7 tags for verbs, 3 for punctuations, 2 for number, and 1 for each adjective, adverb, conjunction, echo, reduplication, intensifier, postposition, emphasize, determiners, complimentizer and question word.
<table>
<thead>
<tr>
<th>S.N.</th>
<th>TAG</th>
<th>DESCRIPTION</th>
<th>S.N.</th>
<th>TAG</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;NN&gt;</td>
<td>NOUN</td>
<td>16</td>
<td>&lt;CVB&gt;</td>
<td>CONDITIONAL VERB</td>
</tr>
<tr>
<td>2</td>
<td>&lt;NNC&gt;</td>
<td>COMPOUND NOUN</td>
<td>17</td>
<td>&lt;NNQ&gt;</td>
<td>QUANTITY NOUN</td>
</tr>
<tr>
<td>3</td>
<td>&lt;NNP&gt;</td>
<td>PROPER NOUN</td>
<td>18</td>
<td>&lt;INT&gt;</td>
<td>INTENSIFIER</td>
</tr>
<tr>
<td>4</td>
<td>&lt;NNPC&gt;</td>
<td>COMPOUND PROPER NOUN</td>
<td>19</td>
<td>&lt;DET&gt;</td>
<td>DETERMINERS</td>
</tr>
<tr>
<td>5</td>
<td>&lt;ORD&gt;</td>
<td>ORDINALS</td>
<td>20</td>
<td>&lt;PPO&gt;</td>
<td>POST POSITIONS</td>
</tr>
<tr>
<td>6</td>
<td>&lt;CRD&gt;</td>
<td>CARDINALS</td>
<td>21</td>
<td>&lt;EMP&gt;</td>
<td>EMPHASIS</td>
</tr>
<tr>
<td>7</td>
<td>&lt;PRP&gt;</td>
<td>PERSONAL PRONOUN</td>
<td>22</td>
<td>&lt;COM&gt;</td>
<td>COMPLIMENTIZER</td>
</tr>
<tr>
<td>8</td>
<td>&lt;ADJ&gt;</td>
<td>ADJECTIVE</td>
<td>23</td>
<td>&lt;CNJ&gt;</td>
<td>CONJUNCTION</td>
</tr>
<tr>
<td>9</td>
<td>&lt;ADV&gt;</td>
<td>ADVERB</td>
<td>24</td>
<td>&lt;QW&gt;</td>
<td>QUESTION WORDS</td>
</tr>
<tr>
<td>10</td>
<td>&lt;VNAJ&gt;</td>
<td>VERB NON FINITE</td>
<td>25</td>
<td>&lt;ECH&gt;</td>
<td>ECHO WORDS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADVERB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>&lt;VNAV&gt;</td>
<td>VERB NON FINITE ADVERB</td>
<td>26</td>
<td>&lt;RDW&gt;</td>
<td>REDUPLICATION WORDS</td>
</tr>
<tr>
<td>12</td>
<td>&lt;VINT&gt;</td>
<td>VERB INFINITE</td>
<td>27</td>
<td>&lt;COMM&gt;</td>
<td>COMMA</td>
</tr>
<tr>
<td>13</td>
<td>&lt;VAX&gt;</td>
<td>VERB AUXILARY</td>
<td>28</td>
<td>&lt;QM&gt;</td>
<td>QUESTION MARK</td>
</tr>
<tr>
<td>14</td>
<td>&lt;VBG&gt;</td>
<td>VERBAL GERUND</td>
<td>29</td>
<td>&lt;DOT&gt;</td>
<td>DOT</td>
</tr>
<tr>
<td>15</td>
<td>&lt;VF&gt;</td>
<td>VERB FINITE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Amrita Vishwa Vidyapeetham
Coimbatore.

26 AUGUST 2010   CEN
CORPUS DEVELOPMENT

- Pre-editing
- Manual Tagging
- Corpus Training
- Tagging using SVMTagger

- Corpus size: 1,00,250 words
Pre-editing (Tokenizer)

- Untagged sentences are downloaded from Malayalam newspaper and commercial website.
- We remove all special characters except dot, comma and question mark.
- After this, tokenize the data i.e. change input text into a column format. Since the input data to the SVM tool must be in form of token.
Manual Tagging

- After pre-editing we get a corpus of untagged tokens.
- Then we tagged the corpus manually using AMRITA tagset.
- We assign POS tag to each and every word in corpora.
- Initially we tagged around 20,000 words manually.
Corpus Training

- The tagged corpus is trained using SVM (SVMTLearn, component of SVM tool).
- After training, we get a dictionary with merged model and its feature set (Lexicon).
- Lexicon contains the Malayalam words with its group of tags probability parameters.
Tagging using SVM

- The pre-edited corpus is given to the SVM (SVMTagger, component of SVM tool) for tagging. The words in the Lexicon are tagged.

- A word that is currently not available in the lexicon may be tagged by the probability parameters, using the probability with the bias occurrence words.
• After tagging the output displayed is checked manually and the tags are corrected.

• Again the corpus training process is done to increase the size of lexicon.

• We had developed our corpus of around 1,00,250 pos tagged words.
SVM (Support Vector Machine)

- Support vector machine is a training algorithm for learning classification and regression rules from data.
- SVM is based on the idea of structural risk minimization, a principled technique for selecting a model which minimizes generalization error.
- SVM is increasingly being used in processing NLP tasks.
SVMTOOL

- This implementation is based on the principle of Support Vector Machines (SVM).

- Trains efficiently and solve real NLP problems like POS tagging and chunking.
Components of SVMTOOL

- SVMTlearn
- SVMTagger
- SVMTeval
• ഉൾപ്പെടെ സവണം എന്ത് ചേർക്കുന്നു?
• എനിക്കു എന്തെന്തുകാര്യം?
• കാരണം പുതിയ കാര്യം?
എന്ത് സവണം?

. 

cകക
. 

tകക
. 

മുകൾ
?
ഞങ്ങള്<PRP> പരിക്കുഴ്ത്തു<VAX>.
. <DOT>
പൊതു<ADJ>
കസർ<NN>
. <DOT>
വിക്കാദ<PRP>
മേള<QW>
വേവാടു<VF>
? <QM>
Algorithm for PoS Tagging

- Take input text.
- Tokenize the input text (Pre-editing).
- Manual Tagging.
- Train the corpus.
- Tagging using SVM.
  - Search for the tokens in lexicon.
  - If it found, give the appropriate TAG from lexicon.
  - If not found, TAG it with SVM probabilities.
- Get the tagged output text.
- Insert those new words in lexicon.
IMPLEMENTATION

Corpus

Tokenization

Tagging

Training

UnTagged words

SVMTagger

Tagged words
What is Chunking

- **Chunking** is the next step after POS tagging, which divides sentences into non recursive inseparable Phrases.
  
  i.e. only one head in a phrase.

- **Chunking**, is the task of identifying and labelling the simple phrases in a sentence from the tagged output.
Why Chunking?

- Chunking is considered as an intermediate step towards full parsing, and it finds application in speech recognition, information retrieval and machine translation.
Chunking in Malayalam

- Malayalam being an agglutinative language have a complex morphological and syntactical structure.

- It is a relatively free word order language but in the phrasal and clausal construction it behaves like a fixed word order language.

- The process of chunking in Malayalam is less complex compared to the process of POS tagging.
Chunker

- **Training**: Word sequences with corresponding POS and Chunk tags.

- **Input**: Word sequence and POS tags

- **Output**: A single best Chunk Tag for each word.
Example

- Assigning Chunk Tags to words in a sentences.

She is a beautiful girl

അവള്‍ ഒരു സുന്ദരിയായ പെണ്കുട്ടി

PRP DET ADJ NN VF

(B-NP)NP (B-NP I-NP I-NP)NP (B-VFP)VF
CHUNK TAGSET

- We have followed the guidelines given in (AnnCorra) for preparing our customized Chunk Tagset.

- We have defined 9 Chunk Tagset for Malayalam.
<table>
<thead>
<tr>
<th>S.N o</th>
<th>Chunk Tag</th>
<th>Tag Name</th>
<th>Possible POS Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NP</td>
<td>Noun Phrase</td>
<td>NN, NNP, NNPC, NNC, NNQ, PR, DET, CRD, ORD, ADJ, INT</td>
</tr>
<tr>
<td>2</td>
<td>AJP</td>
<td>Adjectival Phrase</td>
<td>CRD, ADJ</td>
</tr>
<tr>
<td>3</td>
<td>AVP</td>
<td>Adverbial Phrase</td>
<td>ADV, INT, CRD</td>
</tr>
<tr>
<td>4</td>
<td>VFP</td>
<td>Verb Finite Phrase</td>
<td>VF, VAX</td>
</tr>
<tr>
<td>5</td>
<td>VNP</td>
<td>Verb Nonfinite Phrase</td>
<td>VNAJ, VNAV, VINT, CVB</td>
</tr>
<tr>
<td>6</td>
<td>VGP</td>
<td>Verb Gerund Phrase</td>
<td>VBG</td>
</tr>
<tr>
<td>7</td>
<td>CJP</td>
<td>Conjunctional</td>
<td>CNJ</td>
</tr>
<tr>
<td>8</td>
<td>COMP</td>
<td>Complimentizer</td>
<td>COM</td>
</tr>
<tr>
<td>9</td>
<td>.?</td>
<td>Symbols</td>
<td>O</td>
</tr>
</tbody>
</table>
Chunk Tagset

- **IOB Tag:**
  
The IOB tags are used to indicate the boundaries for each chunk
  
- **B** – the current word is the beginning of a chunk, which may be followed by another chunk.
  
- **O** - indicates the boundary of the sentence.
  
- **I** – the current word is inside a chunk.
YAMCHA

- **YamCha** is a generic, customizable, and open source text chunker.

- **YamCha** is using a state-of-the-art machine learning algorithm called Support Vector Machines (SVMs), first introduced by Vapnik in 1995.
Training and Test File Format

- Both the training file and the test file need to be in a particular format for Yamcha to work properly.
- The training and test file must consist of multiple tokens.
- A token consists of multiple (but fixed-numbers) columns. The tokens are simply correspond to words. Each token must be represented in one line, with the columns separated by white space (spaces or tabular characters). A sequence of token becomes a sentence. To identify the boundary between sentences, an empty line is put.
Training and Test File Format

- We can give as many columns as we like, however the number of columns must be fixed through all tokens.

- There are some kinds of "semantics" among the columns. For example, First column is 'word', second column is 'POS tag' third column is ‘CHUNK tag' and so on.

- The last column represents a true answer tag which is going to be trained by Yamcha.
Training data

തലവേൾ <PRP> <B-NP>

പൊതികുളം <VAX> <B-VFP>

. <DOT> <O>

പൊതി <ADJ> <B-NP>

കസസര <NN> <I-NP>

. <DOT> <O>
Algorithm for Chunking

- Take the tokenized pos tagged text as input.
- Manual Tagging.
- Train the corpus.
- Tagging using Yamcha.
  - Search for the tokens in lexicon.
  - If it found, give the appropriate TAG from lexicon.
  - If not found, TAG it with respect to the probabilities.
- Get the tagged output text.
- Insert those new words in lexicon.
RESULT

• We have developed each corpus size of around 1,00,250 words for both Malayalam POS tagging and Chunking.

• Accuracy

  • Malayalam POS Tagging → 96.51 %
  • Malayalam Chunking → 96.85 %
RESULT

- [root@localhost bin]# perl SVMTeval tag 19.txt 19may.out
- SVMTool v1.3.1
- (C) 2009 TALP RESEARCH CENTER.
- Written by Jesus Gimenez and Lluis Marquez.
- SVMTool v1.3.1 (C) 2009 TALP RESEARCH CENTER.
- Written by Jesus Gimenez and Lluis Marquez.

- * ============= SVMTeval report==============
- * model               = [tag]
- * testset (gold)      = [19.txt]
- * testset (predicted) = [19may.out]
- EVALUATING <19may.out> vs. <19.txt> on model <tag>........2439 tokens [DONE]
- * ============= TAGGING SUMMARY=============
- #TOKENS           = 2439
- AVERAGE_AMBIGUITY = 13.2292 tags per token
- #KNOWN            = 53.0135% -- 1293 / 2439
- #UNKNOWN          = 46.9865% -- 1146 / 2439
- #AMBIGUOUS        = 6.5601% -- 160 / 2439
- #MFT baseline     = 51.6195% -- 1259 / 2439
- * ========= OVERALL ACCURACY ================
- HITS           TRIALS      ACCURACY    MFT
- * --------------------------------------------------
- 2354             2439       96.5150%    51.6195%
GUI

MALAYALAM POS TAGGER AND CHUNKER

INPUT SENTENCE

OUTPUT SENTENCE

Tamil POS Tagger Developed By CEN, Amrita Vishwa Vidyapeetham

Amrita Vishwa Vidyapeetham
Coimbatore.
GUI

MALAYALAM POS TAGGER AND CHUNKER

INPUT SENTENCE

OUTPUT SENTENCE

CHUNK PHRASED OUTPUT

CLEAR  CANCEL
CONCLUSION

- Both POS Tagging and Chunking plays an important role in various Natural language processing applications.
- These tagged corpus can be used for parsing which will provide important syntactic information for machine translation.
- Future possible work is to increase the corpus size.
REFERENCES


