Adhyann – A Hybrid Part-of-Speech Tagger

Nitigya Sharma, Nikki and Gopal Sahni

Department of Computer Science,Bharat Institute of Technology, Meerut (250004)

ABSTRACT

Part of Speech Tagging automatically tags the word of a text by labels that can be used to determine the structure of sentence. In this paper we propose an approach to the problem that is inspired from human behavior. We used a combination of rule based and dictionary based approach to tackle this problem. Our goal in this paper is to design a simple yet effective system to POS tagging that also helps us in more effective understanding of human behavior.

KEYWORDS

POS tagger, Natural language Processing, Rule Based Approach, Dictionary Based Approach.

1. INTRODUCTION

Every language has a set of tags for each word, these tags describes the role of words in a sentence.

A very basic approach to this problems is to use a database that will simply map each word to its corresponding tag. But this approach suffers from various problems, such as.

- New words are created every day, A database to store all these word will grow very fast, However only a small portion of the database will be actively used for tagging.
- Names are also Part of Speech, which are actively used while framing a sentence,
- However they are not supposed to be store in database. For instance "Mohan is hawaldaar**.**" in this sentence, "Mohan" is an Indian **name** and "hawaldaar" is an Indian designation.
- As single word can be mapped to various tags. For instance "**fly**", however only one tag is relevant, based on its role in the sentence. "**Fly**" is a verb if it precedes by "**to**" and it is a noun if it is preceded by "**a**"

Another approach to problem is to use rules that will decide the tag of word in a sentence based onits position with respect to other words. This approach as we observes suffer from following problems

- In rule based approach sentence is just a sequence of character that looks like "**XXXX XX XXX XXX**." in such type of sentence various possibilities lies for tagging.
- Even though if it identifies some tags than also this approach may fail. For instance two sentences "**My car is Black.**" and "**My car is BMW.**" have same structure and length,

DOI : 10.5121/ijit.2015.4301

but cannot be effectively tagged until system understand the difference between "**Black**" and "**BMW**".

Thus, we require a system that does not possess above limitation.

2. METHODALOGY ADOPTED

This System works on human like approach. When a human reads a sentence he identifies the part of speeches in the sentence to understand the meaning of the respective sentence. A sentence may contain various words some, of which are known to the reader and other are new to the reader. For the new words reader reads the complete sentence and tries to identify the proper Part of Speech Tag for the new word and then remembers that word unconsciously.

System also works on the same principle to achieve this objective. It uses two approaches

- Dictionary Based Approach
- Rule Based Approach

2.1.Dictionary Based Approach

In this Approach a direct mapping of each word is done with words already stored in the database. If word is found in database than it's respective tag is fetched and assigned to the word.

Each tag set has its own table and it contains word that belongs to that particular Tag. In this way words belonging to different Tags can be stored easily and separately.

Advantage of dictionary Based Approach :

- It has less chances of error.
- It can also Tag wrong sentences
- It can also tag sentences with ambiguity.

Disadvantage of Dictionary Based Approach :

- It require heavy database initially to work.
- unknown words are Tagged with noun,
- Some word has two or more possible tags.
- In ability to correctly Tags Name

2.2.Rule Based Approach

Rule Based Approach uses handwritten Grammar rules to tag a sentence with proper Tags. Rule based approach can efficiently tag unknown words using the sentence structure.
Some handwritten rules are

- a Noun Phrase consist of following sequence of words

- (Determinant)(Adverb)*(Adjective)(Noun).

- A Verb Phrase consist of (Verb)(Adverb)*.
- Prepositions are followed by Noun.
- Helping verb are followed by either a Verb phrase or a Noun Phrase.
- if Pronoun is possessive than it is followed by a Noun Phrase otherwise a Verb Phrase.
- Noun is never followed by “TO”.

Though each of the above rules has exceptions. And the rule based tagging requires that some of the tags must be known in advance.

Therefore the rule base tagging first guesses the tag for a word and then evaluates the rules to check if the guessed tag fits properly to the sentence or not.

Advantage of Rule based Tagging

- It can effectively remove ambiguous tags.
- It can tag the words which have never been encountered.
- It has the potential to tag almost any sentence.

Disadvantage of Rule Based Tagging

- It is slower than dictionary tagging
- It cannot tag ambiguous sentence such as “My car is _____”.
- It do not improves with data its answer is always fixed.

2.3.Hybrid approach

We combined both the approaches and made a system has the advantage of both the Rule based and dictionary based approach.

Initially we tag the words from the database and then apply rule based tagging on the semi-tagged sentence to fill other empty tags. Then we checks these tags for their syntactic validity and remove ambiguous tags.

After performing validity check we store newly tagged word in a Temporary Database a word moves to its respective tag table only if it is occurred twice a week.

2.3.1. Limitingthe Size of Vocabulary

To keep the size of database as small as possible system removes the words that are not used frequently.

System identifies those words by the equation for all words if (Today-Mentioned) >=28

$$\frac{Occurrence}{Today - Mentioned_{Date}} <= 0.28$$

Where

- Occurrence : No of time a word has occurred
- Today : Today's Date
- Mentioned_day : Date on which the word occurred first time

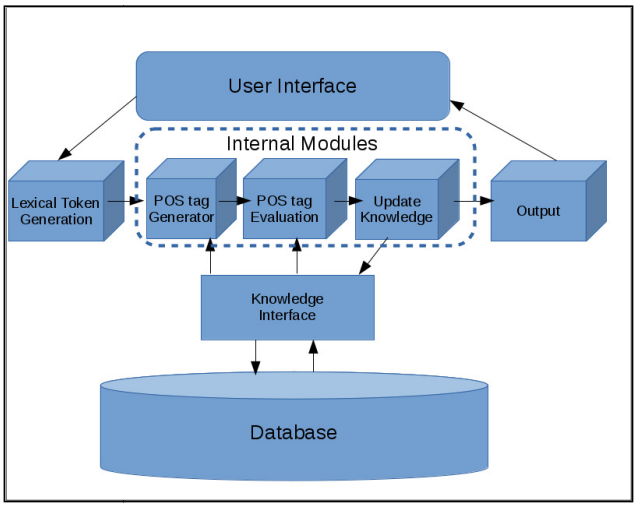
if after 4 week average occurrence is less than twice a week, than the word is removed from the database otherwise it is stored in its respective Tag.. Benefit of performing above operation is that system do not store Words with wrong tags, Names, Spelling mistakes, etc.

Advantage:

- It can effectively remove ambiguous tags.
- It can tag the words which have never been encountered.
- It has the potential to tag almost any sentence.
- It has less chances of error.
- It can also Tag wrong sentences
- It can also tag sentences with ambiguous structure.
-

3 SYSTEM ARCHITECTURE AND DESIGN:

3.1 The Underlying Model



The System follows a sequential Structure that assigns possible tag on the respective words of a sentence. The tool is divided into following main modules.

- **User interface**
- **Lexical Analyzer**
- **POS Tag Generator**
- **POS Tag Evaluator**
- **Knowledge**

3.1.1 User Interface Module:

This Module provides a GUI (Graphical User Interface) to use the system apart from GUI the system can also be operated through command line interface. Various functions provided by this module are :
Spell checker: It checks the spelling errors in real-time so that user can avoid mistakes.
File Chooser: It allows selecting a file containing text to be tagged.
Area to Display tag list : from this are a user can view tags assigned to word

3.1.2 Lexical Analyzer:

Lexical analyzer can divide a continuous string to an array of Token, where each token is a separate entity. It performs following operations on input String.

- Checks for the validity of String.
- Divide the String into various words.
- Assign Specific Type to word depending on its type, such as ATOM, MATH_LITERAL,COMM etc.

3.1.3 POS Tag Generator:

This module is the key module in this tool it decides the tags for a specific word, it do not check for their validity it only assign proper tag depending on their local behavior. It performs following operations in sequence:

- Dictionary Tagging: In this phase it fills the entire tokens that are stored in its knowledge already. For Example PRONOUN, HVERB, TO, CONJUNCTION, INTERJECTION
- Rule Based Tagging: in this step it fills tag on the basis of their behavior in the respective sentence. Rule based tagging is done in two steps.
- Filling Noun Phrase: in this step it fills all the word that can be assigned Noun Phrase tags NOUN, ADJECTIVE, ADVERB.
- Filling Verb Phrase in this step it fills all the word that can be assigned Verb Phrase tags such as VERB, ADVERB.

3.1.4 POS Tag Evaluator:

This is the last step in the tagging process; it checks the complete tag list for any specific contradiction, violation or ambiguity. Than it performs following steps on the Tagged list:

- Validation: in this step the tags are checked for syntactic validity and it removes

any particular tag that violates it.

- Ambiguity removal: In this step if any word has more than one cardinality that the tags of the respective word are checked the one which do not fit the scenario is removed.
- Dictionary validation: after above steps if any tag causes ambiguity than it is removed if it is not in the knowledge of system.

3.1.5 Knowledge:

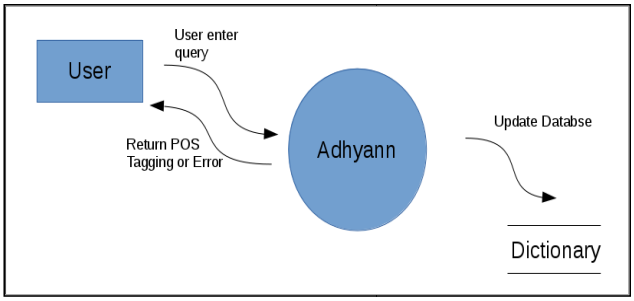
This module defines a way to insert and retrieve data from database. It performs following functionality:

- Insert data into database;
- Retrieval of data from database;
- Deciding correct Part of Speech tag to be inserted in database..
- Removing data which is not used frequently

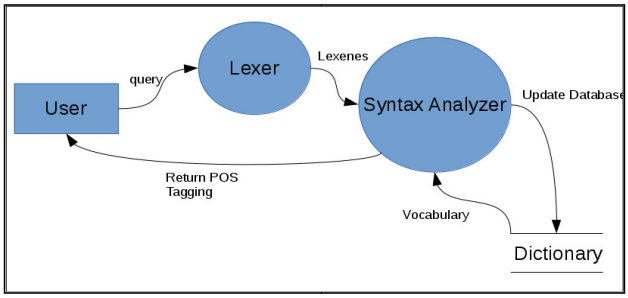
3.2 System Design

Data Flow Diagram:

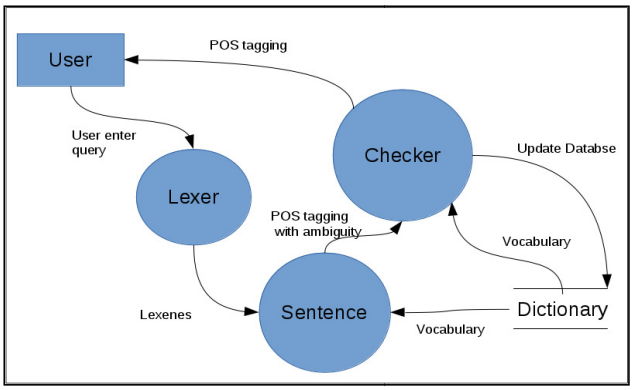
Level 0



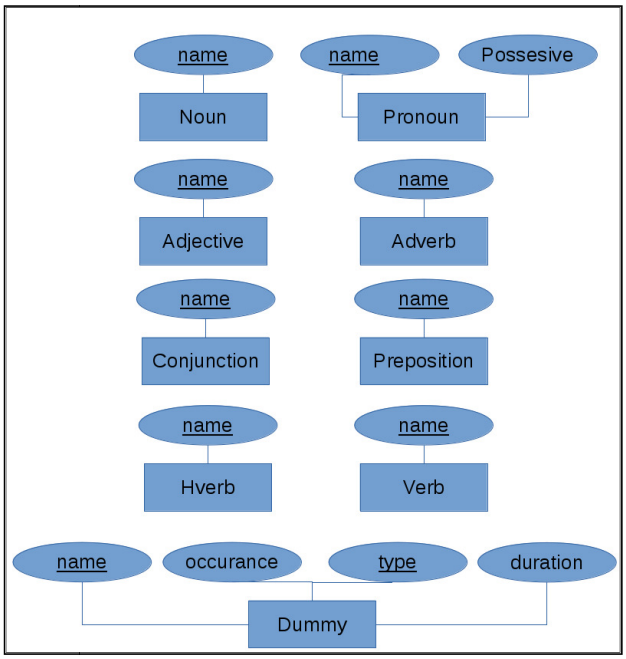
Level 1



Level 2



ER Diagram



4 RESULTS

4.1 Example 1

String entered is

i was born to fly.

<!--------Lexer----------->

Lexene generated

[<ATOM(i) [NULL]>, <ATOM(was) [NULL]>, <ATOM(born) [NULL]>, <ATOM(to) [NULL]>, <ATOM(fly) [NULL]><PERIOD>]

<!--------Lexer----------->

<!-------POS Tag Generation-------->

Complete dictionary tagging

[<ATOM(i) [NULL, PRONOUN]><ATOM(was) [NULL, HVERB]><ATOM(born) [NULL]><ATOM(to) [NULL, TO, ADVERB]><ATOM(fly) [NULL,

ADVERB]><PERIOD>]

Filling Noun Phrase

[<ATOM(i) [NULL, PRONOUN]><ATOM(was) [NULL, HVERB]><ATOM(born) [NULL, NOUN]><ATOM(to) [NULL, TO, ADVERB]><ATOM(fly) [NULL, ADVERB]><PERIOD>]

Filling Verb Phrase

[<ATOM(i) [NULL, PRONOUN]><ATOM(was) [NULL, HVERB, VERB]><ATOM(born) [NULL, NOUN, ADVERB, VERB]><ATOM(to) [NULL, TO, ADVERB, VERB]><ATOM(fly) [NULL, ADVERB, VERB]><PERIOD>]

Intermediate filling

[<ATOM(i) [NULL, PRONOUN, NOUN]>, <ATOM(was) [NULL, HVERB, VERB]>, <ATOM(born) [NULL, NOUN, ADVERB, VERB]>, <ATOM(to) [NULL, TO, ADVERB, VERB]>, <ATOM(fly) [NULL, ADVERB, VERB]><PERIOD >]

<!-------POS Tag Generation-------->

<!-------POS Tag Evaluation------->

After post processing

[<ATOM(i) [PRONOUN]>, <ATOM(was) [HVERB]>, <ATOM(born) [NOUN]>, <ATOM(to) [TO]>, <ATOM(fly) [VERB]><PERIOD>]

END

<!-------POS Tag Evaluation------->

4.2 Example 2

String entered is

My Smart phone is black.

<!--------Lexer----------->

Lexene generated

[<ATOM(My) [NULL]>, <ATOM(Smart) [NULL]>, <ATOM(phone) [NULL]>, <ATOM(is) [NULL]>, <ATOM(black) [NULL]><PERIOD>]

<!--------Lexer----------->

<!-------POS Tag Generation-------->

Complete dictionary tagging

[<ATOM(My) [NULL, PRONOUN]><ATOM(Smart) [NULL]><ATOM(phone) [NULL]><ATOM(is) [NULL, HVERB]><ATOM(black) [NULL, ADJECTIVE]><PERIOD>]

Filling Noun Phrase

[<ATOM(My) [NULL, PRONOUN]><ATOM(Smart) [NULL, ADJECTIVE]><ATOM(phone) [NULL, NOUN]><ATOM(is) [NULL, HVERB]><ATOM(black) [NULL, ADJECTIVE, NOUN]><PERIOD>]

Filling Verb Phrase

[<ATOM(My) [NULL, PRONOUN]><ATOM(Smart) [NULL, ADJECTIVE]><ATOM(phone) [NULL, NOUN]><ATOM(is) [NULL, HVERB, VERB]><ATOM(black) [NULL, ADJECTIVE, NOUN, ADVERB, VERB]><PERIOD>]

Intermediate filling

[<ATOM(My) [NULL, PRONOUN, NOUN]>, <ATOM(Smart) [NULL, ADJECTIVE]>, <ATOM(phone) [NULL, NOUN]>, <ATOM(is) [NULL, HVERB, VERB]>, <ATOM(black) [NULL, ADJECTIVE, NOUN, ADVERB, VERB]><PERIOD>]

<!-------POS Tag Generation-------->

<!-------POS Tag Evaluation------->

AFTER POST PROCESSING

[<ATOM(My) [PRONOUN]>, <ATOM(Smart) [ADJECTIVE]>, <ATOM(phone) [NOUN]>, <ATOM(is) [HVERB]>, <ATOM(black) [ADJECTIVE]><PERIOD>]

END

5 CONCLUSION:

We have successfully designed a tool that can provide POS tag to sentences efficiently. This System is able to tackle various problems that are faced by POS tagger. System can differentiate between ambiguous tags. System uses a combination of Rule based and Dictionary based approach, it combines the strength of both approaches but do not include the weakness of any. System also improves its knowledge from the tag it generates and thus become more stable and accurate.

System is limited to improve its vocabulary only. This System can be made more promising if it also learns rules form the tagged sentences.

6 REFERENCES

- "*How English Works a Grammar Practice Book with Answers*" by Michael Swan and Catherine Walter, Published by Oxford University Press, Sixth Edition.

- Eugene Charniak, Curtis Hendrickson Neil Jacobson, and Mike Perkowitz. 1993, equations for Part of Speech tagger-generator. In *Proceedings of the Workshop on Very Large Corpora*, Copenhagen Denmark.

- Hans van Halteren, Jakub Zarvel, and Walter Daelemans 1998. Improving data driven world class tagging by system combination. In *Proceedings of the International Conference on Computational linguistics COLING-98, pages 491-497, Montreal Canada..*

- *Martin Volk and Gerold Schneider. 1998 Comparing a statistical and a rule-based tagger fro German. In Proceedings of KONVENS-98, page 135-137. Bonn.*

- *Church, K. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In Proceedings of the Second Conference on Applied Natural Language Processing, ACL , 136-143, 1988.*

- *Hindle, D. Acquiring disambiguation rules from text. Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics , 1989*

- *Klein, S. and Simmons, R.F. A Computational Approach to Grammatical Coding of English Words. JACM 10: 334-47. 1963.*

- *Meteer, M., Schwartz, R., and Weischedel, R. Empirical Studies in Part of Speech Labeling, Proceedings of the DARPA Speech and Natural Language Workshop , Morgan Kaufmann, 1991.*

- *Shrivastava, M, B. B. Mahaptra, N. Agarwal, S. Sing, P. Bhattacharya. 2005. Morphology-based Natural Language Processing Tools for Indian Languages. Morphology Workshop, CFILT, IIT Bombay, INDIA.*
- *Aronoff, Mark. 2005. What is Morphology?. Blackwell. UK.*