Spring 2-19-2020

# Parts of Speech Tagging: Rule-Based

Bao Pham
*Harrisburg University of Science and Technology*, bhpham@my.harrisburgu.edu

# *Parts of Speech Tagging: Rule-Based*

Bao Pham
Student
Computer Information Sciences
Harrisburg University of Science and Technology
Harrisburg, Pennsylvania, USA
BHPham@my.harrisburgu.edu

***Abstract*—Parts of speech (POS) tagging is the process of assigning a word in a text as corresponding to a part of speech based on its definition and its relationship with adjacent and related words in a phrase, sentence, or paragraph. POS tagging falls into two distinctive groups: rule-based and stochastic. In this paper, a rule-based POS tagger is developed for the English language using Lex and Yacc. The tagger utilizes a small set of simple rules along with a small dictionary to generate sequences of tokens.**

***Keywords—(POS, tagger, rule, definition, context, syntax)***

## I. Introduction

Part of speech (POS) tagging is the process of marking up a word in a text corresponding to a part of speech [1]. The assignment of the word can be based on its definition or the context to its relationship with adjacent and related words in a phrase, sentence, or paragraph [1]. POS tagging falls primarily into two distinctive groups: rule-based and stochastic [1]. Many natural language processing (NLP) applications utilize stochastic techniques to determine part of speech. The appeal of stochastic techniques over traditional rule-based techniques comes from the ease of the necessary statistics automated acquisition. In addition, rule-based applications are often difficult to implement and not as robust.

Stochastic taggers have obtained a high degree of accuracy without relying on pure syntactic analysis of the input. These POS taggers rely on the Hidden Markov Model (HMM) which captures the lexical and contextual information [1]. The parameters within this model can be estimated from the tagged or untagged text. Once the parameters are estimated, the input is automatically assigned with the highest probability of tag sequence based on the model. The performance of model is often enhanced through higher level of preprocessing techniques or by manually tweaking the model [1].

Although stochastic taggers contain a higher degree of accuracy, there is great redundancy with their permutation generation for the sequence of tags. Rule-based taggers reduce such redundancy with a small set of meaningful rules as opposed to large tables of statistics needed by the stochastic model. These rules are based on the formal syntax of the language. In combination, a probabilistic model can be applied upon the rule-based model [1]. This allows the rule-based model to be more robust and reduced the redundancy that a pure stochastic model has [1].

## II. English Language

Languages were taken to be sociological entities as clusters of properties shared by a group of speakers and lumped together as natural languages [2][3]. These properties were lists of sounds, words, and morphemes [2]. Any other properties were considered as universal logic or related to individual habits. Many linguistic works have dealt with the distribution of words and morphemes, syntax. However, nineteenth century syntax had no inherent structure or system unlike the twenty-first-century syntax [3]. Thus, it is difficult to have a history of syntax unlike words, pronunciations, distributions, and semantics [3].

As sociological entities, natural languages are constantly changing either by figure of speech, context of words, or the human nature. Words can have one fixed meaning in the dictionary. However, when used in a sentence, its meaning might change based on its context or relation to other words. As a result, the determination of a word’s context within a text is inherently difficult.

Formal language is a part of natural language [1]. It only exists in well-formed sentences as specific rules can easily determine the structure of the formal sentence [1]. However, it is unable to determine the semantics due to its reliance on fixed rules or when the structure is no longer formal [1].

The English language is considered as a Germanic language [2][3]. Many factors influenced this language and converted it into the prevalently analytical language of modern time [2]. In combination with scarcity of nominal forms and a verbal system, English outweighs the systems of many other European languages in terms of its segmentation of its verbal component [3]. It has a rich vowel system along with an enormous set of vocabulary that is incomparable to other Germanic and non-Germanic languages [3].

The modern English language reflects many centuries of development. The political and social events occurred in the past have profoundly affected how the language is structured. Similarly, other languages are subjected to the constant growth and decay. When a language ceases to change, it becomes a dead language [2]. One good example is Latin. Currently, English has become native to many large populated countries. It has become a unique tool or even a bridge for mutual understanding between people of all parts of the world [2]. Hence, as the English language becomes more prominent, its complexity will also increase.

## III. English structure

Human language consists of signs, which are defined as things that represent something else. There are three types of signs: iconic, indexical, and symbolic [2]. Iconic signs resemble things they represent, i.e., photographs. Indexical signs point to a necessary connection with things they represent. i.e., a symptom to an illness. Symbolic signs are conventional representation of things. A good example is the indication of a wedding ring to marriage [2]. These signs are often related to the context of a word in a sentence [2].

Certain aspects of word orders are considered as iconic [2][3]. For example, the sentence, *He cooks the food and became ill*, has a different meaning when rearranged as *He became ill and cooks the food*. In addition, rules of the language or syntactic rules limit how words in the sentence are ordered [1][2]. Thus, a sentence, *like soap operas I*, is inaccurate.

Each word in a sentence is identified with a part of speech. A part of speech is consisted of verb (V), adjective (ADJ), pronoun (PN), noun (N), adverb (ADV), conjunction (CONJ), preposition (PREP), determiner (DET), transition (T) and modal (M). Each part can be decomposed into more sophisticated parts or be grouped with others to create another part. For example, modal (auxiliary verb) is a subpart of verb, and a determiner or an article when grouped with a noun creates the subject part.

In combination to the arrangement of words, the tag of the word can change significantly. For example, the word *round* can be tagged differently.

N – a <u>round</u> of drinks

A – a <u>round</u> table

V – <u>round</u> off the numbers

PREP – come <u>round</u> the corner

ADV – come <u>round</u> with some fresh air

There are two fundamental rules in generating the structure of a sentence: phrase structure rules and transformation rules [2]. These two rules are constitutive rules rather than regulatory rules (constraints) [2][3]. Phrase structure rules generate the deep underlying structure (D-struct) of the sentence through determining the linear order of words in a simple, positive and declarative sentence, the lexical and the tag to which the words belong and their hierarchical relationships with each other [2]. The transformational rules either rearrange, add, or delete elements, but the semantic of the sentence remains [2]. Transformation rules generate various sentence types of surface structure (S-struct) [2].

*The dog uncovered the bone.* (D-struct)

*The bone was uncovered by the dog.* (S-struct)

Prior to the phrase structure rules, many linguists used immediate constituent analysis (IAC) which accounted for the linear order of words on the surface and the structure of the sentence [1][2]. However, IAC was proven to be insufficient in dealing with an active and corresponding passive sentence as argued by Noam Chomsky [2]. For example, the sentence, *flying planes can be dangerous*, has great ambiguity. *Flying planes* can either means an action or a noun.

A phrase structure grammar consists of a set of ordered rules known as rewrite rules (Chomsky Normal Form) [2]. A simple sentence can be structured as:

S => NP + VP

NP => DET + N | N

VP => V | ADV V

O => N

The example S => NP + VP + O constitutes the grammar of a sentence that is composed of NP, VP and O. NP is composed of either DET and N or just N. Hence, S => DET + N + VP + O | N + VP + O.

Sentences are composed of phrases, which are either sequence of words or a single word having syntactic significance where they form a constituent. A constituent is a word or a group of words that functions as a single unit in the hierarchical structure [2][3]. *A beautiful flower* is a constituent where *A* (DET) *beautiful* (ADJ) *flower* (N) act as one subject. However, not all sequences of words function as constituents. It is the context that determines whether a sequence forms a constituent [1][2][3].

To transform simple sentences into complex or compound sentences, phrases can be expanded.

| | |
|---|---|
| S => NP VP | |
| NP (noun phrase) => | N |
| | DET N |
| | DET ADJ N |
| | DET ADJ PREP N |
| | DET N PREP N |
| | DET ADJ N PREP |
| | PN |
| | Proper Noun (PrN) |
| AP (adjective phrase) => | ADJ |
| | ADV ADJ |
| | ADV ADV ADJ |
| AdvP (adverb phrase) => | ADV |
| | ADV ADV |
| PP (prepositional phrase) => | PREP NP |
| | PREP PREP NP |
| | PREP PP |
| VP (verb phrase) => | V NP |
| | V NP PP |
| | V NP NP |
| | V AP |
| | V NP AP |
| | V PP |
| | V PP PP |
| CONJ => | PP + NP |
| | P + P |
| | NP + NP |
| | AP + AP |
| | A + A |
| | AdvP + AdvP |

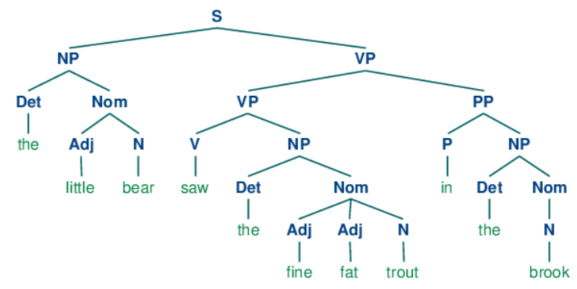


Figure 1. Phrase Structure Tree

In addition, a sentence can be active or passive. It is induced through the usage of auxiliary verbs or modals (M) [2]. The structure of a sentence then becomes:

S => NP Aux VP

Passive sentences are derived from their active counterpart by the insertion of the passive auxiliary in the verb specifier position which causes the NP to move to the *by* phrase [2].

*The art expert detects the forgery.* (active)

*The forgery is detected by the art expert.* (passive)

The most frequent kind of passive sentence in English is the agentless passive where the *by* phrase is not present [2][3]. In addition, not all active sentences can be passivized [2]. For a sentence to be passivized, the subject must be a performer of an action or an agent and the verb must have a direct or prepositional object that allows reorder of the subject position [2][3]. Certain verbs like intransitive, and copulative verbs cannot be passivized as they cannot have an object. Although many transitive verbs can be passivized, some may not be as the subject is not performing an action [2].

*Jack eats the chocolates.* (can be passive)

*Jack hates the chocolates.* (can't be passive)

Many D-struct sentences are active as opposed to passive, declarative, positive, and simple [2]. These sentences serve as a base or a kernel sentence to produce passive, imperative, or negative sentences (S-struct) through the usage of transformation rules [2]. Understanding the structure of a sentence allows the proper tagging or categorizing of its words.

## IV. Related Works

There have been efforts in implementing an effective rule-based POS tagger.

Brill [1] implemented a simple rule-based tagger that utilizes a probability model. The tagger performed as well as existing stochastic taggers with the advantage of performance and portability through elimination of many large tables of statistics.

Amir [4] et al. implemented a stochastic POS tagger for the language. Amazigh Corpus, utilizing HMM. They concluded that a hybrid solution should be implemented as its accuracy can be as great or greater than the stochastic solution with significant gain in performance.

Chana [5] et al. improved the Sanskrit-Hindi translation system which is a hybrid system of rule-based and stochastic. They improved the neural machine translation of the system through the usage of rule-based machine translation. The hybrid system was able to reach an accuracy as high as 99%.

Cutting [6] et al. developed a stochastic tagger using HMM with a small interchangeable lexicon. The tagger can decipher the POS of different languages based on the lexicon and the training set.

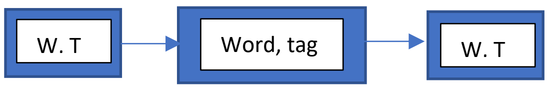
Anbananthen [7] et al. presented a comparison of stochastic and rule-based POS tagging on Malay text. They concluded that rule-based model is a better solution for the Malay language due to its great morphism. Rule-based approach utilizes linguistic rules that allow it to overcome ambiguity instead of the reliance on probability equatio***ns.***

## V. IMPLEMENTATION

The rule-based POS tagger is written in C and implemented on top of Lex and Yacc. There are two main files, *pos.l* and *pos.y*. Lex uses the *pos.l* file for lexicon and scanner rules. It is essentially the tagger file. Meanwhile, Yacc uses the *pos.y* file to create tags and to build structure rules. Yacc is a parser developed for Unix systems. Lex is a lexical analyzer built in conjunction with it.

The tagger is constructed utilizing a very small lexicon. The lexicon primarily contains words that are often used and usually have a fixed tag. For example, words belong in the tag, determiner, are a small set of words.

The tagger is rather reliance on linguistic rules to determine the tag of unknown words and ambiguity. When a word is parsed, it is checked with the lexicon first. If matched, it is passed onto the ***addword*** method, a set of linguistic rules are applied to determine the tag for the word, then the word is allocated within a doubly linked-list structure. The structure contains the word, its tag, its left neighbor, and its next neighbor.



If the word failed to match in the lexicon, it is passed onto the ***lookup*** method and the first round of rules is applied. The first-round checks to see if the word fits in a part of speech by checking its suffix and prefix. For example, if the word is end in *ous* or *est*, it is tagged as an adjective.

In addition, if the word cannot be determined through its suffix or prefix, then linguistic rules are applied to determine its tagged. These rules check the location of the current word in comparison to the previous word. For example, *he is fighting John*, where the current word is *John*, it is checked with *fighting* and see that it is a verb. The tagger will tag it as a noun as it sees that *John* cannot be a conjunction nor a determiner since the word is unknown. The rules within the lookup method follows:

1. *If the word doesn't pass the suffix/prefix check, then it is checked using linguistic rules.*
2. *If the previous word is not tagged as a DET, PN, DET-PN, or POSS-N, then it is tagged as* ***NOUN.***
3. *If the previous word is an ADV, then it is tagged as* ***V****.*
4. *If the previous word is an AUX_BE, then it is tagged as* ***ADJ****.*
5. *If the previous word is a PREP_BASIC, then it is tagged as* ***V.***
6. *If the previous word is the word* ***be****, then it is tagged as* ***ADJ.***
7. *If it is the first word and undetermined, then it is tagged as a noun.*

If the word is matched with any of the rules, then the ***addword*** method is called. Before the word is allocated onto the structure, it goes through another round of rules like the first round with slight variations. This decreases the ambiguity within the sequence of tags.

The ***addword*** method serves to overwrite the state of a word either unknown or known to the lexicon. For example, the word bear can be tagged as a verb initially after matching in the lexicon. After applying the rules, it is found that the previous word was an article or a modal, then it is now tagged as a noun.

Once the word is tagged and added to structure, the lexical analyzer returns its tag. The Yacc parser uses the returned tag and the Backus Normal Form (BNF) to check for the sentence of the structure. The rules mentioned in section English Structure are applied.

In addition, certain tags are broken down into more sophisticated tags to accommodate linguistic rules that deal with ambiguity. For example, noun was broken down further into *det-noun, pronoun, noun-day* and many more. As a result, the rules are more sophisticated when it comes to checking the tag and reduces further ambiguity.

## VI. Evaluation

The rule-based POS tagger utilized many linguistic rules to determine the tag of a word. The set of rules are limited to a small set to prevent the occurrence of contradictions. As the set of linguistic rules grow, the chance of contradiction also increases. Through limitation of a small set of rules, the tagger is unable to properly tag a word in a complex sentence.

Meanwhile, simple sentences and some complex sentences are properly tagged. Simple sentences are properly tagged as they follow formal language rules. Some sentences are properly tagged since no contradictions appeared when the linguistic rules are applied or when matched in the lexicon. For example, *He is happily eating his sandwich today,* and *He is running with his aunt,* are properly tagged.

However, due to contradictions within certain rules, the tagger can incorrectly tag the word. For example, the sentence, *she is going to work today*, is tagged as *pronoun-modal-verb-prep-noun*. The word today is an adverb in this case. This example is a common issue for both rule-based and stochastic approach. It is difficult to find good rules, or a set of training data to set what is after a preposition. In many cases, a noun is often after a preposition which requires a rule for such case. To implement a rule for a verb after a preposition is contradictory to that rule.

Handling unknown words is a main issue for both rule-based and stochastic tagging. Both have different approaches in handling such issue. In the stochastic approach, unknown words are tagged based on transition probability whereas affix analysis is used to tackle ambiguity [7]. The rule-based tagger can determine the tag of unknown words through the usage of rules, which it does not require a large lexicon However, it is unable to pickup an unknown word if it is the first word in a sentence. This is due to the nature of the Lex scanner where it will discard an unknown word immediately.

In addition, when it comes to analyzing the structure of the input, the Yacc parser has troubles in checking the structure utilizing the BNF structure. In some random occurrences, it is unable to recognize the tag of an unknown word. This was subsequently fixed through returning the tag in both ***lookup*** and ***addword*** methods and during the lexicon matching phase. Furthermore, the tagger does not have the ability of error-correction. Spelling error-detection is not necessary as the rule-based approach only look at the structure of the words within the sentence to tag. However, it can be handy when matching words in the lexicon.

## VII. Conclusion

Rule-based tagging is more efficient and faster than stochastic tagging. Through its usage of linguistic rules, it is quick at performing the tagging process. In addition, both rule-based and stochastic approach struggle with ambiguity and unknown words problems. Rule-based approach is better at tagging unknown words in when used in rich morphology languages [7]. Whereas stochastic approach requires greater time to assigning a tag to unknown word due to probability model and further analysis models [1][4][5]. However, with the stochastic approach, there is a higher percentage of accuracy in determining the proper tag of a word that has ambiguity or it is unknown. Although the rule-based approach is faster and efficient, it is not the right approach for POS tagging.

## VIII. Future Work

The rule-based approach is proven to be effective at improving the stochastic approach. A hybrid approach is the next step in improving the tagging process. However, prior to the hybrid approach, an analysis of stochastic approach will be conducted.

## References

[1] Brill, E. (1992). A simple rule-based part of speech tagger. *ANLC '92 Proceedings of the Third Conference on Applied Natural Language Processing* , 152–155. doi: 10.3115/974499.974526

[2] Brinton, L. J. (2000). *The Structure of Modern English: A linguistic introduction*. Philadelphia, PA: John Benjamins Publishing Company.

[3] Verba, L. G. (2004). *History of the English Language Edited by Dr. E. F. Riccio. New York University* (1st ed.). Vinnytsia, Ukraine: Nova Knyha.

[4] Amir, S., Zenkourar, L., & Benkhouya, R. (2019). A Comparative Study on the Efficiency of POS Tagging Techniques on Amazigh Corpus. *NISS19 Proceedings of the 2nd International Conference on Networking, Information Systems & Security*. doi: 10.1145/3320326.3320390

[5] Chana, I., Kumar, R., & Singh, M. (2019). Improving Neural Machine Translation Using Rule-Based Machine Translation. *2019 7th International Conference on Smart Computing & Communications (ICSCC)*. doi: 10.1109/ICSCC.2019.8843685

[6] Cutting, D., Kupiec, J., Pederson, J., & Sibun, P. (1992). A Practical Part-of-Speech Tagger. *Third Conference on Applied Natural Language Processing*, 133–140. doi: 10.3115/974499.974523

[7] Anbananthen, K. S. M., Krishnan, J. K., Sayeed, M. S., & Muniapan, P. (2017). Comparison of Stochastic and Rule-Based

POS Tagging on Malay Online Text. *American Journal of Applied Sciences*. doi: DOI: 10.3844/ajassp.2017.843.851