CSME 206A Natural Language & Speech Processing **Spring Semester**

Lecture 10: Parts of Speech Tagging

Lecturer: K.R. Chowdhary *: Professor of CS*

10.1 Introduction

We consider what tagging models are most appropriate as front ends for probabilistic context-free grammar parsers. In particular, we ask if using a "multiple tagger", a tagger that returns more than one tag, improves parsing performance. Our conclusion is somewhat surprising: single-tag Markov-model taggers are quite adequate for the task. First of all, parsing accuracy, as measured by the correct assignment of parts of speech to words, does not increase significantly when parsers select the tags themselves. In addition, the work required to parse a sentence goes up with increasing tag ambiguity, though not as much as one might expect. Thus, for the moment, single taggers are the best taggers.

Recent years have seen a spate of research on various techniques for "tagging"- assigning a part of speech (or "tag") to each word in a text. Consider the following example:

The	can	will	rust
article	modal-verb	**modal-verb**	noun
	noun	noun	**verb**
	verb	verb	

Under each word we give some of its possible parts of speech in order of frequency. The correct tag is given in bold font.

One justification for tagging research is that a tagger can serve as a front end to a parser: the tagger assigns the tags to the incoming words and thus the parser can work at the tag level, where parsers do best. This raises questions of how well different types of taggers work as front ends to parsers. Despite the abundance of work on taggers, these questions have yet to be addressed; it is still uncommon actually to read of a tagger used with a parser. and when one is so used there is no analysis of suitability.

This question becomes more important because of two strands within tagging research. While most taggers return a single best ta g for each word (we call these "single taggers"). some work has been done on taggers that return a list of possible tags in those cases where a second (or even third) best choice might be close to the best according to the tagger's metric (we call these "multiple taggers"). One obvious reason to do this would be to let the parser make the final decision.

10.2 Parts-of-Speech

Words can be analysed into parts-of-speech, which are major lexical syntactic categories, like, as N(Noun), V(Verb), A(Adjective), P(Preposition), or more minor categories, such as Comp(Complementizer), Det(Determiner),

Deg(Degree intensifier), and so on. Some examples are as follows:

```
N: car, cars; woman, women...
V: thinks, thinking; sold, selling...
A: old, older, oldest; pedantic...
P: in, on, with(out), although...
Comp:  that, if...
Det:  the, a, those, that, some...
Deg:  so, very...
```

The N, V, A are the categories of the contentful or open-class vocabulary. Member- ship of these categories is large (as a glance at any dictionary will tell you) and open-ended (people invent new words (neologisms) like, fax, biro) and often open-class words belong to more than one category (e.g. storm can be a noun or verb, and morphologically-related stormy is an adjective); that is, they are ambiguous in terms of lexical syntactic category. (Some words are ambiguous at the level of lexical semantics though not in terms of lexical syntactic category e.g. match, N: game vs. lighter). Adverbs also form a large open-ended class, but they are highly related to adjectives and often formed by adding the suffix +ly to adjectives (badly, stormily, etc) so we would not give them a separate category but treat them as A[+Adv].

The other categories are those of functional or closed-class words, which typically play a more 'grammatical' role with more abstract meaning. Membership of these categories is smaller and changes infrequently. For example, prepositions convey some meaning but often this meaning would be indicated by case endings or inflection on words in other languages and sometimes there are English paraphrases which dispense with the preposition: "Kim gave a cat to Sandy" / "Kim gave Sandy a cat." Degree intensifiers in adjectival or adverbial phrases very beautiful(ly) convey a meaning closely related to the comparative suffix more beautiful / taller. Determiners, such as the (in)definite articles (the, a), demonstrative pronouns (e.g. this, that) or quantifiers (e.g. some, all) help determine the reference of a noun (phrase) – quite frequently articles are absent or indicated morphologically in other languages (hence the common non-native speaker error of the form: "please, where is train station?").

The complete set of lexical syntactic categories (for English) depends on the syntactic theory, but the smallest sets contain around 20 categories (almost cor- responding to traditional Greek/Latin-derived parts-of-speech) and the largest thousands.

Often words are ambiguous between different lexical categories. What are the possibilities for broken, purchase, that and can? There are diagnostic rules for determining the category appropriate for a given word in context; e.g.s: if a word follows a determiner, it is a noun, e.g., "the song was a hit." If a word precedes a noun, is not a determiner and modifies the noun's meaning, it is an adjective. For example, "the smiling boy laughed." Can you think of an exception to the last rule? These rules and categorical distinctions can be justified by doing distributional analysis both at the level of words in sentences. The process is more long-winded, though. The following template schemata are enough to get you to the rules above, which are abstractions based on identifying the classes, like noun, determiner, and adjective

1. – boy(s) can run
2. – older boy(s) can run
3. The – boy(s) can run
4. The older – can run

There are other ways to make these distinctions too. For example, nouns often refer to fairly permanent

properties of individuals or objects, boy, car, etc., verbs often denote transitory events or actions, *smile, kiss*, etc. However, there are many exceptions: *storm, philosophy, weigh, believe*, etc. Linguists have striven to keep syntax and semantics separate and justify syntactic categories on distributional grounds, but there are many interactions between meaning and syntactic behaviour.

There are eight parts of speech (POS) in English: noun, verb, pronoun, preposition, adverb, conjunction, participle, and article. The POS are also called *word classes, morphological classes*, or *lexical tags*. They are important as they give significant amount of information about word and its neighbours. It is true for nouns and verbs.

Also, when we have identified, e.g., possessive pronouns *my, your, his, her, its* and personal pronouns *I, he, you, me*, we are able to identify the vicinity words.

The POS are also used for Information retrieval, as knowing POS can help us as which morphological affixes it can have. They can also help in selecting important words, like, nouns, from the text.

Some examples of POS are as follows:

- *Prepositions:* on, under, over, near, by, at, from, to, with
- *Pronouns:* she, who, I, others
- *Wh-pronouns:* what, who, whom, why, where
- *Conjunctions:* and, but, or, as, if, whom
- *Auxiliary verbs:* can, may, should, is, are
- *Participle:* up, down, on, off, in, out, at, by

10.3 Tag-sets and Parts-of-Speech Tagging

There are tag-sets used for parts of speech Tagging. The Table 10.1 shows some of the tag sets.

Parts-of-speech tagging or tagging in short is process of assigning a parts-of-speech or other lexical class marker to each works in a given text. The tagging is also called *tokenization* in terms of computer based processing for natural language text.

Input to a tagging algorithm are: a string of words, specified tag-set and output is single best tag for each word (see Fig. 10.1).

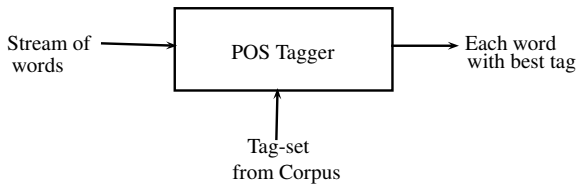

Figure 10.1: Process of Tagging.

Following are the examples of tagged sentences:

Table 10.1: Tag sets for English POS tagging.

Tag	Description
DT	Determiner (a,an, the, this, that, those)
IN	Preposition (of, by, in)
JJ	Adjective (yellow, other)
JJR	Adjective, comparative (bigger)
JJS	Adjective, superlative (biggest)
NNPS	Proper noun, plural (Indians)
NN	Noun (Rajan)
NNS	Plural noun (students)
NNP	Proper noun (IBM)
PP	Personal pronoun (I, you, he)
RB	Adverb (very)
TO	to go 'to'
VB	Verb (eat)
VBD	Verb past tense (ate)
VBZ	Verb 3rd pers. singular (eats)
VBN	Verb, Past participle (taken)
WDT	wh-detemining (which)
WP	wh-Pronoun (who, when)
WP$	Possessive wh-pronoun (whose)
WRB	wh-Adverb (where, when)

Sentence: Book that flight.
Tag Sequence: VB DT NN.

Sentence: Does that flight serve lunch ?
Tag sequence: VBZ DT NN VB NN ?

We note the ambiguous word "book" in the above example, which makes it difficult to resolve the meaning of the sentence. The POS also resolves the ambiguity using a Corpus (like Brown corpus, or Penn Treebank tag-set). The disambiguation is carried out based on frequency of use of those words as well based on the context in that sentence.

Using the tag-set and corpus (tagged collection of sentences) it is possible to tag the words in a sentence and resolve the POS. The following example is a longer sentence with tags:

The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS.

In the followings, we discuss some types of POS tagging.

Rule-based parts-of-speech tagging Various languages follow the structure as SV (subject verb), SVO (subject, verb, object), or SOV (subject, object, verb). They have identify the POS based on the sentence structure.

10.4 Stochastic POS tagging

In this method probabilities are used in POS tagging. This method uses the algorithm known as HMM (Hidden Markov Model). Based on this approach we pick up the most likely tag for the given word. For example, the HMM tagger choose the tag sequence that maximizes the following formula:

$$P(word \mid tag) * P(tag \mid previous\ n\ tags) \tag{10.1}$$

Example. HMM based POS tagger.

Consider the following example sentences:

1. Secretariat/NNP is/VBZ expected/VBN to/TO race/VB tomorrow/NN.

2. People/NNS continue/VBP to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN.

We shall resolve the POS of "race" in the first sentence. Thus we need to find out "to/TO race/??". We assume that tags for the word "race" are not given.

So we want to find the probability $P(VB \mid TO) * P(race \mid VB)$ vs probability of $P(NN \mid TO) * P(race \mid NN)$.

We can apply equation 10.1 to compute the probabilities as follows, using the probabilities given in the standard corpus as follows.

$P(NN \mid TO) = 0.021$

$P(VB \mid TO) = 0.34$

The lexical likelihoods from Brown corpus are :

$P(race \mid NN) = 0.00041$
$P(race \mid VB) = 0.00003$

If we multiply the lexical likelihoods with the tag sequence probabilities, we find the conclusions as follows:

$P(VB \mid TO) * P(race \mid VB) = 0.00001$
$P(NN \mid TO) * P(race \mid NN) = 0.000007$

Hence, in the first sentence the POS of word *race* is VB.

In the similar way we can resolve for "the/DT race/??" in the second sentence, and we find that in the case of "race" as NN, the probability is higher than VB.

□

10.5 Transformation-based POS tagging

It is an instance of Transformation based Learning (TBL) approach to machine learning. It draws the inspiration from rule based tagging as well as from Stochastic tagging.

10.6 Tools for NLP

There exists number of tools, either as open source or research tools created by some research laboratories, as well closed source for natural language processing, and speech processing. These tools have built-in functions to perform number of commonly used tasks, which can be directly called, or using these script can be written to perform more complex jobs of NLP and speech processing. For example, for NLP, they can tokenize the given text, can do stemming and POS (parts of speech) tagging, can find out word frequencies in given documents, parsing of NL sentences, etc. These inputs can help to compute, for example, $tf \times idf$ (term frequency * inter-document frequency), which can be helpful in IR (Information Retrieval), IE (Information Extraction), text classification, etc. In the following part, we discuss some such tools, which are either open source or they can be obtained on request from respective research laboratories.

10.6.1 NLTK

The NLTK (Natural Language Toolkit) is a collection of Python libraries and programs for *symbolic* and *statistical natural language processing*. It is suited to practitioners as well those who are learning natural language processing (NLP), those who are conducting research in NLP or areas close to NLP, like, empirical linguistics, cognitive science, AI, IR, and machine learning.

The NLTK has been also used as a teaching tool, as a study tool for individuals, and as a prototyping and building platform for research systems. Python language has been chosen due to its shallow learning curve, its transparent syntax and semantics, and due to its extraordinary capability for handling strings. The python is an interpreted language, which facilitates interactive exploration. It is an object-oriented language, allows data and methods to be encapsulated and re-used easily. Python is also available with an extensive standard library, that includes tools for speech processing, natural language processing, numerical processing, and graphical programming [nltk17].

Consider that tasks of stemming and parts-of-speech (POS) tagging are independent, and both operate on sequences of tokens. If the stemming task is done first, the information required for POS tagging is lost. If tagging task is performed first, the stemming process must be able to skip over the tags. If these two tasks are done independent to each other, it become difficult to align the resultant texts. Hence, as the combinations of tasks increase, it becomes extremely difficult to manage the data. To address this problem, NLTK version 1.4 onward comes with a new architecture where tokens are based on Python's native dictionary datatype, such that the tokens can have an arbitrary number of *named properties*. The *Tag* and *Stem* are the examples of these properties. The NLTK allows for even whole sentence and document to be represented as single token, with *Sub-tokens* attribute that hold sequences of smaller tokens.

A *parse-tree* can also be treated as a token, which have special property/attribute of *Children*. The benefit of this type of architecture in NLTK is that, it unifies many different data types, and allows distinct tasks to be run independently. Of course, this architecture comes with an overhead for programmers, because the program need to keep track of a growing number of property names.

10.6.2 NLTK examples

Example. Tokenization of natural language text.

The following commands in Python, with NLP tool NLTK installed, demonstrates the tokenization of a given text into sentence tokens, and word tokens. Since there is only one sentence, the sentence token is one only.

```
$ python
Python 2.7.14 (default, Sep 23 2017, 22:06:14)
[GCC 7.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more
                                                    information.
>>> from nltk.tokenize import sent_tokenize, word_tokenize
>>> text="Fundamentals of Artificial Intelligence."
>>> print(sent_tokenize(text))
    ['Fundamentals of Artificial Intelligence.']

>>> print(word_tokenize(text))
    ['Fundamentals', 'of', 'Artificial', 'Intelligence', '.']
>>>
```

□

Example. Stemming of given set of words.

Following are the commands for stemming a set of word to reduce them to their stem words. This makes use of Porter Stemmer algorithm.

```
>>> from nltk.stem import PorterStemmer
>>> ps=PorterStemmer()
>>> words=["python", "pythoning", "pythonize", "pythonly"]
>>> for w in words: print(ps.stem(w))
...
python
python
python
pythonli
>>>
```

The parts-of-speech tagging (grammatical tagging) or disambiguation of word-category, is a process of marking-up word in a text (corpus) corresponding to a particular POS. This is carried out based on its definition as well as its context[1]. Various POS in English language are: noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection.

The POS tagging is carried out as part of computational linguistics, using some algorithms. These algorithms associate discrete terms, as well as hidden parts of speech, in accordance with a set of descriptive tags. POS-tagging algorithms fall into two distinctive categories: *rule-based* and *stochastic* based. For example, Brill's

[1]Context: Relationship with adjacent and related words in a sentence, or phrase, or a paragraph.

tagger, one of the first and most widely used English POS-taggers, makes use of rule-based algorithms. The following example demonstrates the POS tagging using NLTK.

Example. Parts-of-speech tagging.

```
>>> import nltk
>>> text=nltk.word_tokenize("Part of speech tagging and POS
                                                    tagger")
>>> text
 ['Part', 'of', 'speech', 'tagging', 'and', 'POS', 'tagger']
>>> nltk.pos_tag(text)
 [('part', 'NN'), ('of', 'IN'), ('speech', 'NN'),
 ('tagging', 'NN'), ('and', 'CC'), ('POS', 'NNP'), ('tagger',
                                                      'NN')]
```

Review Questions

1. What are the wh-pronouns? What are their significance?

2. Which of the following is correct?

   ```
   Pronoun -> wh-pronoun
   ```

   ```
   wh-pronoun -> noun
   ```

   Justify it.

Exercises

1. Write an algorithm of POS tagging, which uses some hypothetical Corpus lexical database of already tagged text.

2. Write an algorithm for POS tagging, that uses Stochastic method discussed in this chapter.

3. Search on the net and find out the Corpus databases, available as open source, and then write a small note about each in brief, covering mainly the technical aspects.

4. Give a comparison of three approaches discussed in this chapter for POS tagging, in respect of efficiency, complexity, and accuracy.

5. What can be the POS immediately following / immediately preceding to the following parts-of-speech words, in any syntactically correct English language sentence:
   Noun, Pronoun, Preposition, Adjective, Adverb, Verb, Conjunction, Connective.

6. Discuss the application of POS-tagging for NLP? Give sufficient examples, with logical explanations.

7. Explain the difference between POS tagging and Tag-sets. Give a list of five words in each category. What is application of tag-sets?

References

[1] D. Jurafsky and J. Martin, *Speech and Language Processing, Pearson India*, 2002.

[2] Ted Briscoe, *Introduction to Linguistics for Natural Language Processing*, October 6, 2010, Computer laboratory, University of Cambridge, Monograph.

[3] `https://www.nltk.org/`, Cited 19 Dec 2017