ML for Human Vision and Language

MSc Artificial Intelligence

Lecturer: Tejaswini Deoskar
t.deoskar@uu.nl

UiL-OTS, Utrecht University

Block 1, 2019

Parts-of-speech

also called : *word classes, lexical classes, grammatical classes, lexical tags, ...*

- Linguists/philosophers have been classifying words for a long time
- *Dionysius Thrax of Alexandria* (c. 100 BC) wrote a grammatical sketch of Greek involving **8 traditional parts-of-speech**

noun	verb	pronoun	preposition
conjunction	adverb	particle	article/determiner

Criteria for classifying words

When should words be put into the same class?

- Semantic criteria : What does the word refer to? (Nouns often refer to 'people', 'places' or 'things')
- Distributional criteria : In what contexts can the word occur ?
  - *crocodile*, *pencil*, *mistake* have different meanings, but can occur in the same contexts (for e.g. after ‘the’).
- Formal criteria : What form does the word have? (e.g. -tion, -ize). What affixes can it take (e.g. -s, -ing, -est)
  - *walk*, *slice*, *donate*, *believe* don't have much in common semantically, but can all combine with suffix *-s* or *-ed*

Criteria for classifying words

	Semantically	Formally	Distributionally
Nouns	refer to things, concepts	-ness, -tion, -ity, -ance	After determiners, possessives
Verbs	refer to actions, states	-ate, -ize	infinitives: to jump, to learn
Adjectives	properties of nouns	-al, -ble	appear before nouns
Adverbs	properties of actions	-ly	next to verbs, beginning of sentence

Importance of formal and distributional criteria

Even if we don't know its meaning, formal and distributional criteria help people (and machines) recognize which (open) class a word belongs to.

Those zorls you splarded were malgy.

Formal and distributional criteria are also useful when we (or our system) comes across unknown words

POS tagging: Why do we care?

- First step towards syntactic analysis (which in turn, is often useful for semantic analysis).
- Simpler models and often faster than full syntactic parsing, but sometimes enough to be useful
  - ∗ POS tags can be useful features in e.g. text classification, authorship identification, etc.
  - ∗ Useful for applications such as text to speech synthesis: "it is time to wind the clock up" versus "the wind was strong"
- POS tagging task also helps introduce some useful techniques: Hidden Markov models(HMMs) or Recurrent Neural networks (RNNs), which are used for many other sequence labelling or sequence modelling tasks.

How many parts of speech?

- Both linguistic and practical considerations
- Should we distinguish between
  - proper nouns (names) and common nouns ?
  - past and present tense verbs?
  - auxiliary and main verbs?
- Coarse or fine-grained tag sets can be picked.
- Brown corpus (**87** tags)
- Penn Treebank corpus (**45** tags)

Universal POS tags

- Recently promoted by Google and others.
- Simplify the set of tags to lowest common denominator across languages
- Map existing annotations onto universal tags
  VBD, VBN, VB, VBG, VBP $\rightarrow$ VERB
- Allows interoperability of systems across languages

NOUN (nouns), VERB (verbs) , ADJ (adjectives), ADV (adverbs), PRON (pronouns), DET (determiners and articles), ADP (prepositions and postpositions), NUM (numerals), CONJ (conjunctions), PRT (particles), ? (punctuation marks), X (anything else, such as abbreviations or foreign words)

The tagging problem (example of POS tagging *inference*)

Given an input text, we want to tag it correctly with POS tags for each word:

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

There/EX was/VBD still/JJ lemonade/NN in/IN the/DT bottle/NN ./.

- In the first example, number and bottle are nouns, not verbs.
- In the second example, still could be an adjective or adverb.

The POS tagging problem is : *to determine the POS tag for a particular instance (token) of a word in context.*

Why is POS tagging hard?

The usual reasons!

- Ambiguity: Words often have more than one POS

  back

  - The *back* door = JJ (**bijvoeglijk naamwoord**)
  - On my *back* = NN (**zelfstandig naamwoord**)
  - Win the voters *back* = RB (**bijwoord**)
  - Promised to *back* the bill = VB (**werkwoord**)

- Sparse data: Words we haven't seen before ; Word-tag pairs that we haven't seen before

Extent of POS Ambiguity

The Brown corpus (1M word tokens) has 39,440 different words (types).

- 89.6% word types (35340) have only **1** POS tag anywhere in corpus
- 10.4% word types (4100) have **2 to 7** POS tags

So why does just 10.4% POS-tag ambiguity by *word type* lead to difficulty?

Many high-frequency words have more than one POS tag.

In fact, more than 50% of the word *tokens* are ambiguous.

He wants to/TO go.	He wants that/DT hat.
He went to/IN the store	It is obvious that/CS he wants a hat.
	He wants a hat that/WPS fits.

Extent of ambiguity in Different languages

Ambiguity by part-of-speech tags:

Language	Type-ambiguity	Token-ambiguity
English	13.2%	56.2%
Greek	<1%	19.14%
Japanese	7.6%	50.2%
Czech	<1%	14.5%
Turkish	2.5%	35.2%

Some tagging strategies

- One simple strategy: just assign to each word its most common tag. (Call this Uni-gram tagging)
- Surprisingly, even this crude approach typically gives around 90% accuracy. (State-of-the-art (English) is about 98%).
- Can we do better?

Bi-gram tagging

- We can do much better by looking at pairs of adjacent tokens.
- For each word (e.g. still), tabulate the frequencies of each possible POS given the POS of the preceding word.

still	DT	MD	JJ
NN	8	0	6
JJ	23	0	12
VB	1	12	2
RB	6	45	5

- Given a new text, tag the words from left to right, assigning each word the most likely tag given the preceding one.
- Could also do trigram, 4-gram etc., but frequencies might be too sparse to be useful...

Problems with bi-gram tagging

- One incorrect tagging choice might have unintended effects:

	The	still	smoking	remains	of	the	campfire
Intended:	DT	RB	VBG	NNS	IN	DT	NN
Bigram:	DT	JJ	NN	VBZ	...		

- No lookahead: choosing the "most probable" tag at one stage might lead to highly improbable choice later.

	The	still	was	smashed
Intended:	DT	NN	VBD	VBN
Bigram:	DT	JJ	VBD?	

We want to find the **overall most likely** tagging sequence given the bigram frequencies.