T-(538|725)-MALV, Natural Language Processing
PoS tagging – with rules

Hrafn Loftsson[1] Hannes Högni Vilhjálmsson[1]

[1]School of Computer Science, Reykjavik University

October 2010

Outline

Outline

What is PoS tagging (í. mörkun)?

A definition

- To label (í. marka) each word in a text with the appropriate *word class* (í. orðflokkur) and *morphological features* (í. beygingarleg einkenni).
- The string used as a label is called *a tag* (í. mark)

Why is this difficult?

- Some words are ambiguous (í. margræð).
  - A tagger is sometimes called *a disambiguator*, since it performs ambiguity resolution (í. einræðing).
- When looking up a word in a dictionary or performing morphological analysis $\Rightarrow$ more than one tag (analysis) for the word is possible.

PoS tagging

Significance

- The tag for a word gives important information about the word and its neighbors.
  - "You shall know a word by the company it keeps" (Firth, 1957)
  - For example, the gender, number and case of an adjective signify comparable features for the following noun.
- Helps with speech synthesis.
  - OBject (noun) vs. obJECT (verb)
- The base for grammar checking, machine translation, parsing.
- Used in the construction of annotated corpora.

PoS tagging

Tagset

- *A tagset* (í. markamengi) is the set of all possible tags (labels)
- Different languages have different tagsets.
- The same language can have more than one tagset.
- **Icelandic**: *The Icelandic Frequency Dictionary (Íslensk orðtíðnibók)* – 700 tags.
- **English**: *Penn TreeBank*: 45 tags, *Brown Corpus*: 87 tags.
- **Swedish**: *Parole*: 139 tags.
- **Czech**: 1000-2000 tags.

PoS tagging

Full disambiguation (í. full einræðing)

- A single tag is assigned to each word (token).
- The most common method, but . . .
- . . . sometimes a tagger cannot perform full disambiguation.
- In that case, the tagger returns a set of possible tags for a given word.

Example: English PoS tagging

Penn Treebank tagset: `http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`

- The *back* door = JJ
- On my *back* = NN
- Win the voters *back* = RB
- Promised to *back* the bill = VB

Example: Icelandic PoS tagging

"Gamli maðurinn borðar kalda súpu með mjög góðri lyst" (Old man-the eats cold soup with very good appetite)

gamli	lkenvf
maðurinn	nkeng
borðar	sfg3en_sfg2en
kalda	lveosf_lkfosf_lkeovf_lkeþvf_lkeevf_lvenvf_ lhenvf_lheovf_lheþvf_lheevf
súpu	nveo_nveþ_nvee
með	aþ_aa
mjög	aa
góðri	lveþsf
lyst	nven_nveo_nveþ

Icelandic PoS tagging – disambiguation

"Gamli maðurinn borðar kalda súpu með mjög góðri lyst"

gamli	lkenvf
maðurinn	nkeng
borðar	sfg3en
kalda	lveosf
súpu	nveo
með	aþ
mjög	aa
góðri	lveþsf
lyst	nveþ

PoS tagging

Base tagging

- In a research for English and French: 50-60% of tokens have only one possible tag, 15-25% have only two possible tags.
- Assigning the most frequent tag for a word yields more than 75% accuracy.
- This is called *base tagging* (í. grunnmörkun)
- Charniak (1993) has obtained more than 90% accuracy by applying base tagging for English.
- Note that the underlying tagset plays an important role here.

Baseline tagging

In the Icelandic Frequency Dictionary (IFD):

- Unambiguous word forms: 84.16%
- Ambiguous word forms: 15.84%
- Ambiguous word forms with 2 tags: 11.07%
- Ambiguous word forms with 3 tags: 2.96%
- Ambiguous word forms with 4 tags: 0.97%

Which words are ambiguous?

- Usually the most common words, the function words.

Baseline tagging

Common words and their tags in the IFD

```
33181 .  .
22176 og    c
22083 ,     ,
21011 að    cn_c_aþ_aa
15319 í     aþ_ao_aa
12450 á     aþ_ao_sfg1en_sfg3en_aa_nven_nveo_nveþ_au
8040  hann  fpken_fpkeo
7905  var   sfg3eþ_sfg1eþ_lkensf
7676  sem   ct_c_aa_sfg1en
6357  er    sfg3en_sfg1en_ct_c
```

Outline

Measuring the accuracy

Full disambiguation

$$\text{accuracy (í. hittni)} = \frac{\#\text{ correctly tagged tokens}}{\text{total number of tokens}} \quad (1)$$

Not full disambiguation

$$\text{precision} = \frac{\#\text{ correct tags generated by the tagger}}{\text{total number of tags generated by the tagger}} \quad (2)$$

$$\text{recall} = \frac{\#\text{ correct tags generated by the tagger}}{\text{total number of correct tags}} \quad (3)$$

$$\text{ambiguity rate} = \frac{\#\text{ tags generated by tagger}}{\text{total number of tokens}} \quad (4)$$

Measuring the accuracy

Example: 100 tokens

- A tagger performs full disambiguation and correctly tags 95 tokens. $\Rightarrow$ *accuracy* $= 95/100 = 95\%$
- A tagger doesn't perform full disambiguation and returns 105 tags, of which 95 are correct.
  - $\Rightarrow$ *precision* $= 95/105 = 90.5\%$
  - $\Rightarrow$ *recall* $= 95/100 = 95.0\%$
  - $\Rightarrow$ *ambiguity rate* $= 105/100 = 1.05$

  Note that when full disambiguation is applied then *accuracy*=*precision*=*recall* and *ambiguity rate*=1.0.
- Icelandic terms: precision=nákvæmni, recall=griphlutfall, ambiguity rate=margræðnihlutfall

Accuracy in PoS tagging

What can affect the accuracy?

- The type of tagger – the quality of the language model.
- The size of the tagset.
- The ratio of unknown words.
  - The possible tags for unknown words are not known!
  - An *unknown word guesser* is needed.
- The size of the training corpus.
- The type of the test corpus.

Accuracy in PoS tagging

- English:
  - 96.7% (Brants, 2000)
  - Ratio of unknown words: 2.9%
  - Training corpus: 1,000,000 words.
  - Tagset: 45 tags (Penn TreeBank).
- Swedish:
  - 93.6% (Megyesi, 2002)
  - Ratio of unknown words: 15.0%
  - Training corpus: 100,000 words.
  - Tagset: 139 tags.
- Icelandic:
  - 92.5% (Loftsson et al., 2009)
  - Ratio of unknown words: 6.8%
  - Development corpus: 59,000 words.
  - Tagset: 700 tags.

Outline

Type of taggers

Rules vs. statistics

- Rules use the context of a word to eliminate or change a particular tag.
- Rules can be hand-written or learned in a data-driven manner from a tagged corpus.
- Statistical methods are used to assign words in a sentence the most likely tag sequence.
- Statistical methods use frequency information (e.g. n-grams) which are derived from a tagged corpus.

Type of taggers

Linguistic rule-based taggers (í. málfræðilegir reglumarkarar)

- Are based on hand-written linguistic rules.
- Only used to tag a particular language using a specific tagset.

Data-driven taggers (í. gagnamarkarar)

- Language and tagset independent.
- Use PoS tagged corpora to automatically collect information which is later used for disambiguation of new texts.
- This information can, for example, be in the form of statistics or rules.

Outline

Linguistic rule-based taggers

Typical functionality

1. Each word is assigned its *tag profile*, the set of possible tags for that word
   - Using a dictionary, or a morphological analyser, and/or an unknown word guesser.
2. Disambiguation using rules
   - Inappropriate tags eliminated with regard to context (reductionist approach)

Linguistic rule-based taggers

Typical functionality

Use rules about the nature of sentences and phrases to tags the words.

- A preposition does (usually) not appear before a verb
  - The word *fórum* is a noun in the context *í fórum mínum*.
- A possessive pronoun agrees with the following noun in gender, number and case.
  - In the context *hesta þinna* (horses yours), the word *þinna* is unambiguosly genitive case and therefore the word *hesta* is also genitive, but not accusative.

Linguistic rule-based taggers

Constraint Grammar Framework (Fred Karlsson 1990)

- A morphological analyser (based on two-level morphology) returns all possible analysis for each word.
- Rules (constraints) are written to eliminate tags with regard to context.
- Often thousands of rules, e.g. EngCG-2 with 3,600 rules.
- Time-consuming but recall is high. Does not perform full disambiguation for all words.
- Samuelsson and Voutilainen (1997):
  - Recall: 99.6%
  - Ambiguity rate: 1.02.
- Demo: `http://www2.lingsoft.fi/cgi-bin/engcg`

Linguistic rule-based taggers

IceTagger - Hrafn Loftsson

- Unknown word guesser: *IceMorphy*
- Local rules
  - About 175 rules.
  - Eliminate a specific tag in a particular context.
  - The local context is 5 words.
- Global rules
  - Heuristics (í. leitaraðferðir)
  - Guess the syntactic functions of words (subject, verb, object).
  - Mark preposition phrases.
  - Use the above to force feature agreement.

IceTagger

Full disambiguation

- The most frequent tag for a word is selected if a word is still ambiguous after the application of local and global rules.
- *IceTagger* is thus a combination of a linguistic rule-based tagger and a base tagger.

Test

- `http://nlp.cs.ru.is` and select *IceNLP*.

Outline

A tagger which learns rules

Brill's tagger (Eric Brill 1992)

- A data-driven tagger.
- Learns rules in training which change one tag to another.
  - $\Rightarrow$ "Transformation-based learning"
- A dictionary is derived from the training corpus.
  - Keeps track of the most frequent tag for a word.

Brill's tagger

Functionality

- Initially, assigns the most frequent tag to each word (base tagging)
- Applies a list of rules (transformations) to change the initial tagging.
- The rules are applied in a specific order and each transformation is applied on the text from left to right.
- An example for English:
  - "The can rusted"
  - With the most likely tag: The/**art** can/**modal** rusted/**verb**.
  - Rule: *Change the tag from modal to noun if the previous word is an article*.
  - Result: The/**art** can/**noun** rusted/**verb**.

Brill's tagger

How are the rules derived?

- Rules are based on templates.
- The templates restrict the type of rules that can be generated.
- Example template:

```
alter(A, B, prevtag(C))        Change A to B if preceding tag is C.
alter(A, B, nextbigram(C,D))   Change A to B if next bigram tag is C D.
```

- Brill used 11 templates for English, which resulted in about 500 rules, sufficient for achieving about 97% accuracy.

Brill's tagger – The training algorithm

St.	Operation	Input	Output
1.	Base tagging	Corpus	Corpus(1)
2.	Compare PoS of each word in *Gold standard* and Corpus(i)	*Gold standard* Corpus(i)	List of errors
3.	For each error, instantiate the rule templates to correct the error	List of errors	List of tentative rules
4.	For each rule, compute on Corpus(i) # of good transf. - # of bad transf.	Corpus(i) Tentative rules	Scored tentative rules
5.	Select the rule that has the greatest error reduction and append it to the ordered list of tranformations	Tentative rules	Rule(i)
6.	Apply Rule(i) to Corpus(i)	Corpus(i) Rule(i)	Corpus(i+1)
7.	If number of errors $< \delta$ exit else go to step 2		

Brill's tagger – Example of rules for **known** words

Training on the Icelandic Frequency Dictionary

- GOOD:1505 BAD:9 SCORE:1496 RULE: pos_0=sfg3eþ pos:[-2,-1]=fp1en ⇒ pos=sfg1eþ
  - Change tag **sfg3eþ** to **sfg1eþ** if one of the two previous tags is **fp1en**.
  - This rule corrected 1505 errors in the initial tagging.
- GOOD:836 BAD:38 SCORE:798 RULE: pos_0=aþ pos:[1,2]=nkeog ⇒ pos=ao
- GOOD:563 BAD:20 SCORE:543 RULE: pos_0=aþ pos:[1,2]=nveog ⇒ pos=ao

Brill's tagger

Unknown words

- Labels unknown words as proper nouns if they start with a capital letter.
- Labels all other unknown words as common nouns.
- Then applies special templates (see page 155 in the textbook) for generating rules which change the tag of an unknown word from X to Y.

Brill's tagger – Example of rules for **unknown** words

Training on the Icelandic Frequency Dictionary

- GOOD:558 BAD:3 SCORE:555 RULE: pos=nken word::∼∼1=∼∼a ⇒ pos=sng
  - Change tag **nken** to **sng** if the last letter of the word is "a".
  - This rule corrected 558 errors in the initial tagging but only added 3 errors.
- GOOD:438 BAD:6 SCORE:432 RULE: pos=nken word::∼∼3=∼∼nni ⇒ pos=nveþg
- GOOD:275 BAD:4 SCORE:271 RULE: pos=nheþ word::∼∼3=∼∼að ⇒ pos=sfg3eþ