Error-driven transformation-based PoS-tagging

Brill tagger

Miguel A. Alonso

Departamento de Computación, Facultad de Informática, Universidade da Coruña

Internal architecture of Brill tagger

Brill tagger is based on **error-driven, transformation-based** learning

It consisst of:

- Lexical tagger
- Unknown-words tagger
- Contextual tagger

Outline

Lexical tagger

- Each word is initially tagged with its most likely tag
- The context in which it appears is not taken into account
- Unknown words:
  - proper noun if the first letter is uppercase
  - otherwise, common noun

Outline

Unknown words tagger

Try to *guess* a tag for a unknown word as a function of its suffix, its prefix, and other relevant similar properties

Each transformation consists of two parts:

- a description of the application context
- a rewrite rule that replaces one tag with another

Generic template of lexical transformations

- `A` and `B` are variables over the set of all the tags
- `x` is any string of length 1, 2, 3 or 4
- `l` is the length of such a string
- `w` is any word
- `z` is any character

General template of lexical transformations

- `x haspref l A`: if the first `l` characters are `x`, the tag is set to `A`
- `A x fhaspref l B`: if the current tag is `A` and its first `l` characters are `x`, the tag is changed to `B`
- `x deletepref l A`: if erasing the prefix `x` of length `l` we get a known word, the tag is set to `A`
- `A x fdeletepref l B`: if the current tag is `A` and erasing the prefix `x` of length `l` we get a known word, the tag is changed to `B`
- `x addpref l A`: if adding the prefix `x` of length `l` we get a known word, the tag is set to `A`
- `A x faddpref l B`: if the current tag is `A` adding the prefix `x` of length `l` we get a known word, the tag is changed to `B`
- `x hassuf l A`: if the last `l` characters are `x`, the tag is set to `A`
- `A x fhassuf l B`: if the current tag is `A` and its last `l` characters are `x`, the tag is changed to `B`
- `x deletesuf l A`: if erasing the suffix `x` of length `l` we get a known word, the tag is set to `A`
- `A x fdeletesuf l B`: if the current tag is `A` and erasing the suffix `x` of length `l` we get a known word, the tag is changed to `B`
- `x addsuf l A`: if adding the suffix `x` of length `l` we get a known word, the tag is set to `A`
- `A x faddsuf l B`: if the current tag is `A` adding the suffix `x` of length `l` we get a known word, the tag is changed to `B`

General template of lexical transformations

- `w goodright A`: if it appears inmediatly to the right of word `w`, the tag is set to `A`
- `A w fgoodright B`: if the current tag is `A` and it appears inmediatly to the right of word `w`, the tag is changed to `B`
- `w goodleft A`: if it appears inmediatly to the left of word `w`, the tag is set to `A`
- `A w fgoodleft B`: if the current tag is `A` and it appears inmediatly to the left of word `w`, the tag is changed to `B`
- `z char A`: if it contanins the character `z`, the tag is set to `A`
- `A z fchar B`: if the current tag is `A` and it contains the character `z`, the tag is changed to `B`

Examples in Spanish

- `rse hassuf 3 V000f0PE1`: if the last 3 characters are `rse`, the tag is set to `V000f0PE1` (infinitive verb with enclitic pronoun)
- `r hassuf 1 V000f0`: if the last character is `r`, the tag is set to `V000f0` (infinitive verb)
- `V000f0 or fhassuf 2 Scms`: if the current tag is `V000f0` (infinitive verb) and the last 2 characters are `or`, the tag is changed to `Scms` (common noun, masculine, singular)
- `ría deletesuf 3 Vysci0`: if erasing the suffix `ría` of length 3 we get a known word, the tag is set to `Vysci0` (verb, first and third person, singular, postoperative of indicative)
- texttttScfs r faddsuf 1 V3spi0: if the current tag is `Scfs` (common noun, feminine, singular) and adding the suffix `r` of length 1 we get a known word, the tag is changed to `V3spi0` (verb, third person, present of indicative)

Examples in Spanish

- `el goodright Scms`: it it appears inmediatly to the word `el`, the tag is set to `Scms` (common noun, masculine, singular)
- `Scmp las fgoodright Scfp`: if the current tag is `Scmp` (common noun, masculine, plural) and the word appears inmediatly to the right of the word `las`, the tag is changed to `Scfp` (common noun, feminine, plural)
- `% goodleft Ncyyp`: if it appears inmediatly to the left of `%`, the tag is set to `Ncyyp` (numeral)
- `w char Ze00`: if the character `w` appears in the word, the tag is set to `Ze00` (foreign word)

Issue: there is a limit of 4 character in the length of prefixed and suffixes of the rules

Outline

Contextual tagger

- The contextual tagger proceeds just after the tagger of unknown words
- It applies, in order, a sequence of contextual rules that have been learned automatically from the training corpus
  - `A`, `B`, `C` and `D` are variables on the set of all tags
  - `w` and `x` are words

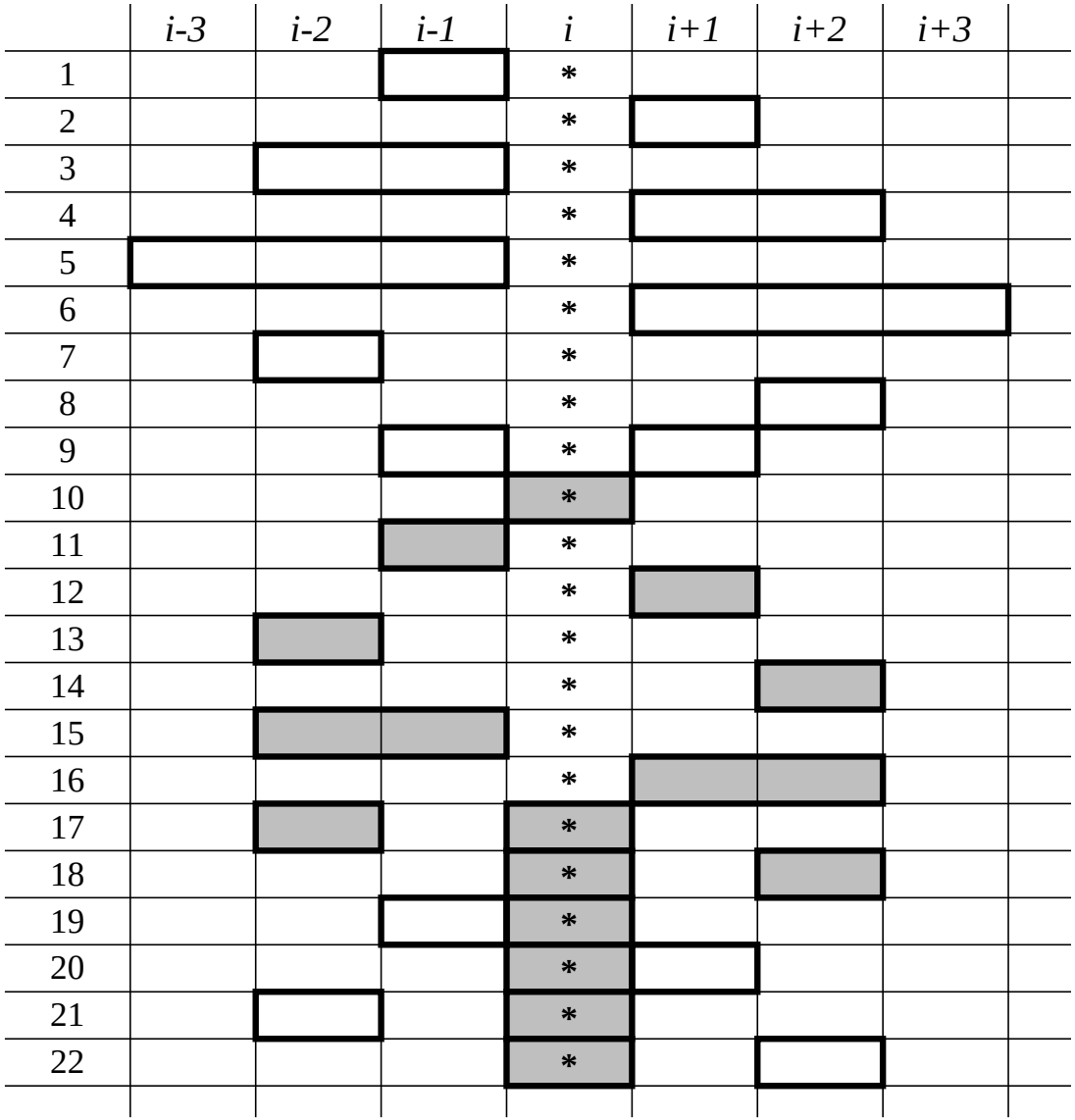
General template of contextual transformations

- `A B prevtag C`: change the tag `A` to `B` if the previous word is tagged as `C`
- `A B prev1or2tag C`: change the tag `A` to `B` if one of the two previous words is tagged as `C`
- `A B prev1or2or3tag C`: change the tag `A` to `B` if one of the three previous words is tagged as `C`
- `A B prev2tag C`: change the tag `A` to `B` if the previous second word is tagged as `C`
- `A B nexttag C`: change the tag `A` to `B` if the next word is tagged as `C`
- `A B next1or2tag C`: change the tag `A` to `B` if one of the two following words is tagged as `C`
- `A B next1or2or3tag C`: change the tag `A` to `B` if one of the three following words is tagged as `C`
- `A B next2tag C`: change the tag `A` to `B` if the following second word is tagged as `C`
- `A B prevbigram C D`: change the tag `A` to `B` if the previous word is tagged as `C` and the previous second word as `D`
- `A B nextbigram C D`: change the tag `A` to `B` if the next word is tagged as `C` and the following second word as `D`
- `A B surroundtag C D`: change the tag `A` to `B` if the previous word ias tagged as `C` and the following one as `D`

General template of contextual transformations

- `A B curwd w`: change the tag `A` to `B` if the current word is `w`
- `A B prevwd w`: change the tag `A` to `B` if the previous word is `w`
- `A B prev1or2wd w`: change the tag `A` to `B` if one of the two previous words is `w`
- `A B prev2wd w`: change the tag `A` to `B` if the previous second word is `w`
- `A B nextwd w`: change the tag `A` to `B` if the next word is`w`
- `A B next1or2wd w`: change the tag `A` to `B` if one of the two following words is `w`
- `A B next2wd w`: change the tag `A` to `B` if the following second word is `w`
- `A B lbigram w x`: change the tag `A` to `B` if the two previous words are `w` and `x`
- `A B rbigram w x`: change the tag `A` to `B` if the two following words are `w` and `x`
- `A B wdand2bfr x w`: change the tag `A` to `B` if the current word is `w` and the previous second word is `x`
- `A B wdand2aft w x`: change the tag `A` to `B` if the current word is `w` and the following second word is `x`
- `A B wdprevtag C w`: change the tag `A` to `B` if the current word is `w` and the previous word is tagged as `C`
- `A B wdnexttag w C`: change the tag `A` to `B` if the current word is `w` and the next word is tagged as `C`
- `A B wdand2tagbfr C w`: change the tag `A` to `B` if the current word is `w` and the previous second word is tagged as `C`
- `A B wdand2tagaft w C`: change the tag `A` to `B` if the current word is `w` and the following second word is tagged as `C`

Rule contexts



$*$= word to be tagged; boxes = tags; shadow boxes = words

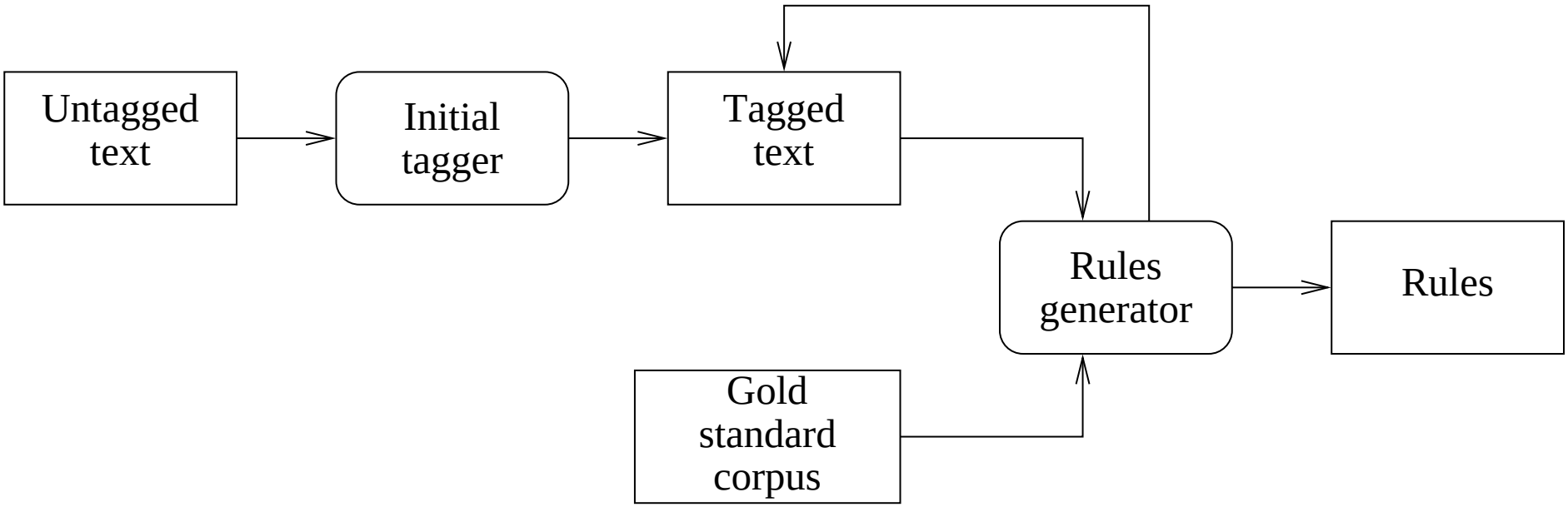
Examples in Spanish

- `Afp0 Amp0 prevtag Scmp`: change the tag `Afp0` (adjective, feminine, plural) to `Amp0` (adjective, masculine, plural) if the previous word is tagged as `Scmp` (common noun, masculine, plural)
- `Scms Ams0 wdprevtag Scms receptor`: change the tag `Scms` (common noun, masculine, singular) to `Ams0` (adjective, masculine, singular) if the curent word is `receptor` and the previous word is tagged as `Scms`
- `Scms Ams0 wdand2bfr el transmisor`: change the tag `Scms` (common noun, masculine, singular) to `Ams0` (adjective, masculine, singular) if the current word is `transmisor` and the previous second word is `el`
- `P Scms nexttag P`: change the tag `P` (preposition) to `Scms` (common noun, masculine, singular) if the next word is tagged as `P`

Outline

Error-driven, transformation based, learning

1. A portion of untagged text is taken
2. It goes through the tagging phase or phases
3. The output is compared to the correctly tagged text, and a list is generated of tagging errors with their corresponding counters.
4. For each error, determine which instance of the generic template of rules produces the greatest error reduction.
5. The rule is applied, the new error set is computed, and the process is repeated until error reduction falls below a given threshold.


Untagged text
Initial tagger
Tagged text
Rules generator
Rules
Gold standard corpus

Outline

Efficiency considerations

- Brill tagger is considerably slower than taggers based on probabilistic models, both in the training and labeling phases
- Reason: the interaction between the rules, so that the algorithm may perform unnecessary computations
- Brill tagger could need to perform RKn elemental steps elementals to tag a sentence of n words, with R applicable rules in a context of up to K words.

Example of rule interaction

- Tags (Brown corpus):
  - `VBN`: verb, past participle
  - `VBD`: verb, past tense
  - `NP`: proper noun
  - `BEDZ`: the word `was`
  - `BY`: the word `by`
  - `PPS`: nominative pronoun, third person, singular
- `VBN` are `VBD` are the most likely tags for words `killed` and `shot`, respectively

Example of rules interaction

- After the lexical tagger:

(1) `Chapman/NP killed/VBN John/NP Lennon/NP`
(2) `John/NP Lennon/NP was/BEDZ shot/VBD by/BY Chapman/NP`
(3) `He/PPS witnessed/VBD Lennon/NP killed/VBN by/BY Chapman/NP`

- Errors:

in (1) `killed` wrongly tagges as past participle

in (2) `shot` wrongly tagged as past tense

Example of rules interaction

- The contextual tagger applies the rules:

  `VBN VBD prevtag NP` (*change the tag* `VBN` *to* `VBD` *if previous tag is* `NP`)

  `VBD VBN nexttag BY` (*change* `VBD` *to* `VBN` *if next tag is* `BY`)

- After applying the first rule the word `killed` in sentences (1) and (3) change its tag from `VBN` to `VBD`:

  (4) `Chapman/NP killed/VBD John/NP Lennon/NP`
  (5) `John/NP Lennon/NP was/BEDZ shot/VBD by/BY Chapman/NP`
  (6) `He/PPS witnessed/VBD Lennon/NP killed/VBD by/BY Chapman/NP`

Example of rules interaction

- After applying the second rule:

  the word `shot` in sentence (5) change its tag from `VBD` to `VBN`
  the word `killed` in sentence (6) change its tag from `VBD` to `VBN` again

  (7) `Chapman/NP killed/VBD John/NP Lennon/NP`
  (8) `John/NP Lennon/NP was/BEDZ shot/VBN by/BY Chapman/NP`
  (9) `He/PPS witnessed/VBD Lennon/NP killed/VBN by/BY Chapman/NP`

End of presentation