

Brill's Tagger from UNIX

Natural Language Understanding

CAP6640

Spring 2008

Overview

- Using UNIX
- Input to the tagger
- Running the tagger
- Output
- Rules of the game

Using UNIX

- Reference for basic commands:
 - <http://www.emba.uvm.edu/CF/basic.html>
- “cd”: Change directory
- “ls”: List directory contents
- “cp”: Copy file
- “passwd”: Change password

Using UNIX

- PuTTY: ssh client
 - <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
- Cygwin: “Linux-like environment for Windows”
 - <http://www.cygwin.com/>

Using UNIX

- On `monroe.cs.ucf.edu`:
 - AI Directory:
`"/a/ai"`
 - Brill's Directory:
`"/a/ai/new-guest/tagger/RULE_BASED_TAGGER_V1.14"`
 - Executables and Rule Files are in `./Bin_and_Data`
 - Lexicon: "LEXICON"
 - Lexical Rules: "LEXICALRULEFILE"
 - Contextual Rules: "CONTEXTUALRULEFILE"
 - Executable: "tagger" or shell script: "tag"

Using UNIX

- Adding aliases

- From home (~) directory: “pico .alias”

- Add to file (example):

```
alias dir ls -al
```

```
alias brill cd /a/ai/newguest/tagger/RULE_BASED_TAGGER_V1.14
```

- Using Scripts

- Example

```
#!/bin/sh
```

```
cd /a/ai/new-guest/tagger/RULE_BASED_TAGGER_V1.14/Bin_and_Data/
```

```
./tagger LEXICON /home/hschwartz/$1 BIGRAMS /home/user/LEXICALRULEFILE  
/home/user/CONTEXTUALRULEFILE >/home/user/out.txt
```

- The 1st line required to identify as shell script.

- The 3rd line runs Brill’s on rules in your home directory.

- “\$1” is a variable for line argument (“tagger file.txt”) \$1 <= file.txt

Input to the Tagger

- One sentence per line
- Spaces between punctuation (except for 's)
- “Double quotes” changed to ``two single quotes”
- Example:

example.txt

```
I am using "Brill's  
tagger". To use it  
correctly, would be  
ideal.
```



input.txt

```
I am using ` Brill 's tagger ' ' .  
To use it correctly , would be ideal .
```

*A script can do the conversion for you.

- lexicon, bigrams, lexical rule file, and contextual rule file are also input

Running the Tagger

“tagger <lexicon> <text-input> <bigrams> <lexical rule file> <contextual rule file>”

- Must run from same directory as “tagger” file.
- Simple “tag” script is also available in the Bin_and_Data directory
- Direct output to your directory:
 - “tagger ... >/home/user/output.txt”

Output

- Penn Treebank Tagset

- <http://www.mozart-oz.org/mogul/doc/lager/brill-tagger/penn.html>

- Example:

input.txt

```
I am using `` Brill ' s tagger '' .  
To use it correctly , would be ideal .
```

↓
Tagger ... input.txt ... >output.txt

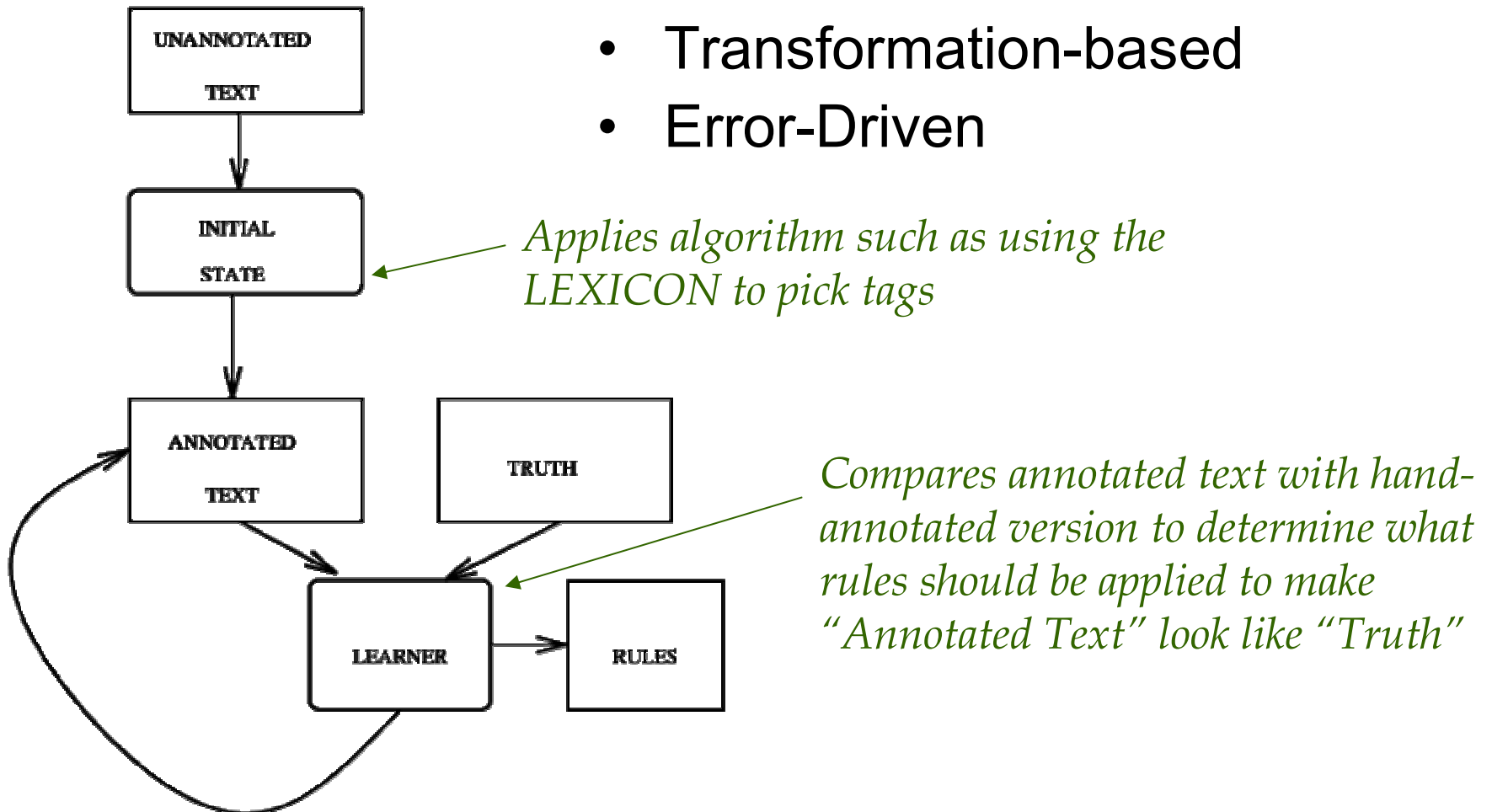
output.txt

```
I/PRP am/VBP using/VBG ``/`` Brill/NNP '/' s/PRP tagger/VBP ''/'' ./.  
To/TO use/VB it/PRP correctly/RB ,/, would/MD be/VB ideal/JJ ./.
```

Rules

TRAINING

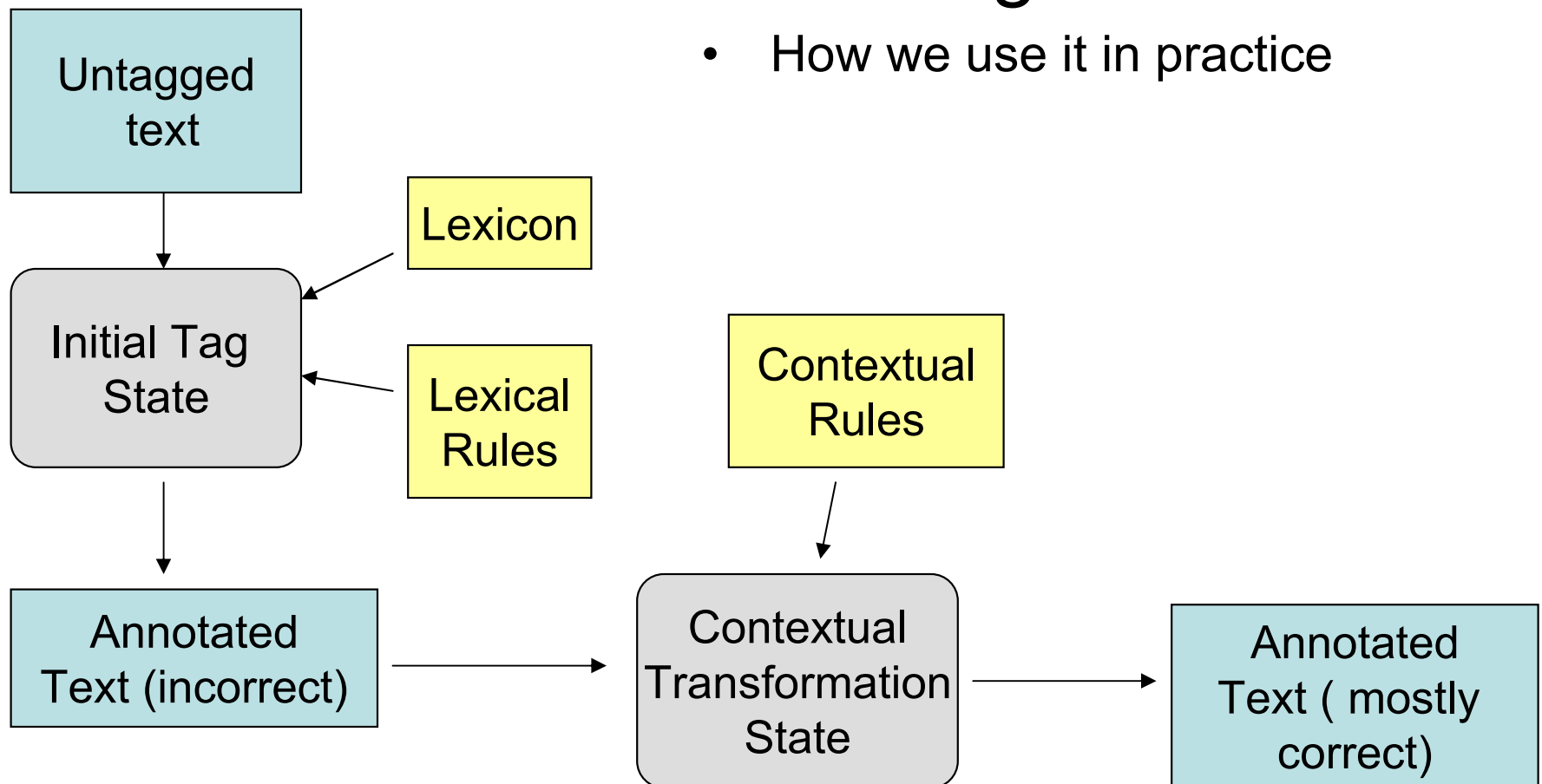
- Transformation-based
- Error-Driven



Rules

Testing

- How we use it in practice



Rules

- Lexical Rules

- Used to initially tag words which were not in the lexicon
- Unknown words initially marked as nouns

Change the tag of an unknown word (from X) to Y if:



- **NN s fhasuf 1 NNS** change the tag of an unknown word from NN to NNS if it has suffix -s
- **NN . fchar CD** change the tag of an unknown word from NN to CD if it has character '.'
- **NN - fchar JJ** change the tag of an unknown word from NN to JJ if it has character '-'
- **NN ed fhasuf 2 VBN** change the tag of an unknown word from NN to VBN if it has suffix -ed
- **NN ing fhasuf 3 VBG** change the tag of an unknown word from NN to VBG if it has suffix -ing
- **ly hassuf 2 RB** change the tag of an unknown word from ?? (any) to RB if it has suffix -ly
- **ly add suf 2 JJ** change the tag of an unknown word from ?? to JJ if adding suffix -ly results in a word
- **NN \$ fgoodright CD** change the tag of an unknown word from NN to CD if the word \$ can appear to the left
- **un delet pref 2 JJ** change the tag of an unknown word from ?? to JJ if deleting the prefix un- results in a word

Rules

- Contextual Rules

- Used for transforming tags based on context in a sentence.
- Can only change from a word from a tag in the lexicon to another tag in the lexicon – unless the word is unknown (not in the lexicon)

Change tag A to B when....

1. PREV --- previous(preceding)
2. PREVTAG --- preceding word is tagged
3. PREV1OR2TAG --- one of the two preceding words is tagged
4. PREV1OR2OR3TAG --- one of the three preceding words is tagged
5. WDAND2AFT --- the current word is x and the word two after is y
6. PREV1OR2WD --- one of the two preceding words is
7. NEXT1OR2TAG --- one of the two following words is tagged
8. NEXTTAG --- following word is tagged
9. NEXTWD --- following word is
10. WDNEXTTAG --- the current word is x and the following word is tagged z
11. SURROUNDTAG --- the preceding word is tagged x and the following word is tagged y
12. PREVBIGRAM --- the two preceding words are tagged
13. CURWD --- the current word is

Further Information

- README.* Docs
 - In /Docs directory.
- Brill's Papers
 - See class website:
www.cs.ucf.edu/courses/cap6640/