Segundo TP de Procesamiento de Lenguaje Natural

Autor: Martín Ezequiel Langberg

El siguiente trabajo tiene como objetivo analizar una herramienta para Tagging/Chunking sus resultados con diferentes tipos de corpora y conocer los problemas que presentan.

Aplicaciones utilizadas durante el trabajo:

- NLTK 3.0 (para Python 2.7)
- nltk-trainer https://github.com/japerk/nltk-trainer
- YamCha 0.33

1- Resultados PoS Tagging (PennTreeBank)

- Accuracy **WsjSubset**: 93.56%

Falla	Esperado	Encontrado
8.84%	'JJ'	'NN'
7.44%	'NN'	'JJ'
4.09%	'VBN'	'VBD'
3.67%	'VBG'	'NN'
3.21%	'JJ'	'NNP'

- Accuracy **Genia**: 79.38%

Falla	Esperado	Encontrado
34.88%	'NN'	'NNP'
7.36%	'JJ'	'NN'
6.16%	'NN'	'-NONE-'
6.07%	'NN'	'JJ'
3.82%	'('	':'

2- Resultados Chunking (CoNLL 2000)

- Precision: (Numero de Tags de Chunk Nominales Correctos Producidos) / (Numero Total de Chunk Tags Nominales Producidos) y
- Recall: (Numero de Tags de Chunk Nominales Correctos Producidos) / (Numero Total de Chunk Tags Nominal en el Gold Standard)

WsjSubset

Precision(Nominal): 89.37%

Recall(Nominal): 89.95%

Precision(Verbal): 83.64%

Recall(Verbal): 88.66%

Genia

Precision(Nominal): 84.32%

Recall(Nominal): 85.46%

Precision(Verbal): 71.75%

Recall(Verbal): 81.29%

Los resultados fueron obtenidos usando NLTK y el NLTK-trainer (ver `chunker.py` y `tagger.py`)

3- Evaluación de un chunker en español

Se utilizó YamCha para entrenar un Chunker con el corpus CESS-P-train, y se analizaron dos textos del corpus CESS-P-test. Los resultados fueron:

Resultados Español 1	Resultados Español 2
Precisión Nom: 92%	Precisión Nom: 95%
Recall Nom: 95%	Recall Nom: 96%
Precisión Ver: 100%	Precisión Ver: 100%
Recall Ver: 100%	Recall Ver: 98%