Proyecto No. 2 de PLN (Análisis sintáctico probabilístico)

Raúl Gutierrez de Piñerez Reyes *Ph.D*

November 10, 2017

1 Introducción

El análisis sintáctico por constituyentes se realiza bajo tres enfoques, el primero *shallow parsing* conocido como la extracción de constituyentes o de trozos (NP-chunking), el segundo enfoque *full parsing* tiene que ver con la estructura interna de una oración a través de su aceptación o rechazo sobre un lenguaje; el tercer enfoque trata de la generación de árboles sintácticos usando modelos pobabilísticos. Este taller se enfocará en los analizadores sintácticos probabilísticos; el analizador de stanford y el analizador de Bikel los cuales son PCFGs lexicalizados. El analizador de Bikel se basa en los modelos 1 y 2 de Collins [Col97a, Col97b] en el cual se define un modelo probabilístico muy sofisticado basado en el conteo lexicalizado de las reglas sintácticas que son generadas mediante el aprendizaje del corpus. El analizador de Stanford introduce el concepto de vectores gramaticales composicionales (CVG: Compositional Vector Grammar) que permiten la adición de información sintáctica y semántica al análisis de la sentencia. La importancia del método radica en la continuidad de los vectores a diferencia de la información discresta de los modelos usados en Collins [Col97a, Col97b], Bikel [Bik02] y Charniak [Cha97]. Es importante anotar que los CVGs actuan sobre los modelos PCFGs lexicalizados incrementando la precisión de los analizadores en un 3.8%. El problema del análisis sintáctico de una sentencia s se plantea como la búsqueda del árbol sintáctico con la mayor probabilidad en la cual el analizador retorna a:

$$t^* = \underset{t \in \mathcal{T}_G(s)}{argmax}\ \mathcal{P}(t) \tag{1}$$

t^* es el árbol sintáctico con la máxima probabilidad para la sentencia s.

2 Problema

Este proyecto consiste en resolver el problema del análisis sintáctico integrando las tareas de segmentación, tokenización, POS tagging y medida de desempeño. El proyecto tiene varias fases; una primera, la **fase de preprocesamiento** que consiste en segmentar, tokenizar el documento y luego tagear cada una de las sentencias del texto. Una segunda fase, **el análisis sintáctico probabilístico** usando los dos modelos (Stanford y Bikel) para el inglés sobre PTB. La tercera fase, es el uso del algoritmo **PARSEVAL** para el cálculo de las medidas de desempeño; precisión, recall y F1. Todas estas tres fases se deben integrar en una aplicación en la Web. Las siguientes son las recomendaciones que deberá tener la aplicación:

1. La interfaz web debe permitir escribir un texto en inglés de no más de 300 palabras.

2. Para cualquier un texto se debe realizar la fase de preprocesamiento, se debe desplegar el POS tag de cada una de las sentencias del texto de entrada.

3. Dado el texto se debe desplegar el análisis sintáctico de las sentencias del texto en forma parentizada bajo PTB y en la interfaz se deben presentar las dos opciones la de Stanford y la de Bikel.

4. Se debe desplegar el árbol sintáctico para cada árbol parseado (No desbordar).

5. Finalmente, para medir el desempeño de la anotación sintáctica del analizador de Stanford se utiliza un conjunto de árboles **Gold-standard** sobre PTB. El conjunto **Gold-standard** es la versión anotada de datos de testeo de PTB sin errores. Los datos de testeo en PTB son el conjunto de sentencias no anotados usados como entrada del parser (sección raw 23 de PTB con al menos 2400 sentencias que equivalen al 6% del conjunto de entrenamiento). Para medir el desempeño del analizador se usará el algoritmo **PARSEVAL** el cual mide la precisión, recall y f1 del conjunto de testeo. El desempeño se mide probando las sentencias del conjunto de testeo en raw text (test data) se produce el árbol parseado o analizado y se contrastan con los árboles Gold-Standard. Para cada sentencia anotada y la sentencia gold deberán ser entradas del algoritmo PARSEVAL. El desempeño se debe medir para los dos analizadores y se debe presentar una tabla de las medidas de desempeño (P, R y F1) por analizador, se deben probar sentencias de menos de 20 palabras y sentencias entre 20 y 30 palabras.

3 Insumos de trabajo

- El archivo de entrenamiento que produce el modelo y el obervado: wsj-02-21.mrg
- El analizador dbparser
- La carpeta de testeo y la gold standard
- Para el preprocesamiento se puede usar Stanford o Freeling pero se deberá implementar la estructura de parentizado que usa la entrada de Bikel
- El parser de Stanford en `http://nlp.stanford.edu/software/lex-parser.shtml`

4 Evaluación

1. Interfaz 20%
2. Preprocesamiento del texto 20%
3. Funcionamiento de los analizadores 20%
4. Medidas de desempeño 40%

References

[Bik02] Dan Bikel. Design of a multilingual, parallel processing statistical parsing engine. In Proc. of the Second International Conference on Human Language Technology Research HLT'02, 2002.

[Cha97] Eugene Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, AAAI'97/IAAI'97, pages 598–603. AAAI Press, 1997.

[Col97a] Michael Collins. New statistical parser based on bigram lexical dependencies. ACL, 1997.

[Col97b] Michael Collins. Three generative, lexicalised models for statistical parsing. ACL, 1997.