Team Ruby

Ben Nelson and Mike Roylance

Context and tools

Ruby: fluent apis, lambdas and collections of data are fun!

Github too!

Editors: Sublime2 / Vi

www.coursera.org/saas/class <- ruby on rails class

Development Strategy

Kanban development

Once weekly meetings

Individual work, email for any questions/concerns

Github for issue tracking

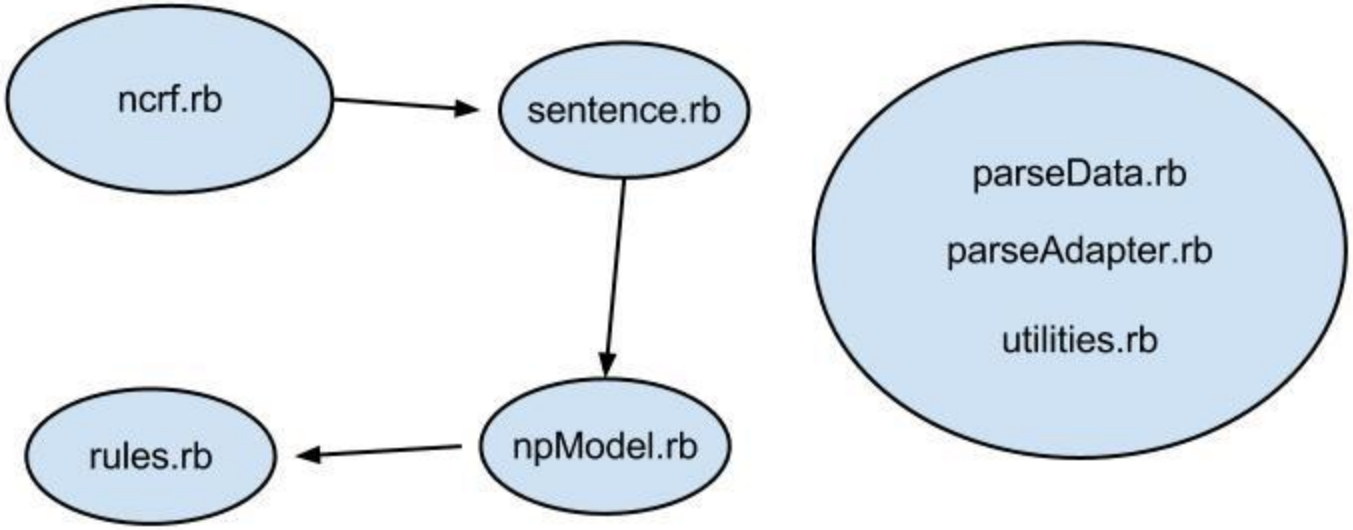
Architecture Overview

Based solution on "Noun Phrase Coreference as Clustering" by Cardie and Wagstaff

Started by simple string matching

Advanced heuristics later including editdistance, headnoun matching, pronoun agreement etc.

Loosely Typed is fun!



ncrf.rb

Represents the crf file.

Factory method handles processing listfile

Internal method handles I/O

Transforms XML to usable structures

Assigns new IDs for found NPs

Delegates how sentences are created

Delegates when the npModels should find coref

sentence.rb

Essentially a collection of words, and their corresponding noun phrases

Adding of text, coref elements and noun phrases

to_xml and to_s overrides

npModel.rb

Keeps track of identified noun phrase.

Ctor initializes properties such as:

- position in document
- pronounType
- semantic class
- head noun
- proper noun

Uses rules to find best match to antecedent

rules.rb

incompatibility functions (word substring, pronouns etc...)

all static, didn't want to instance this class

findCorrectAnt function runs through all the rules, counts the score and selects the best one

Some scores have -infinity, others +infinity

helper classes

parseAdapter.rb - handles Stanford Parser stuff

parseData.rb - parses the Stanford Parser stuff

utilities.rb - has editDistance and other algorithms.

External Resources

- Stanford Named Entity Recognizer
- Stanford Parser
- Rexml gem for XML parsing
- RSpec for Unit Testing
- SimpleCov for Code Coverage
- Screen-scraping (http://names.mongabay.com/) with Nokogiri gem for names used in gender recognition

Contributions

Mike:

- XML parsing / output
- 1/2 Feature Rules
- Debugging like crazy

Ben:

- Research tools and resolution strategies
- Code Testing
- 1/2 Feature Rules

Technique

- Using properties of Noun Phrases to determine resolutions.
- Emphasis on non-machine learning techniques.

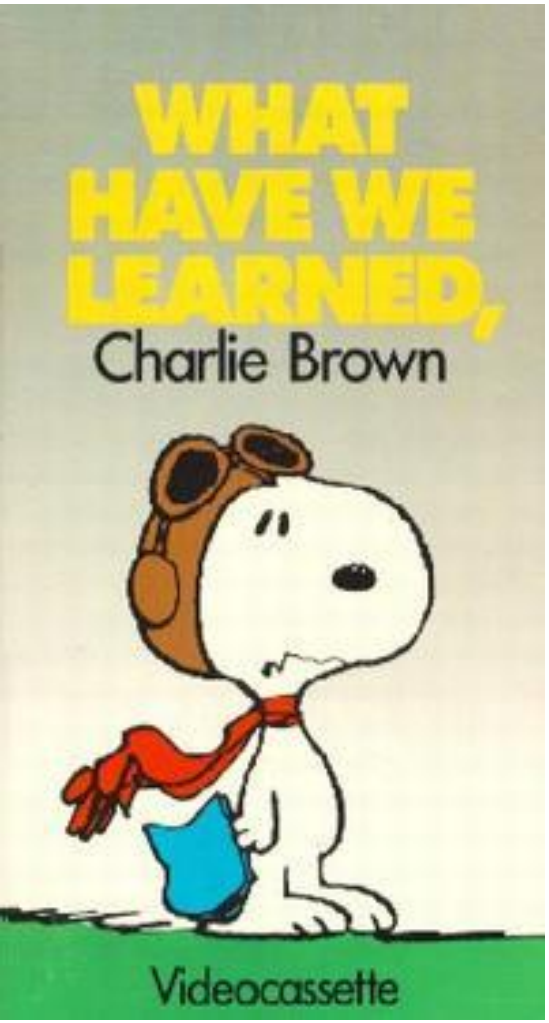
Results

53.50% devset

47.35% testset1

47.20% testset2

Lessons Learned



Most successful: String Matching

Least successful: Appositive

Natural Language Processing == Hard