LING 520: Computational Analysis of English

Semester: FALL '16

Instructor: Sowmya Vajjala

Iowa State University, USA

13 September 2016

Class outline

- Assignment 1 discussion
- Revision of tokenization and sentence splitting
- Assignment 2 description
- Text normalization and spelling correction: overview
- Problem Set 2 practice in Class

Assignment 1 Discussion

I need volunteers to discuss their solutions to Assignment 1.

Tokenization and Sentence Splitting: Revision

- Did you write your own tokenizer, sentence splitter, and test how they are doing?
- What were the patterns they missed?

Tokenization and Sentence Splitting: Revision

- Did you write your own tokenizer, sentence splitter, and test how they are doing?
- What were the patterns they missed?
- Did you explore tokenizing and sentence splitting options in NLTK?
- How do they work? Did you find cases where they fail?

Tokenization and Sentence Splitting: Revision

- Did you write your own tokenizer, sentence splitter, and test how they are doing?
- What were the patterns they missed?
- Did you explore tokenizing and sentence splitting options in NLTK?
- How do they work? Did you find cases where they fail?
- What is the difference between WordPunktTokenizer and PunktWordTokenizer?

Tokenization and Sentence Splitting: Revision

- Did you write your own tokenizer, sentence splitter, and test how they are doing?
- What were the patterns they missed?
- Did you explore tokenizing and sentence splitting options in NLTK?
- How do they work? Did you find cases where they fail?
- What is the difference between WordPunktTokenizer and PunktWordTokenizer?
- Did anyone check out NLTK's tokenizing and sentence splitting options for a non-English language?

Assignment 2 Description

File on Blackboard (3 Questions, 5 marks for each).
Deadline: 27 September.

Why/When are these different pre-processing tasks done?

1. lower casing

Why/When are these different pre-processing tasks done?

1. lower casing
2. tokenization, sentence splitting

Why/When are these different pre-processing tasks done?

1. lower casing
2. tokenization, sentence splitting
3. removing most frequent, or most rare words

Why/When are these different pre-processing tasks done?

1. lower casing
2. tokenization, sentence splitting
3. removing most frequent, or most rare words
4. stemming, lemmatization

Why/When are these different pre-processing tasks done?

1. lower casing
2. tokenization, sentence splitting
3. removing most frequent, or most rare words
4. stemming, lemmatization
5. text normalization: substituting contractions, abbreviations, spelling normalization and correction etc.

Why/When are these different pre-processing tasks done?

1. lower casing
2. tokenization, sentence splitting
3. removing most frequent, or most rare words
4. stemming, lemmatization
5. text normalization: substituting contractions, abbreviations, spelling normalization and correction etc.

How do you do these tasks?

1. removing most frequent or most rare words

How do you do these tasks?

1. removing most frequent or most rare words
2. If am not talking about English, think about any other language you know, and tell if the task is as simple.

How do you do these tasks?

1. removing most frequent or most rare words
2. If am not talking about English, think about any other language you know, and tell if the task is as simple.
3. substituting contractions

How do you do these tasks?

1. removing most frequent or most rare words
2. If am not talking about English, think about any other language you know, and tell if the task is as simple.
3. substituting contractions
4. substituting abbreviations

How do you do these tasks?

1. removing most frequent or most rare words
2. If am not talking about English, think about any other language you know, and tell if the task is as simple.
3. substituting contractions
4. substituting abbreviations
5. converting all dates into one standard form

How do you do these tasks?

1. removing most frequent or most rare words
2. If am not talking about English, think about any other language you know, and tell if the task is as simple.
3. substituting contractions
4. substituting abbreviations
5. converting all dates into one standard form
6. stemming

How do you do these tasks?

1. removing most frequent or most rare words
2. If am not talking about English, think about any other language you know, and tell if the task is as simple.
3. substituting contractions
4. substituting abbreviations
5. converting all dates into one standard form
6. stemming
7. lemmatization

Text Normalization

- Normalization refers to all forms of pre-processing that tries to bring text representations into some standard form (lower casing, substituting abbreviations etc.)
- Reason: makes comparison between documents/words easier
- normalization may look like a simple, straight forward task which can be done with regular expressions and string substitutions.
- However, there are several design issues. Here is a graduate level course on text normalization: `http://www.csee.ogi.edu/~sproatr/Courses/TextNorm/`
- Today's class: two forms of spelling normalization - soundex, edit distance.

Name normalization: Soundex Algorithm

- The purpose of name normalization methods is to capture different spelling variations of proper names.
- Soundex is one such method, which is a phonetic algorithm for English names. The goal is to group similar sounding names together. This is mainly useful in information retrieval from databases etc.
- Very first algorithm is almost a century old now!
- Simple rules, and straight forward mapping.
- Any surname gets converted to a code of a single character and three digits seperated by hyphen.

Soundex coding rules

http://www.archives.gov/research/census/soundex.html
(Use this to answer a question in Assignment 2)

Spelling normalization

The idea of "distance between words"

- "distance between words" refer to some way of quantifying how much two words are seperated from each other orthographically.
- This is useful in applications such as information retrieval (for capturing spelling variations), spelling suggestions (suggesting the closest possible alternative to an unknown word).
- Several measures of orthographic distance exist: https://en.wikipedia.org/wiki/Category:String_similarity_measures
- I will discuss one: Minimum edit distance, and introduce the concept of "dynamic programming" through that on thursday

Minimum edit Distance: Introduction

- Idea: minimum number of edits required to transform one word into another.
- What are edits: insertions, deletions, substitutions
- From Creep to Crap, there is one deletion (remove one e) and one substitution (second e to a)
- From Sleep to slept, there are: one deletion (delete second e), one insertion (insert t)

Minimum edit Distance: Introduction

- Idea: minimum number of edits required to transform one word into another.
- What are edits: insertions, deletions, substitutions
- From Creep to Crap, there is one deletion (remove one e) and one substitution (second e to a)
- From Sleep to slept, there are: one deletion (delete second e), one insertion (insert t)
- Alternative: 2 substitutions. Substitutions in edit distance metrics have more penalty though.

Other measures of Similarity between words

- distributional similarity: words that are used in similar contexts are perhaps related to each other
- other form of semantic similarity: computed based on the presence of large lexico-semantic resources like wordnet.
- http://wordnet.princeton.edu
- Chapter 2.5 in NLTK book has an overview of some such measures available in NLTK.
- Useful for some NLP problems like word sense disambiguation.

Next Class

- Continuation of spell check/correction discussion.
- Optional to do: Read "How to write a spelling corrector" by Peter Norvig (http://norvig.com/spell-correct.html)
- One announcement: On thursday, our class will take place in Ross 420.

Practice exercises

1. Figure out whether NLTK has a distance metric such as levenshtein or other such orthographic distances, and learn how to use one such measure to get distance between words.
2. Check for any python based spell checking libraries. If you do not find any, learn to use PyEnchant library for spell checking.
3. Start doing problems in Problem Set 2 (see Blackboard)