From Text Analysis to Actionable Knowledge: Data-intensive Methods for Unstructured and Text-heavy Data
Natural Language Processing

Hilke Reckman

Aarhus University & Unsilo

Tuesday Nov 29 2016

Natural Language Processing

Progress and opportunities (facilitated by ever faster computers)

Useful everyday applications: Google Search, Siri, autocomplete for SMS, Google Translate, . . .

This Lecture:

- How can Natural Language Processing help you analyze text
- What to consider when preprocessing your data

Division of labor: You and your computer

- Identify research question, data, methods (You)
- Collect data (You / You+Computer)
- Get familiar with your data (You, You+Computer)
  Formulate hypotheses (You)
- Make decisions on preprocessing, tools, and settings (You)
- Process large amount of data (Computer)
- Inspect results, interpret, indentify next steps (You)

Quality or quantity: don't tell me they're the same...
- Bad Religion

Hypothetical example: Study equal opportunity efforts

- Text data: company policy documents, interviews, company internal/outward communications, data reflecting broader societal debate, social media data about the company's reputation, ...
- Hypothesis: policy predicts outcome, company culture predicts outcome, outcome affects company reputation, ...
- Difference in content? Difference in style?
- Find documents / passages that are relevant to the topic
- What do you notice upon close reading? How could you find more evidence of that at scale?
- Zoom in and out (detail can be ignored in some steps, but is not lost)

What is a text?

- A coherent set of signs that transmits some kind of informative message
- A written representation of a message in natural language
- Words organized into meaningful sentences, organized into meaningful paragraphs...
- To the computer:
  a sequence of bytes that encode characters
  lower case letters, upper case letters, punctuation, spaces, ...
  "Text" $\Rightarrow$ \u0054\u0065\u0078\u0074

The NLP Pipeline: From characters to meaning

- Tokenization: from characters to words and sentences
- Normalization: reducing the number of distinct forms
- Part-of-speech tagging
- (Partial) syntactic processing: Structural analysis - how do the units combine together?
- (Partial) semantic processing: Interpretation

Also: filtering - what information to keep and what to throw out?

From characters to words

“This is a sentence.”
\u0054\u0068\u0069\u0073\u0020\u0069\u0073\u0020\u0061
u0020\u0073\u0065\u006e\u0074\u0065\u006e\u0063\u0065\u002e

- Tokenization / segmentation
  - Word tokenization: split into words
    needed for most types of processing
  - Sentence tokenization: split into sentences
- Split on white space, punctuation?
- Keep or discard punctuation?

Tokenization challenges

- abbreviations:
  - '.' is part of the word
  - if the abbreviation is at the end of the sentence, the period does double duty
- apostrophies, hyphens, contractions *'won't'*
- proper names with unusual characters *'M*A*S*H'*
- inconsistent use of punctuation and white space *'lower case'*, *'lowercase'*, *'lower-case'*
- compounds (German, Danish, Dutch, . . .)
- multi-word units (words with spaces)
- emoticons
- hashtags, urls, dates, words containing numbers, . . .

Specialized tokenizers exist, e.g. for social media text.

Tokens, Types, and Normalization

- Tokens can be sorted into types: distinct forms
- The token is then an instance of the type
- Tokens that are identical belong to the same type (but word-sense ambiguity!)
- Sometimes it makes sense to map tokens that are not identical to the same type
  - normalize capitalization (common: lowercasing) *'The'*, *'the'* but: *'PET'*, *'pet'*, ambiguity: $A \rightarrow a$ / *à* (fr)
  - Spelling correction/normalization
  - Lemmatization: map inflected words to their lemmas (usually dictionary-based) *'use'*, *'uses'*, *'used'*, *'using'*
  - Stemming: map inflected and derived words to their stems (usually rule-based) *'user'*, *'usage'*, *'usability'*, ...
    more reduction than lemmatization, but also more errors
  - Mapping synonyms

Multi Word Units, Collocations

- Words with spaces (*'by and large'*, *'San Francisco'*)
- Constructions with syntactic variation (*'kick the bucket'*, *'make one's way'*, *'call off'*)
- Terms that are more loosely connected

Which terms occur together significantly more often than you would expect if they were distributed randomly over the text?

You can also use this to find terms that are associated with a particular category of text.
Top terms associated with positive sentiment tweets:
!, :), :, good, Happy, fun, ..., happy, ?, great, Good, amazing, -, :D, love, Looking_forward_to, <3, ;), tonight, Great, better_than, LOVE, Can, Great, birthday, perfect, cool, awesome, the_best, glad, I_love, the, song, exciting, love_you, Happy_Birthday, best, LOL, nice, excited, Thank_you, luck, ever, said, so_much, a_good, excited_for, Yay, Hopefully, a_great

Part of Speech tagging

- Parts of Speech (PoS): Noun, Verb, Preposition, etc.
- Can be useful in refining search, filtering
- Ambiguity: identical looking words can have different PoS (systems use context)
- Reasonable PoS-tagging systems are available for many languages
- Basis for further syntactic processing

Syntactic processing: Chunking & Parsing

- Chunking: e.g. extracting noun phrases
- Dependency parsing
- Parsing is difficult (Existing parsers work OK-ish on English news wire text)
- Consider work-arounds with PoS-tagging and chunking

One morning I shot an elephant in my pajamas.
How he got into my pajamas I'll never know.
-Groucho Marx

Semantic processing

- **Semantic role labeling**
- **Named entity recognition**
- **Coreference resolution**
- **Sentiment analysis**
- Document similarity
- **Word similarity**
- Topic modeling
- Textual entailment, question answering, information extraction
- Lexical semantic resources: ontologies, thesauri (e.g. WordNet)

Named Entity Recognition & Coreference Resolution

- Identify names of people, places, organizations, ...
- Identify dates, monetary amounts, ...
- Which names refer to the same entities?
- What do pronouns like *'he'*, *‘she’*, *‘it’*, *‘this’* refer to?

Very useful, but very challenging!

Semantic Role Labeling

- Identify roles independent of syntactic construction (e.g. object becomes subject in passive)
- Different types of verbs assign different roles: e.g. agent vs experiencer
- Needs parsing (or at least advanced chunking)
- Possible TM relevance: Which entities are depicted as active, which as passive? (political discourse, gender studies, . . .)

Sentiment Analysis

- Detect sentiment expressed in a text: in general or about something specific (a brand, a product, a politician, . . .)
- Often based on lists of positive and negative words
- Better: phrases (e.g. *'not so nice' 'Yeah, right!'*)
- Or machine learning
- Problems:
  - Syntactic analysis may be needed: in some languages (e.g. German) negations can be quite far away from the term they are negating
  - What is sentiment? Some words have positive or negative associations, but are not necessarily always expressions of sentiment (*'death'*)
  - Ambiguity, e.g. slang expressions like *'sick'*
  - Irony and sarcasm
  - Book / movie reviews: distinguish between description of content and evaluation
  - In machine learning, best predictors sometimes not sentiment words, e.g. certain actor names in movie reviews

Word Similarity with Distributional Semantics

Words that occur in similar contexts have similar meanings

Text Mining with a Seed List:

- Make a list of terms related to a concept/topic that you are interested in
- Use a Vector Space Model to find similar terms and expand the list
- Use the expanded list to extract passages for further study
- Refine if needed

Supervised Machine Learning for Classification

Can a machine tell the difference?

- Labeled data
- Features: Which information could be important for correct classification?
- Train a classifier
- How well can the classifier distinguish between the categories, based on the extracted features?

Concluding remarks

- Segmentation is needed for most task (and will not be perfect)
- Normalization can help with sparse data, but will also introduce errors
- Annotion/disambiguation can help make useful distinctions, but is not error-free either
- Using PoS-tagging is often a good idea, as well as extracting collocated terms
- More advanced syntactic and semantic processing technologies can be tempting but are often not mature and should be used with care
- Supervised machine learing can help you test hypotheses about how categories of text differ from each other