Room for Improvement: Aspect-Specific Statistical Opinion Mining of Online Hotel Reviews

Lynd D. Bacon
Loma Buena Associates
Notre Dame Mendoza College of Business
Northwestern University

August 1, 2016

Joint Statistical Meetings
Chicago IL USA
Online Reviews

• Poorly structured data- mix of:
  – quantitative measures
  – text and/or other poor- or unstructured data
  – metadata on time, date, etc.

• Many examples:
  – Amazon
  – Trip Advisor
  – Product forums of various kinds, e.g. Edmund's
The Challenge(s) of NLP of User Generated Content

- Loosely (or un-) controlled vocabulary
- Coreferencing
- Colloquialisms
- Emoticons
- Slang
- Sarcasm
- Negation
- The NLP “evil twins:”
  - polysemy
  - semantic ambiguity
“McDonald's Fries the Holy Grail for Potato Farmers”
That Evil Semantic Ambiguity

“Time flies like an arrow.

Fruit flies like a banana.”

- Groucho Marks

news headline examples:
(some courtesy Nick Haddock)

“Kids Make Good Snacks”

“Ban on Nude Dancing on Governor's Desk”

“Iraqi Head Seeks Arms”

“Hospital Sued by 7 Foot Doctors”

“Death Happens More Slowly Than Thought”
The Data

- Scraped by Wang, Lu, and Zhai (2010, 2011) to study ratings of experience aspects
  - https://www.cs.virginia.edu/~hw5x/dataset.html
- Number of reviews: 1,609,233
- Number of hotels: 12,773
- Number of “tokens” (non-whitespace) in reviewer comments: 1,332,895 (But it depends...)

Example Review (JSON format)

{"Author": 'PDXKev',
 'AuthorLocation': 'Portland, Oregon',
 'Content': "I stayed here overnight after attending the U2 concert recently. It was really bad! They didn't even have an alarm clock in the room. It's way overpriced. About the only thing good is that the location is quite convenient.",
 'Date': 'June 7, 2011',
 'Ratings': {"Cleanliness": '1',
              'Overall': '1.0',
              'Service': '1',
              'Sleep Quality': '2',
              'Value': '1"},
 'ReviewID': 'UR112081499',
 'Title': ""Awful, Simply Awful""}
The Project

• Phase I
  - tokenization and EOS (end of sentence) detection
  - POS tagging

• Phase II
  - chunking
  - “aspect” recognition
  - Aspect “sentiment scoring”
Some of The Challenges

• Sorting out the “vocabulary” w.r.t. EOS, POS
  - Reviews aren't in the “King's English:”
    • slang, emoticons, acronyms, initialisms, “poor” grammar
    • The VADER lexicon (Hutto & Gilbert, 2014) has ~7,500 of such “non-standard”
    • What's a sentence?
      - is an emoticon a “sentence?” If at all, always?

• (All) The other ones...
Tokens (of sorts)

- Total: 1,332,895 tokens
  - After mangling some emoticons, separating periods, splitting on whitespaces
  - Words and non-words (not lemmatized)
- 10 most prevalent:
  - [', 'the', 'and', 'a', 'to', 'was', 'in', 'of', 'I', 'is']
- VADER sentiment tokens: 4,924
  - out of 7,517
  - 10 most prevalent:
    - ['great', 'good', 'nice', 'no', 'clean', 'like', 'well', 'friendly', 'helpful', 'comfortable']
  - 10 least:
    - ['challengingly', 'successively', 'uglified', 'tmi', 'dismaying', 'rebels', 'resentfulness', 'doomsayers', 'ignorances', 'rebelling']
  - 74 w/o any alphas, e.g. [':', ':-)', ';)'],(':(', ':-)', '8)'), '(8', '="'), ':D', ':o'), ':-(', '86', etc.
  - 19,144 :)
Vagaries of the Vocabulary and POST

- Tokenize w/o eliminating emoticons
  - Whitespace tokenize
  - Tag emoticons, etc. as a “emotion” speech type for later processing.

- The POST challenge: No training data, so
  - as a “kluge,” train on one (or more) tagged corpuses
  - then tag any emoticons
  - default tag unknowns as prevalent type in corpus
POS tagging

- POS taggers are either rule-based or “learner” based
- Learners are trained on a corpus, may or may not generalize well to other corpora
- Many learners are “sequence learners,” predicting likely POS based on some degree of context
Sequence Learners

- Typically used for part of speech (POS), labelling, named entity recognition (NER)

- Examples:
  - Hidden Markov Models
  - Maximum Entropy Markov Models
  - Normalized transition-based neural networks
  - Conditional Random Fields
Conditional Random Fields

- Used for labeling network nodes
  - *Sequences* are a special case: sequences of nodes
- Finite state model with unnormalized transition probabilities
- Convex loss function
- Various estimation methods:
  - ML, MAP, Broyden/Fletcher/Goldfarb/Shanno (BFGS), stochastic gradient descent, gradient tree boosting
- Lafferty, McCallum & Pereira (2001)
  - ML by iterative scaling
- See Wallach (2004) for an introduction
The Basic CRF Model
(as used for sequence learning)

- X: random variable over data to be sequenced
  - X might be NL sentences
- Y: random variable over label sequences
  - Y might range over POS taggings or NER labels
  - All $Y_i$ are in a finite set of labels
- $X, Y$ are jointly distributed
CRF Definition (cont.)

- $G = (V,E)$ a graph
- $Y = (Y_v)_{v \in V}$ indexed by $V$
- Then $(X,Y)$ is a “Conditional Random Field”
- When conditioned on $X$, $Y_v$ is Markov w.r.t. $G$:

\[
p(Y_v | X, Y_w, w \neq v) = p(Y_v | X, Y_w, w \sim v)
\]

where $w \sim v$ means $w$ and $v$ are neighbors $\in G$
Joint Distribution of Y over X

\[ p_\theta(y|x) \propto \exp\left( \sum_{\epsilon \in \mathcal{E}, k} \lambda_k f_k(\epsilon, y|\epsilon, x) + \sum_{\nu \in \mathcal{V}, k} \mu_k g_k(\nu, y|\nu, x) \right) \]

where:
- \( x \) is a data sequence
- \( y \) is a label sequence
- \( y|_S \) is the components of \( y \) associated with the subgraph \( S \)
- \( \theta = (\lambda_1, \lambda_2, ..., \mu_1, \mu_2, ...) \) are parameters to be estimated
“Label Bias”

- A weakness of some sequence learning methods
- Can ignore information when nodes have only one subsequent edge
Tools

- Python 3, Anaconda distribution
- MongoDB
- VADER lexicon
- Trying to be as DRY as possible, so (in Python)
  - NLTK (incl. CRFSuite), NLTK-trainer
  - TextBlob
  - pandas
  - collections
- Bits and pieces of custom code, the usual (ever growing) set of regex's, etc.
To-Date Approach to EOS, POST

- Pre-process entire corpus:
  - mangle a few emoticons to replace spaces
  - replace Vader non-alpha emoticons with tokens E0-E73
  - set off single puncts “. ! ?” to be EOS tokens

- Select a random 600 comments for initial testing

- Train and test CRF tagger using Treebank

- Apply CRF tagger to 600 comments
  - L-BFGS algorithm
  - back-off tagging for VADER emoticons
(Very) Preliminary Results

• 200 randomly selected, manually tagged comments
• Overall Accuracy: 0.643
• Measures based on word type classes:
  – “Noun” Precision: 0.720
  – “Noun” Recall: 0.741
  – “Noun” F1: 0.730
  – “Verb” Precision: 0.683
  – “Verb” Recall: 0.692
  – “Verb” F1: 0.687
What's Next

- Bootstrap a larger project-specific corpus for training purposes
  - Combine corpuses
- Sentence chunking
- Aspects identification
  - Not a sequential learning task
  - Could be by:
    - topic modeling (e.g. via LDA)
    - topic mapping (network clustering w/ LDA)
      - Lancichinetti et. al (2015)
References


Bird, Steven, Ewan Klein, and Edward Loper (2009), Natural Language Processing with Python, O'Reilly Media. (See also www.nltk.org for updated Python 3 version, also https://github.com/nltk)


