Tell me a good joke !

Evelina Dumitrescu, SCPD
Marian Bou, IA

March 26, 2015

Contents

- Project description
- Current approaches in literature for sentiment analysis
- Our approach
- Accuracy of our solution

Project description

- We want to determine what makes a good joke based on users' comments
- A problem of sentiment analysis and classification
- positive/negative/neutral comments
- Tell how funny a joke is
- Corpus based on comments from joke pages available on Facebook(ie: 9GAG, Joke Of The Day)

Current approaches in literature for sentiment analysis

- Lexicon based technique
- Noisy emoticons labels
- Machine learning methods

Current approaches in literature for sentiment analysis (2)

Lexicon based technique

- Take individual words/concepts determining the emotion by obtaining word polarities from a lexicon
- Domain independent/specific
- Part of Speech (POS) information indicated in polarity lexicons to overcome word-sense ambiguity; some parts might be more relevant (ie: adjectives, adverbs)
- Bag of words: SentiWordNet [1]
- Bag of concepts: SenticNet [2]

[2] http://sentic.net/

[1] http://sentiwordnet.isti.cnr.it/

Current approaches in literature for sentiment analysis (3)

Noisy emoticons labels

- Relevant for short text messages => one can assume that an emoticon withing a message represents an emotion for the whole message and all the words of the message are related to this emotion
- Happy: ":-)", ":)", "=)", ":D
- Sad: ":-(", ":(", "=(", ";("
- Neutral: ":|"

Current approaches in literature for sentiment analysis (4)

Machine learning methods

- Multinomial Naive Bayse
- TF-IDF scores
- Support Vector Machines
- Maximum entropy
- etc

Our approach

Corpus creation

- Extract comments from Facebook pages using Facebook's Graphs API and create the corpus
- Compose a "bag of words" ("bag of cocepts" would me more desirable)
- Consider emoticons as words
- Pay attention to repeated letters (ie: "happppy"), negations, contractions, eliminate stopwords
- Use Natural Language Toolkit(NLTK) [3]for tokening/stemmatization

[1]http://www.nltk.org/

Our approach (2)

Form the training data set

- Split the processed comments into positive and negative/neutral categories
- For each processed word compute the relevance for the category that it belongs
- Use TF-IDF(term frequency–inverse document frequency) score:
- “laugh”, “funny”, “:)” $\rightarrow$ relevant for a positive comment
- “cry”, “horrible”, “disgusting”, “:((” $\rightarrow$ relevant for a negative comment

Our approach (3)

Classify new comments

- Multinomial Naive Bayes
- In order to classify new comments, take into account also the TF-IDF score, not just the probabilities $\rightarrow$ how frequent occurs the term in a specific class, but also how relevant it is
- "Transformed Weight-normalized Complement Naive Bayes (TWCNB)"
- Scikit-learn [4]framework already has this implemented

[1]scikit-learn.org/

Our approach (4)

Tell if the joke is funny

- Count the number of positive/negative comments to evaluate the quality of the joke
- Evaluate how funny was the joke perceived:
- For each comment compute $\sum_{word \in Words} Polarity_{word} * TFIDF_{word} \rightarrow$ the score is weighted by the positive/negative polarity, but also by the relevance for the positive category
- Use SenticNet [5]for polarity; include additional polarities for emoticons Examples:
- "think" and "funny" have positive polarities of 0.061 and of 0.619; however, "think" won't be as relevant as "funny"
- "ugly" has a negative polarity of -0.581

[1]view-source:http://sentic.net/api/en/concept/

Accuracy of our solution

- Use testing samples for comments and see if classifier's prediction corresponds to our judgement $\rightarrow$ identify false positive (FP), false negative (FN), true positive (TP) and true negative (TN) cases
- Compute F1, precision (positive predictive value), recall (sensitivity, true positive rate) scores for measuring the efficiency
- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F1 = 2 * \frac{Precision*Recall}{Precision+Recall}$

References

- Taboada, Maite, et al. "Lexicon-based methods for sentiment analysis." Computational linguistics 37.2 (2011): 267-307.
- Gonçalves, Pollyanna, et al. "Comparing and combining sentiment analysis methods." Proceedings of the first ACM conference on online social networks. ACM, 2013.
- Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." LREC. Vol. 10. 2010.
- Poria, Soujanya, et al. "Sentic patterns: Dependency-based rules for concept-level sentiment analysis." Knowledge-Based Systems 69 (2014): 45-63.
- Read, Jonathon. "Using emoticons to reduce dependency in machine learning techniques for sentiment classification." Proceedings of the ACL Student Research Workshop. Association for Computational Linguistics, 2005.

References (2)

- Pak, Alexander, and Patrick Paroubek. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." LREC. Vol. 10. 2010.
- Gezici, Gizem, et al. "Su-sentilab: A classification system for sentiment analysis in twitter." Proceedings of the International Workshop on Semantic Evaluation. 2013.
- Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." Foundations and trends in information retrieval 2.1-2 (2008): 1-135.
- Rennie, Jason D., et al. "Tackling the poor assumptions of naive bayes text classifiers." ICML. Vol. 3. 2003.

Questions

?