Michael Superdock
Winnie Ngo

Lab 8 Write-Up

The SemEval-2014 Task 9: Sentiment Analysis in Twitter paper summarizes and analyzes the results of the 46 teams that contributed submissions to subtask A and subtask B. The goal of the SemEval task 9 is to create classification systems that better understand how sentiment is conveyed in informal text genres found in Social Media. They created the SemEval Tweet corpus which contains tweets and SMS messages with sentiment expressions annotated using both contextual *phrase-level* and *message-level* polarity. The 2014 test set contained regular tweets, sarcastic tweets and LiveJournal sentences. 2013 test sets were also included. Each team was allowed to submit results for a constrained or unconstrained system per subtask. Teams were scored based on their performance on Subtask A and/or Subtask B. Subtask A tested a classifier's ability to determine whether a marked instance in a given message is positive negative or neutral. Subtask B tested a classifier's ability to determine the positive, negative or neutral sentiment of a given message. On the Twitter test, *NRC-Canada* obtained the 2014 best score for subtask A and subtask B. On the Progress SMS test, *ECNU* obtained the best score for subtask A and *NRC-Canada* outscores in subtask B. The most popular classifiers used were SVM, MaxEnt and NaiveBayes. NLP preprocessing methods such as tokenization, stemming, lemmatization, stop-word removal and POS tagging combined with Twitter-specific processing such as substitution/removal of URLs, substitution of emoticons, word normalization, abbreviation lookup and punctuation removal. Notably, for most teams there was a significant drop in performance for sarcastic tweets, and the best results for the Twitter test were lower than those for the 2013 Twitter test on both subtasks.

The NRC-Canada-2014: Recent Improvements in the Sentiment Analysis of Tweets paper describes the classification systems used by the NRC-Canada team in their SemEval-2014 submissions. Their submissions obtained the highest score in the term-level sentiment classification subtask on the tweets test set, and in the message-level sentiment classification task on the LiveJournal blog posts and sarcastic tweets. Their approach used an in-house implementation of an SVM with a linear kernel. Their features included lexicon features, n-grams, negation, POS, cluster features and encodings. They obtained significant improvements over their previous SemEval-2013 approach by estimating the sentiments of works in affirmative and negated contexts separately using two tweet-specific sentiment lexicons: *Affirmative Context Corpus* and *Negated Context Corpus*. They split the tweet corpus into two parts, creating separate sentiment lexicons for the affirmative and negated contexts, which removed the need to employ explicit assumptions to model negation. The negated context lexicons are further split into *immediate context* and *distant context*. The immediate context consists of tokens that directly following a negation word while the distant context consists of the rest of the tokens in the negated context. Thus, a sentiment word can have up to three scores, for affirmative context, immediate negated context, and distant negated context. Notably, their results show that their models generalize well to data from other domains.

The TeamX: A Sentiment Analyzer with Enhanced Lexicon Mapping and Weighting Scheme for Unbalanced Data paper describes the system used by TeamX in SemEval-2014 Task 9 Subtask B which outperformed all others. Their system consists of a variety of pre-processors, feature extractors, and a machine learner. These tools were used together by TeamX to develop a sentiment analyzer that was based on a supervised text categorization approach. The pre-processing methods for their sentiment analyzer involved a text normalizer--which made use of case-folding, unicode normalization, and url replacement--a spelling corrector (Jazzy), multiple POS taggers, a word sense disambiguator, and a negation detector. Their feature extractor identified various features involving the use of word ngrams, character n-grams, tweet clustering, word senses, and word-to-lemma mappings. Their machine learner used Logistic Regression as an algorithm for supervised learning, and using the aforementioned features, it trained in order to label whole tweets according to their polarities. By the introduction of a prediction adjuster, they were also able to bias the output of their Logistic Regression algorithm in order to handle the unbalanced data provided to them. By using various weights for their prediction adjuster and by varying the cost parameter of their Logistic Regression, they found that their system was able to perform well so long as these parameters were adjusted according to the specific data set. However, they were unable to find a parameter combination that generalized well to all message types.

5. **New Classifier Additions:**
    1. text normalizer
        a. remove words with ‘@’, which usually indicate Twitter handles
        b. remove URLs

From NRC-Canada (Zhu et al.)

2. word n-grams
3. POS tagger

From TeamX (Miura et al.)

4. text normalizer
    a. case-folding
    b. URL removal
5. a spelling corrector
    a. spell-check words in tweet (excluding URLs) using Jazzy

From RTRGO (Gunther et al.)

6. text normalizer
    a. ‘#’ removal from hashtags
7. a features weighting scheme - all features receive weight 1 if they are present and weight 0 if they are absent

a. special cases:
    i. the original (non-normalized) token is all uppercase (+1)
    ii. the original token has more than three adjacent repetitions of one letter (+3)
    iii. the token is an adjective or emoticon, according to its POS tag (+3)
b. if token is in a question context, divide weight in half

Results

Unigrams, Conflate

Tokenize		X			X	X	X	X	X	X
Remove URLs					X	X				X
Remove @'s							X	X	X	
Stopwords			25	50						
Spellcheck									X	X
Remove '#'						X		X	X	X
Accuracy:	61.34	63.24	60.93	60.89	63.50	63.44	63.28	63.30	63.51	63.80

Tokenize +
Remove URLs +
Remove @'s 0
Stopwords -
Spellcheck +
Remove '#' 0

Tokenize, Remove URLs, Remove '#' (Unigrams, Conflate)

Upper +1		X				X	X	X	
Repetition +3			X		X		X	X	
Adjective +3				X	X	X		X	
Accuracy:	63.50	63.56	63.59	64.90	65.02	64.86	63.68	64.91	

Upper +
Repetition +
Adjective ++

Repetition, Adjective (Tokenize, Remove URLs, Remove ‘#’, Unigrams, Conflate)

Lowercase		X		X
Negate			X	X
Accuracy:	65.02	65.62	65.56	66.09

Lower +
Negate +

<u>Summary - Best Performing Classifier Modifications</u>
Unigrams, Conflate
Tokenize, Remove URLs, Remove ‘#’
Repetition, Adjective
Lower, Negate

Test ngrams

n-grams	1	2	3	discontinuous 3	4	discontinuous 4	5
Accuracy:	66.09	65.28	65.22	64.75	65.60	62.83	65.53

Unigrams