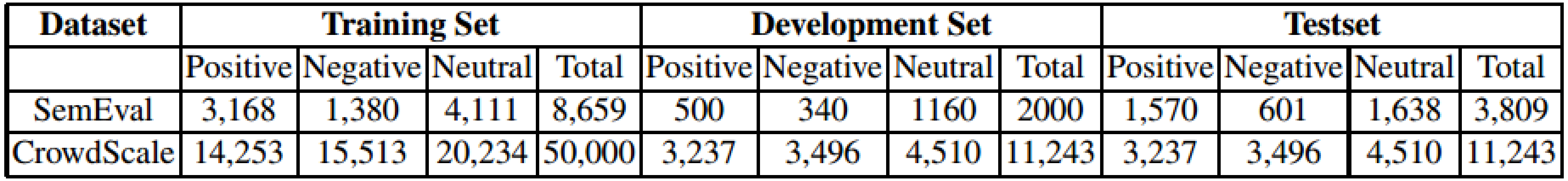
Sentiment Analysis Study

Hani Amr
Software Engineer

Overview

Datasets:



Dataset	Training Set				Development Set				Testset			
	Positive	Negative	Neutral	Total	Positive	Negative	Neutral	Total	Positive	Negative	Neutral	Total
SemEval	3,168	1,380	4,111	8,659	500	340	1160	2000	1,570	601	1,638	3,809
CrowdScale	14,253	15,513	20,234	50,000	3,237	3,496	4,510	11,243	3,237	3,496	4,510	11,243

Classifiers:

Two-class logistic regression

Technology:

Azure Machine Learning Studio, R, Python

Data Preprocessing

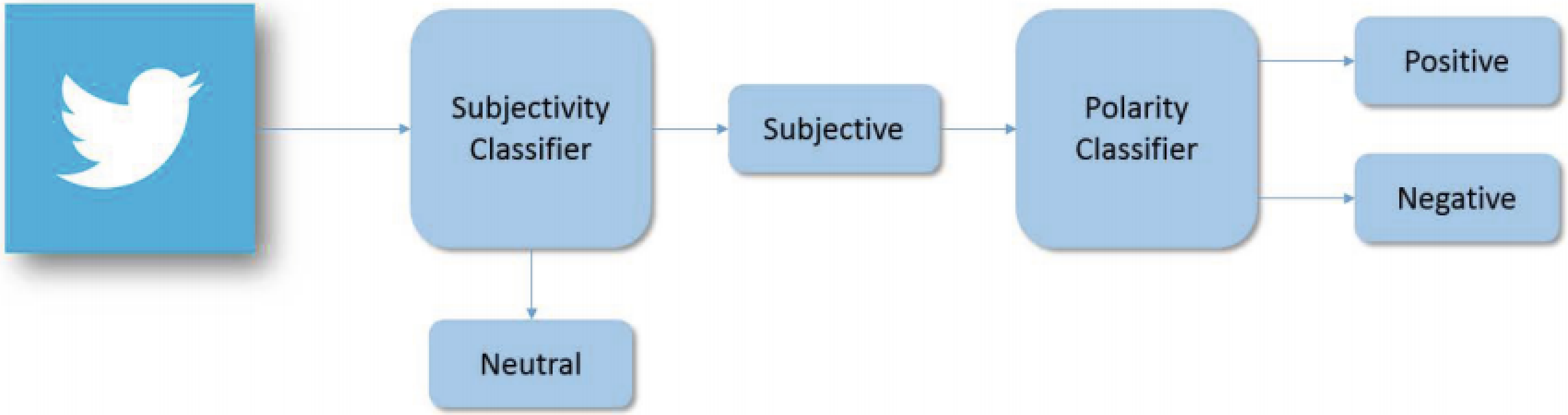
Cleaning text:

- Change negation words (not, cannot, never, etc.) to “not”
- Remove numbers, Unicode characters, URLs and stop words
- Replace more than 2 consecutive characters by only 2 (i.e. heeeey heey)
- Replace emoticons by their polarity (i.e. :=) positive_emoticon)

Stemming using Porter’s algorithm

Tokenization using CMU tokenizer

Pipeline approach



Feature extraction

Baseline features

N-Grams:

- Uni/bigrams using binary word presence
- Apply Log Likelihood scoring to select the top 20K ngrams

Example:

N-Gram	Score
hate	235.0607
love	200.4914
positive_emoticon	167.5600
spain	111.5650
ger	106.3451
Germany	93.3149
amazing	91.9482
support	90.7155

Senti-Features

Polar features:

- # of (+/-) POS (JJ, RB, VB, NN)
- # of negation words, positive words, negative words
- # of positive and negative emoticons
- # of (+/-) hashtags and capitalized words
- For POS JJ, RB, VB, NN, sum of polarity scores
- Sum of prior polarity for all words

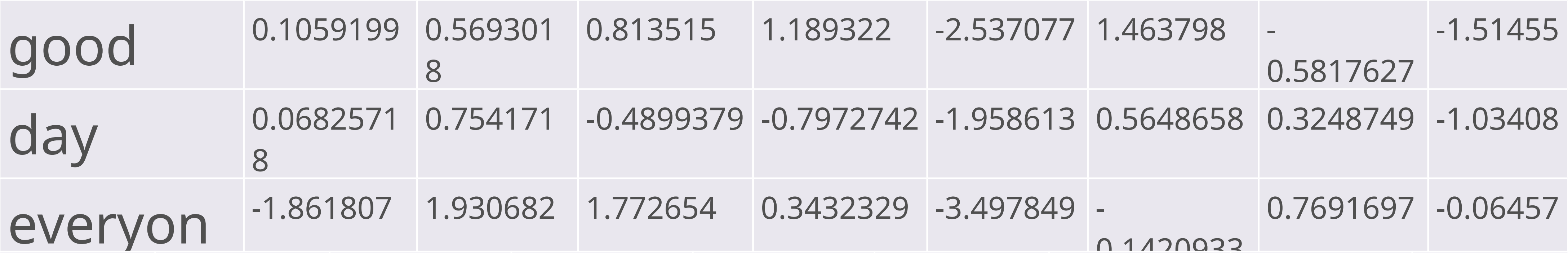
Non-polar features:

- # of JJ, RB, VB, NN
- # of slangs, Latin alphabets, dictionary words, words
- # of hashtags, URLs, mentions
- Percentage of capitalized text
- Exclamation, capitalized text

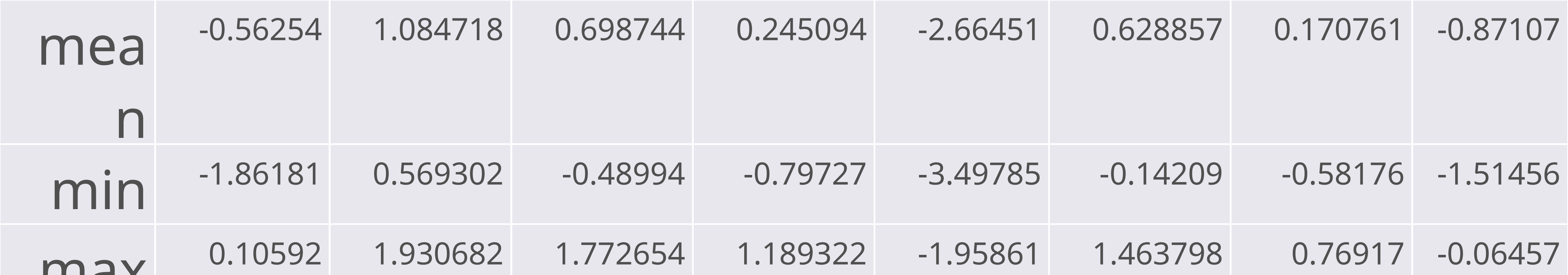
Sentiment Specific Word Embedding

- RNN trained on 10M auto-labeled tweets using emoticons
- For each tweet, calculate the mean/min/max and generate a feature vector of size 150
- Example:

Tweet: good day everyone!



good	0.1059199	0.5693018	0.813515	1.189322	-2.537077	1.463798	-0.5817627	-1.51455
day	0.0682571 8	0.754171	-0.4899379	-0.7972742	-1.958613	0.5648658	0.3248749	-1.03408
everyon	-1.861807	1.930682	1.772654	0.3432329	-3.497849	-0.1420933	0.7691697	-0.06457



mean	-0.56254	1.084718	0.698744	0.245094	-2.66451	0.628857	0.170761	-0.87107
min	-1.86181	0.569302	-0.48994	-0.79727	-3.49785	-0.14209	-0.58176	-1.51456
max	0.10592	1.930682	1.772654	1.189322	-1.95861	1.463798	0.76917	-0.06457

NRC Features

Counting features:

- All-caps words
- Number of occurrences of each POS tag
- Number of hashtags
- # of elongated words (i.e. heey)
- Number of negated contexts (i.e. I don’t like Arsenal today!)
- Presence in the pre-defined clusters (provided by CMU POS tagger)
- Punctuation: contiguous sequences of exclamation marks, questions marks or both

Lexicon features:

- Lexicon features for all tokens in the tweet, each POS, hashtags and all-caps. For each token w, calculate the following:
  1. Total count of tokens with score(w,p) > 0
  2. Total score for all tokens
  3. Maximal score among all tokens
  4. Score of the last token where score(w,p) > 0

Experimentation results

Subjectivity Classifier

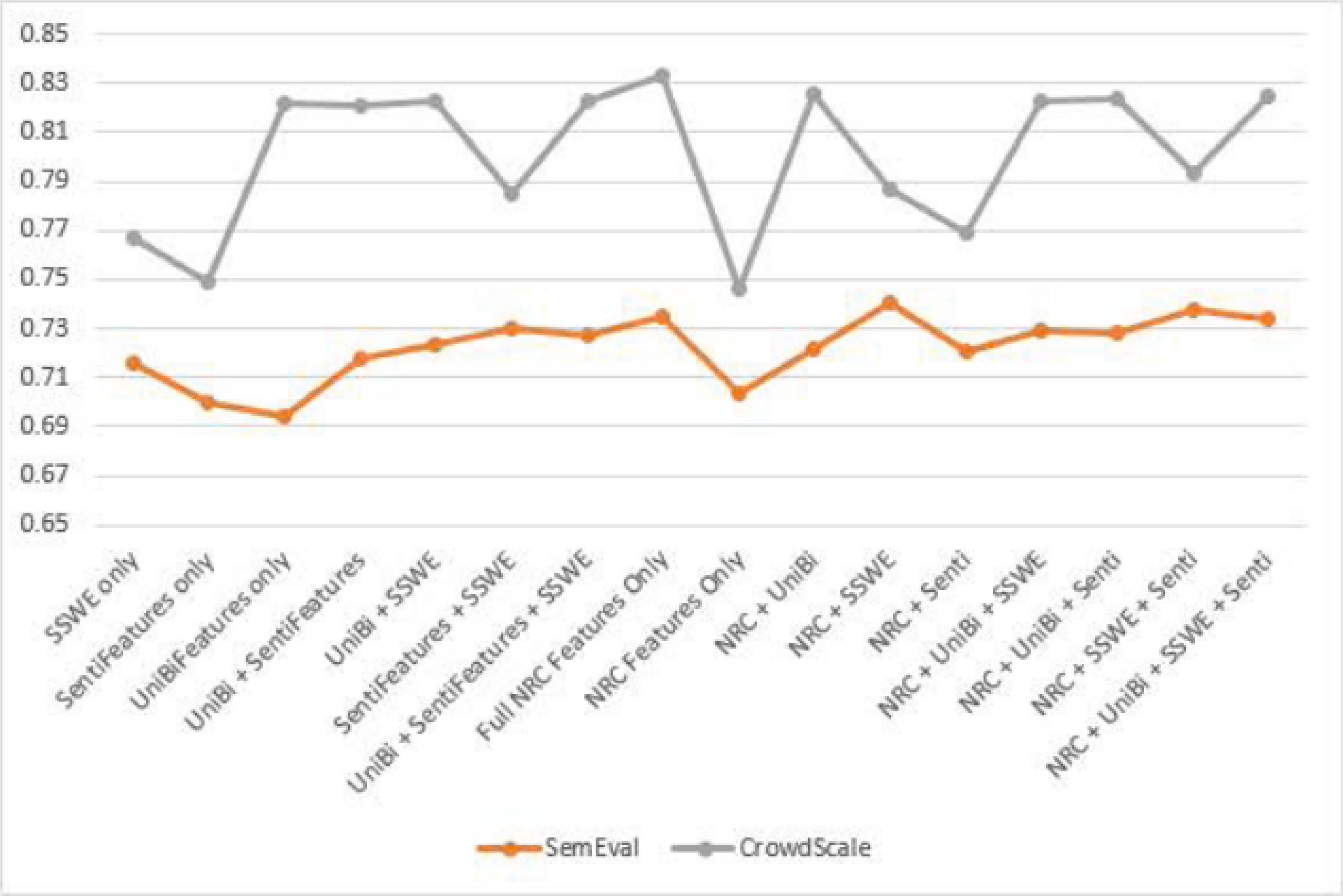


Fig. 2. Macro-F1 of subjectivity classifier for each feature set on each dataset

Polarity Classifier

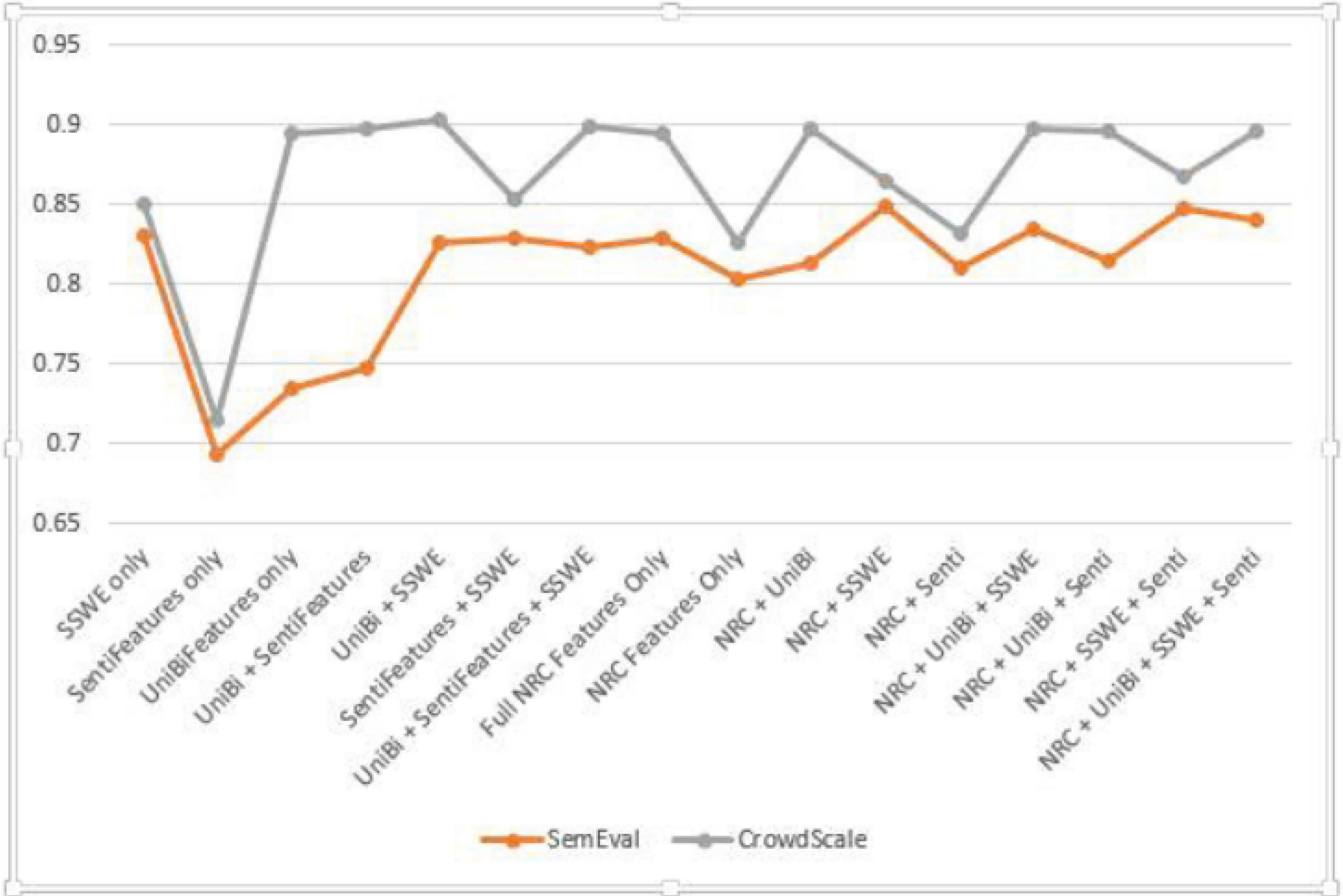
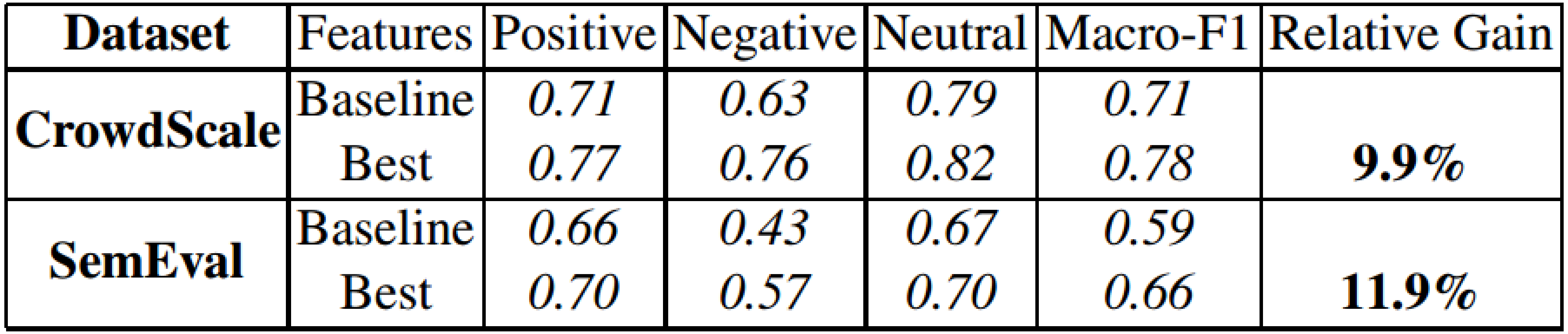


Fig. 3. Macro-F1 of polarity classifier for each feature set on each dataset

Ensemble



Dataset	Features	Positive	Negative	Neutral	Macro-F1	Relative Gain
CrowdScale	Baseline	*0.71*	*0.63*	*0.79*	*0.71*	
	Best	*0.77*	*0.76*	*0.82*	*0.78*	**9.9%**
SemEval	Baseline	*0.66*	*0.43*	*0.67*	*0.59*	
	Best	*0.70*	*0.57*	*0.70*	*0.66*	**11.9%**

Resources

Microsoft Cognitive Services (Text Analytics API):

- https://www.microsoft.com/cognitive-services/en-us/text-analytics-api

Publication:

- http://rd.springer.com/chapter/10.1007%2F978-3-319-18117-2_7

Microsoft