
[CMPUT 551] - Project Report

Sentiment Analysis in Figurative Language

(Using Twitter Data)

Anonymous Author(s)

Affiliation

Address

email

Abstract

The abstract paragraph should be indented 1/2 inch (3 picas) on both left and right-hand margins. Use 10 point type, with a vertical spacing of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

1.1 The Problem

Sentiment analysis is the area of natural language processing (NLP) concerned with computationally identifying the emotion expressed by an author of a piece of text (positive, negative, neutral). Determining the sentiment of a piece of text has important implications for opinion mining, parsing users reviews and recommendation systems. Humans communicate in complicated ways and often don't strictly use literal language. Figurative language, such as sarcasm, irony, and metaphors, are quite prevalent in standard human communication. Hence, in order to create better representations of human language, systems must take figurative language into account.

1.2 Related Work

1.3 Motivation

2 Problem Formulation

2.1 Twitter Data

Twitter is a web platform where users can post short messages with up to 140 characters to broadcast things they want the world to know. These messages are called *tweets* and the whole Twitter system became very famous in the last years. That's the reason why there's a high interest in performing sentiment analysis on twitter data. We are provided with a set of 8000 tweets where each tweet was annotated by 7 persons. Each one scored the given tweet on a range from -5 to 5 (where -5 indicates a very negativ, 0 a neutral and 5 a positive sentiment). The highest and lowest rate were ignored and the average of the remaining 5 scores is given for every 8000 tweets. This data is provided by *SemEval-2015 Task 11*.

2.2 Histogram

Figure 1 shows the scores distribution of the given data. Obviously most of the tweets have a negative score, so our mean score equals -1.992 and a standard deviation of 1.377.

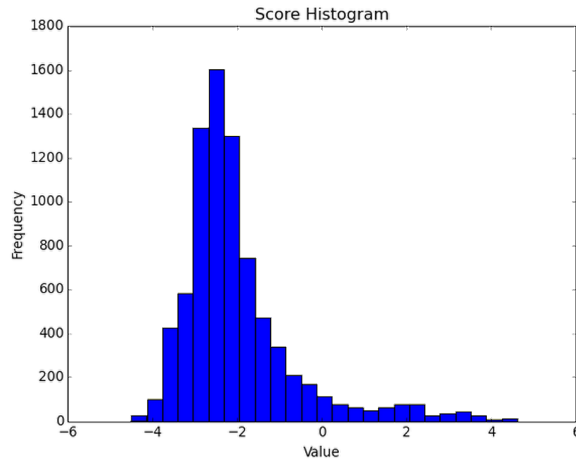


Figure 1: Histogram showing the score distribution for the given data

2.3 Learning Task

The learning task in this projects is to predict the score for a new tweet. In order to do this we have to ...

2.4 Pipeline

In order to have a running system where we can use our raw twitter data as an input and get an error value as output, we need to implement a couple of steps. This ends up as a chain of processes, which we defines in a system pipeline model. This model is shown in figure 2.

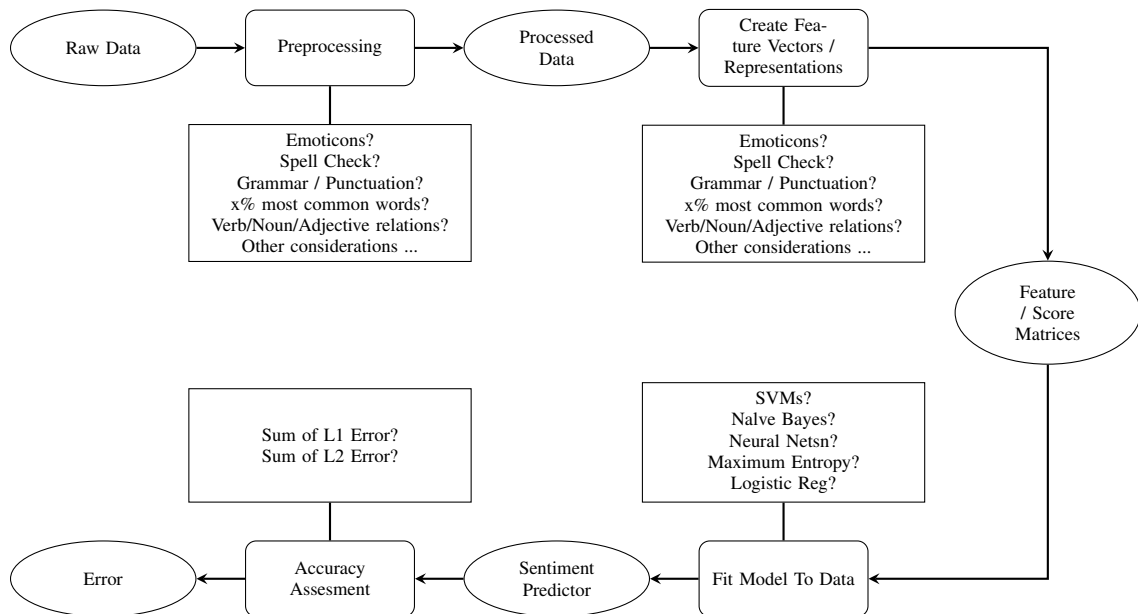


Figure 2: Pipeline Model

Table 1: Sample table title

NAME	DEFINITION	EXAMPLE		UNIQUE TOKENS COUNT
		BEFORE	AFTER	
Convert to lowercase	Converts all characters from a word to lowercase	PLEASE	please	20092
Remove apostrophe	Removes apostrophe and return the original words for most common cases where apostrophe is used as vowel substitution	you're	you are	22718
		I've	I have	
De-elongate words	Convert elongated words to normal version (works in combination with automated grammar corrector	pleeeeeeeese	please (please after grammar corrector)	22640
Remove stop words	There does not exist a single definition of stop words. For our task stop words are words which do not have any sentimental load. We used the stop word list from nltk corpus except words no and not under	under		22640
		before		
		very		
		to		

2.5 Evaluation

3 Preprocessing

3.1 pp1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3.2 pp2

4 Features

4.1 n-Grams

4.2 Part of Speec (POS) Tagging

One of the most fundamental parts of the linguistic pipeline is the part-of-speech (POS) tagging, a basic form of syntactic analysis which has countless applications in Natural Language Processing (NLP). We studied some of the best POS tagging tools & settled on using Tweet NLP, a twitter specific POS tagger. Weve used part-of-speech tagging as a feature for our task wherein every word/token in a given tweet is tagged based on its part-of-speech, some of which are twitter specific.

For our task, we selected 15 of the most common tags in linguistics (including the twitter specific tags). we made a count of the number of tagged tokens (with a confidence score ≥ 0.9) in a tweet & then used that as a feature.[rough work]

Tag	Description	Examples
O	pronoun	it you u me
N	common noun	central park
V	verb	amuse attach
A	adjective	good fav lil
R	adverb	2 (i.e. too)
D	determiner	the the it's its
P	pre or postposition	4 (i.e. for)
#	hashtags	#ualberta, #ml
@	at-mentions	@USA @Ualberta
~	discourse markers	RT for retweets
U	URL or email address	http://bit.ly/abc
E	emoticons	:-) :(:P :O
,	punctuations	!!! ... ?!?!
!	interjections	lol haha ftw yea
^	proper noun	John USA Ipad

Figure 3: caption msdbkjsghsg

Table 2: Sample table title

TAG	DESCRIPTION	EXAMPLES
O	pronoun	it you u me
N	common noun	central park
V	verb	amuse attach
A	adjective	good fav lil
R	adverb	2 (i.e. too)
D	determiner	the the it's its
P	pre or postposition	4 (i.e. for)
#	hashtags	#ualberta, #ml
@	at-mentions	@USA @Ualberta
~	discourse markers	RT for retweets
U	URL or email address	http://bit.ly/abc
E	emoticons	:-) :(:P :O
,	punctuations	!!! ... ?!?!
!	interjections	lol haha ftw yea
^	proper noun	John USA Ipad

4.3 Sentiment

We used *SentiWordNet* to add features specifically related to the sentiment of the tweets. This is dictionary, designed for opinion mining, where each of the over 100,000 words is assigned a positive and negative sentiment score. For a given tweet, we calculate a positive sum feature: $p_{sum} = \sum_i^n p_i$, where p_i is the positive sentiment score from SentiWordNet for the i th word in a tweet with n words. We similarly calculate a negative sum feature. If a word is not found in the dictionary, then it does not contribute to either of the two sentiment features. These two features are added to the bag of words feature vector as a real number. It is possible that different preprocessing steps could affect the sentiment score of a tweet by modifying said tweet (such as by stemming); future experimentation is required to determine the effects of different combinations of preprocessing on sentiment score.

4.4 Irony

To account for possible irony in the tweets we implement two features based on the work of Reyes et. al [year]. The features are created to detect the so-called *counter-factuality* and *temporal com-*

pression of a tweet. Reyes et. al. determined that ironic tweets were more likely to have a high level of these two measures.

Counter-factuality: The first measure, counter-factuality, is focused on “discursive terms that hint at opposition or contradiction in a text, such as about, nevertheless, nonetheless, and yet.” (Citation). The full list of counter-factual words includes 41 entrees; there are a total of 4187 occurrences of these words in the data set, and 3123 tweets contain at least one counter-factual word.

Temporal Compression: The second measure of tweet irony that we considered is temporal compression, which focuses on words related to an opposition in time, thus indicating an abrupt change in narrative. (citation) The list of temporal compression words contains 13 words such as suddenly, abruptly, and now. There are only 170 instances of a temporal compression word in the dataset, with only 103 tweets even containing a single instance.

Feature Creation: For each measure, we create a real-numbered feature based on the ratio of how many words in a given tweet possess the characteristic we are analyzing. That is, we have two ratio features $r_t = \frac{n_t}{n}$, where $t \in \{\text{counterFactuality}, \text{temporalCompression}\}$, n_t is the number of words in a given tweet that fit into category t , and n is the total number of words in the tweet.

270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

5 Experiments / Results

5.1 Experiment 1

5.2 Experiment 2

6 Conclusion

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to 'small' (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609-616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural Simulation System*. New York: TELOS/Springer-Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.