CISUC-KIS (supervised)

Leal, Joao, et al. "CISUC-KIS: Tackling Message Polarity Classification with a Large and Diverse Set of Features." SemEval@ COLING. 2014.

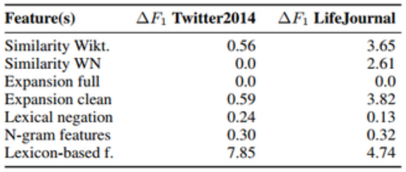
Técnica	Params	Treino modelo	Léxicos	Features	Resultados	Conclusões
SVM	• $C = 4$ • $\mu = 0$ e $\sigma = 1$ • $\gamma = 0.0003$ • For the parameter values, we did a grid search using the development set as a test. We also found that large values of C (25) and small γ values (0.0001) performed worse than smaller values of C (4) with a slightly higher γ (0.0003) when using the development set but not when using the training set under K-Folds. For the official evaluation, we opted for the best-performing results on the development set.	• Using tweets (no SMS for training) • Fscore of pos/neg classes • 10-fold cross validation • 70.41% on the training set (using 10-Folds) and 71.03%on the development set, after train on the training set. When tested against the training set,after train in the same set, we get a score of 84.32%, **which could indicate a case of under-Fitting**	• Bing Liu's Lexicon • AFINN list • NRCEmoticon Lexicon • MPQA Subjectivity Lexicon • Sentiment140 Lexicon • NRC Hashtag Lexicon • LabMT 1.0 • SentiWordNet 3.0 • Q-WordNet • Sentiment140 (API) • SentimentAnalyzer (API) • SentiStrength (API)	• Content of the tweets ○ Tokenized/PoS ○ CMU ARK and Stanford CoreNLP ○ emoticons ○ length ○ elongated words ○ hashtags ○ topic modelling ○ capital letters ○ punctuation ○ dashes and asterisks ○ PoS • Sentiment Lexicon (contribute highly) ○ # of pos and neg words ○ sum of all pos/neg polarity ○ the highest pos/neg polarity ○ the polarity value of the last word ○ sense disambiguation ○ full WordNet 3.0 ▪ get the previous scores for the sense ○ Lesk Algorithm adapted to wordnets ▪ using all the tweet's content words as the word context and the synset words, gloss words and words in related synsets as the synset's context	• 74.46% for the LiveJournal (2nd) • 65.9% for the SMS2013 (7th) • 67.56% for the Twitter2013 (7th) • 67.95% for the Twitter2014 (4th) • 55.49% for the Sarcasm (4th)	We followed a machine learning approach, with a diversified set of features, which tend to complemented each other. Some of the main takeaways are that the most important features are the lexicon related ones, including the n-grams and POS tags. Due to time constraints, we could not take strong conclusions on the impact of the word sense disambiguation related features alone.

Features Relevance

In order to get some insights on the most relevant group of features, **we did a series of experiments where each group of features were removed for the classification, then tested against the original score**. **We concluded that the lexicon related features contribute highly to the performance of our system**, including the set of features with **n-grams and POS**. Clusters, sport score, asterisks and elongated words provide little gains but, on the other hand, **emoticons and hashtags showed some importance and provided enough new information for the system to learn.** The API information is largely captured by some of our features and that makes it much less discriminating than what they would be on their own, but still worth using for the small gain. We also observed that it is **best to create a diversified set of lexicon features with extra very specific targeted features, such as punctuation, instead of focusing on using a specific lexicon alone.** Even though they usually overlap in information and may perform worse individually than a hand-refined single dictionary approach, they complement each other and that results in larger gains.

▪ UKPDIPF (supervised)

Flekova, Lucie, Oliver Ferschke, and Iryna Gurevych. "UKPDIPF: Lexical Semantic Approach to Sentiment Polarity Prediction in Twitter Data." SemEval@ COLING. 2014

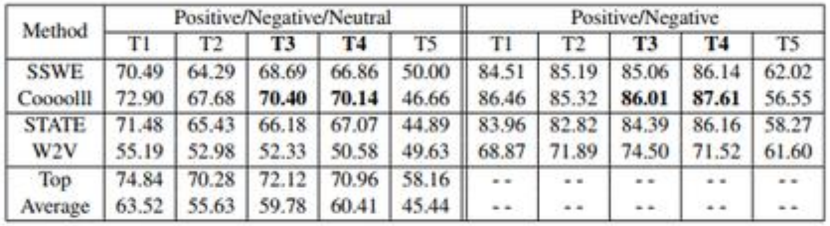
Técnica	Params	Treino modelo	Léxicos	Features	Resultados	Conclusões
SVM-SMO classifier with a gaussian kernel	• C = 1 • G = 0.01	• We trained our system on the provided **training data only, excluding the dev data**	• Bing Liu's Lexicon • Smiley polarity lexicon by Becker et al. (2013) • Swear word list • We further measure lexicon hits normalized per number of tweet tokens for the following lexicons • LIWC • NRC Emotion Lexicon • NRC Hashtag Emotion • Sentiment140 • Steinberger et al. (2012) • Combine each of the lexicons above with a list of weighted intensifying expressions as published by Brooke (2009) • List of invertors from Steinberger's Lexicon	• Negation • N-gram • Tweet expansion • Word similarity thesaurus computed on 80 million English tweets from 2012 using JoBim contextual semantics framework • Semantic similarity ○ ESA similarity measure • Elongated words • Ratio of sentences ending with exclamation • Ratio of questions • Number of pos and neg smileys and their proportion • We also separately measure the sentiment of smileys at the end of the tweet body, i.e. followed only by a hashtag, hyperlink or nothing • Ignore the first part of the tweet when the word **but** is found. This approach helps to reduce the misleading positive hits when a negative message is introduced positively (It'd be good, but) • We first segment the data with the Stanford Segmenter, apply the Stanford POS Tagger with a Twitter-trained model and subsequently apply the Stanford Lemmatizer, TreeTagger Chunker, Stanford Named Entity Recognizer and Stanford Parser to each tweet. After this linguistic preprocessing, the token segmentation of the Stanford tools is removed and overwritten by the ArkTweet Tagger, which is more suitable for recognizing hashtags and smileys as one particular token	• For subtask B, we scored 71.92 on LifeJournal and 63.77 on Twitter 2014, while the highest F-scores reported by related work were 74.84 and 70.96 • The majority of errors resulted from misclassifying neutral tweets as emotionally charged. This was partly caused by the usage of emoticons and expressions such as haha in a neutral context • Similar domain-specific polarity distinction could be applied to certain verbs, e.g., lose weight vs. lose a game • Another challenge for the system was the nonstandard language in twitter with a large number of spelling variants, which was only partly captured by the emotion lexicons tailored for this domain. A twitter-specific lemmatizer, which would group all variations of a misspelled word into one, could help to improve the performance • Double negations such as I don't think he couldn't... can easily misdirect the polarity score  Table 3: Performance increase where feature added to the full setup	We presented a sentiment classification system that can be used on both message level and expression level with only small changes in the framework configuration. We employed a contextual similarity thesaurus for the lexical expansion of the messages. The expansion was not efficient without an extensive stopword cleaning, overweighting more common words and introducing noise. Utilizing the semantic similarity of tweets to lexicons instead of a direct match improves the score only with certain lexicons, possibly dependent on the coverage. Negation by dependency parsing was more beneficial to the classifier than the negation by keyword span annotation. Naive combination of sentiment lexicons was not more helpful than using individual ones separately. Among the common source of errors were laughing signs used in neutral messages and swearing used in positive messages. Even within Twitter, same words can have different polarity in different domains (lose weight, lose game, game with nice violent fights...). Deeper semantic insights are necessary to distinguish between polar words in context.

Feature(s)	ΔF_1 Twitter2014	ΔF_1 LifeJournal
Similarity Wikt.	0.56	3.65
Similarity WN	0.0	2.61
Expansion full	0.0	0.0
Expansion clean	0.59	3.82
Lexical negation	0.24	0.13
N-gram features	0.30	0.32
Lexicon-based f.	7.85	4.74

▪ Coooolll (supervised)

Tang, Duyu, et al. "Coooolll: A Deep Learning System for Twitter Sentiment Classification." *SemEval@ COLING*. 2014.

Técnica	Treino modelo	Léxicos	Features	Resultados	Conclusões
SSWE	• We train the Twitter sentiment classifier on the **benchmark dataset in SemEval 2013**. The training and development sets were completely in full to task participants of SemEval 2013. However, we were unable to download all the training and development sets because some tweets were deleted or not available due to modi- fied authorization status. **We train sentiment classifiers with LibLinear** on the **training set and dev set**, and **tune parameter −c, −wi of SVM on the test set of SemEval 2013**. In both experiment settings, the **evaluation metric is the macro-F1 of positive and negative classes**.	• Bing Liu's Lexicon • SentiStrength • MPQA • NRC Emotion • NRC Hashtag • Sentiment140Lexicon	• All-Caps • Emoticons • Elongated Units • Sentiment Lexicon ○ Total sentiment score ○ Maximal sentiment score for each lexicon ○ Number of sentiment words ○ Score of last sentiment words • Negation • Punctuation • Cluster ○ The presence of words from each of the 1,000 clusters from the Twitter NLP tool • Ngrams. ○ The presence of word ngrams (1-4) and character ngrams (3-5)	• Among 45 systems of SemEval 2014 Task 9 subtask(b), Coooolll yields **Rank 2 on the Twitter2014** test set, along with the SemEval 2013 participants owning larger training data. • Details on table A	We develop a deep learning system (Coooolll) for message-level Twitter sentiment classification in this paper. The feature representation of Coooolll is composed of two parts, a state-of-the-art hand-crafted features and the sentiment-specific word embedding (SSWE) features. The SSWE is learned from 10M tweets collected by positive and negative emoticons, without any manual annotation. The effectiveness of Coooolll has been verified in both positive/negative/neutral and positive/negative classification of tweets.



Method	Positive/Negative/Neutral					Positive/Negative				
	T1	T2	**T3**	**T4**	T5	T1	T2	**T3**	**T4**	T5
SSWE	70.49	64.29	68.69	66.86	50.00	84.51	85.19	85.06	86.14	62.02
Coooolll	72.90	67.68	**70.40**	**70.14**	46.66	86.46	85.32	**86.01**	**87.61**	56.55
STATE	71.48	65.43	66.18	67.07	44.89	83.96	82.82	84.39	86.16	58.27
W2V	55.19	52.98	52.33	50.58	49.63	68.87	71.89	74.50	71.52	61.60
Top	74.84	70.28	72.12	70.96	58.16	--	--	--	--	--
Average	63.52	55.63	59.78	60.41	45.44	--	--	--	--	--

Table A

▪ KUNLPLab (supervised)

Assefa, Beakal Gizachew. "KUNLPLab: Sentiment Analysis on Twitter Data." *SemEval@ COLING*. 2014.

Técnica

L2 regularized logistic regression and used the LIBLINEAR implementation

Params

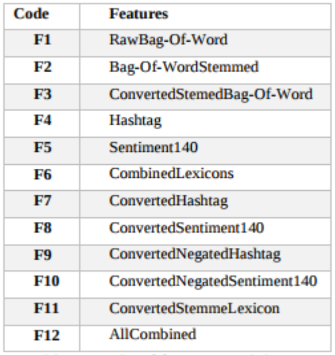
- C =1
- To estimate the hyper parameters, we applied a 10-fold cross validation on the training set
- Liblinear implementation of a L2 regularized logistic regression takes a single cost C parameter. The value of the cost C parameter decides the weight between the L1 regularization term and L2 regularization term. If the value of C is less than one, it means the more weight it given to the L1 regularization term. On the other hand C values more than one gives more weight to the L2 regularizing term. The cost parameter C=1 gives the best result on the cross-validation test. The same value is used to train our Model.

Léxicos

- NRCHashtag Sentiment Lexicon
- Sentiment140

Features

- We employed two major pre-processing in the datasets. Converting terms to their correct representation, and stemming
- There are two main categories of features used in the development of this system. Bag-of-Words and sentiment lexicon features.

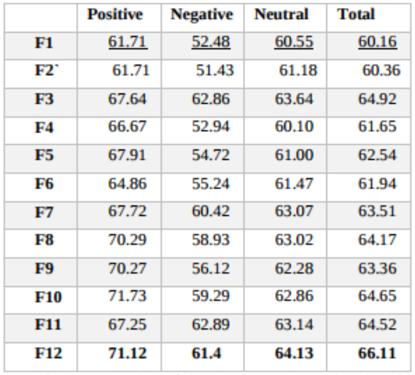


Code	Features
F1	RawBag-Of-Word
F2	Bag-Of-WordStemmed
F3	ConvertedStemedBag-Of-Word
F4	Hashtag
F5	Sentiment140
F6	CombinedLexicons
F7	ConvertedHashtag
F8	ConvertedSentiment140
F9	ConvertedNegatedHashtag
F10	ConvertedNegatedSentiment140
F11	ConvertedStemmeLexicon
F12	AllCombined

Table 2. Code of features and their names

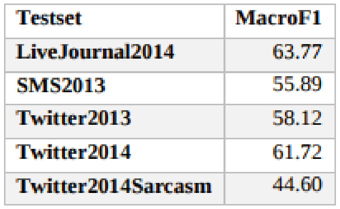
Resultados

- F1 is a baseline feature (raw Bag-Of-Word), with a total accuracy of 60.16. Simply converting the elongated terms to their normal form and applying stemming on the corpus increase the accuracy from 60.16 to 64.92 (4.76%).



	Positive	Negative	Neutral	Total
F1	61.71	52.48	60.55	60.16
F2`	61.71	51.43	61.18	60.36
F3	67.64	62.86	63.64	64.92
F4	66.67	52.94	60.10	61.65
F5	67.91	54.72	61.00	62.54
F6	64.86	55.24	61.47	61.94
F7	67.72	60.42	63.07	63.51
F8	70.29	58.93	63.02	64.17
F9	70.27	56.12	62.28	63.36
F10	71.73	59.29	62.86	64.65
F11	67.25	62.89	63.14	64.52
F12	**71.12**	**61.4**	**64.13**	**66.11**

Table 3. Results of the evaluation on the development set



Testset	MacroF1
LiveJournal2014	63.77
SMS2013	55.89
Twitter2013	58.12
Twitter2014	61.72
Twitter2014Sarcasm	44.60

Table 4. Evaluation result on test set

Conclusões

The performance of a classifier depends on feature representation, hyperparameter optimization and regularization. In this work, we mainly used bag-of-word features and sentiment lexicon features. We trained a L2 regularized logistic regression model. Two major features are used to represent the datasets; Bag-of-Word features and Lexical features. It has been shown that stemming the terms increases accuracy of the classifier in either case. The accuracy of the classifier on development set and training set is reported and has shown an increase of 6% in accuracy form the baseline with 95% confidence interval..The evaluation of our system on SemEval-2014 test data is also shown with an F measure of 44.60 to 63.77%