CSE 3021 - SOCIAL AND INFORMATION NETWORKS

PROJECT REPORT

SUBMITTED BY:

MANSI AGARWAL (16BCE0804)

SHUBHAM (16BCE0808)

ISHITA KUSHWAHA (16BCE0974)

SUBMITTED TO:

PROF. RAJKUMAR R.

<u>TOPIC</u>: SENTIMENT ANALYSIS USING EMOTICONS

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ABSTRACT

Emoticons, such as :) ;) :-) and :(, are frequently used online in social media, IM (e.g., Skype), blogs, forums, and other kinds of online social interactions. Because they are commonly used in online communications and they are often direct signals of sentiment, emoticons in text were widely used by NLP researchers in tasks such as sentiment analysis as features to machine learning algorithms or as entries of sentiment lexicons for rule-based approaches. Emoticons (e.g., :) and :( ) have been widely used in sentiment analysis and other NLP tasks as features to machine learning algorithms or as entries of sentiment lexicons. While emoticons are strong and common signals of sentiment expression on social media, the relationship between emoticons and sentiment polarity are not always clear. Thus, any algorithm that deals with sentiment polarity should take emoticons into account but extreme caution should be exercised in which emoticons to depend on. Different online communities and tools may elicit varied degrees of emoticon usage. Twitter, a microblogging site, is one of most popular social media. For researchers and businesses, having access to its huge amount of user-generated data is critical for understanding user behaviour and the sentiment expressed.

In *sentiment analysis*, polarity of sentiment (e.g., positive, negative or neutral) is of particular interest to researchers and business applications. However, the emotions expressed by the emoticons often cannot be captured by the three polarity categories. Many emoticons do not belong to exactly one of the categories. For example, :/ is often used to express an emotional state of annoyed and uneasy, which could be an indication of negative sentiment for some people but neutral for others.

INTRODUCTION

Sentiment Analysis refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral. Advanced, "beyond polarity" sentiment classification looks, for instance, at emotional states such as "angry", "sad", and "happy".

Precursors to sentimental analysis include the General Inquirer, which provided hints toward quantifying patterns in text and, separately, psychological research that examined a person's psychological state based on analysis of their verbal behaviour.

Subsequently, the method described in a patent by Volcani and Fogel, looked specifically at sentiment and identified individual words and phrases in text with respect to different emotional scales. Many other subsequent efforts were less sophisticated, using a mere polar view of sentiment, from positive to negative, such as work by Turney, and Pang who applied different methods for detecting the polarity of product reviews and movie reviews respectively. This work is at the document level. One can also classify a document's polarity on a multi-way scale, which was attempted by Pang and Snyder among others: Pang and Lee expanded the basic task of classifying a movie review as either positive or negative to predict star ratings on either a 3- or a 4-star scale, while Snyder performed an in-depth analysis of restaurant reviews, predicting ratings for various aspects of the given restaurant, such as the food and atmosphere (on a five-star scale).

First steps to bringing together various approaches—learning, lexical, knowledge-based, etc.—were taken in the 2004 AAAI Spring Symposium where linguists, computer scientists, and other interested researchers first aligned interests and proposed shared tasks.

LITERATURE REVIEW

Sentiment analysis is the multidisciplinary field of study that deals with analyzing people's sentiments, attitudes, emotions and opinions about different entities such as products, services, individuals, companies, organizations, events and topics and includes multiple fields such as natural language processing (NLP), computational linguistics, information retrieval, machine learning and artificial intelligence. It is set of computational and NLP based techniques which could be leveraged in order to extract subjective information in a given text unlike factual information, opinions and sentiments are subjective [1].

Despite the recent surge of interest in sentiment analysis since the term was coined by Nasukawa et al. [2] in 2003, the demand for information on sentiment and opinion during decision-making situations dates back to long before the widespread use of the World Wide Web. Opinions are central to almost all human activities as they could influence our behaviors specially when making a decision.

For example, many of us may have asked their friends to recommend a dishwasher or to explain who they might vote for during elections, or even requested reference letters from colleagues regarding job applications.

Now, opinions and experiences of numerous people that are neither our acquaintances nor professional critics are readily available thanks the Internet and the Web [3]. This is not limited to individuals only; businesses, organizations and companies are also eager to know consumers' opinions about their products and services. In the past, when a business needed consumer opinions, it conducted surveys and opinion polls. Nowadays, one is no longer limited to asking friends and family or conducting surveys for opinions about products; instead one can use volumes of user reviews and discussions in public forums on the Web [1]. Indeed, the Web has dramatically changed the way that people express their opinions about products, services, companies, individuals and social events. There are now many Internet forums, discussion groups, blogs and even micro-blogs that are well suited
for the users to freely post reviews about products and express their views on almost anything online. These users-generated contents and word-of-mouth behavior are sources of information with many immediate and practical applications.

The research on sentiment analysis appeared even earlier than 2003 [4, 5, 6, 7, 8, 9], while there were also some other earlier work [10, 11] on beliefs as frontiers or later work [12, 13, 14, 15, 16, 17, 18, 19] on interpretation of metaphors, sentiment adjectives, subjectivity, view points, affects and related areas [1, 3]. In contrast to the long history of linguistics and NLP, the area of analyzing people's opinions and sentiments has been virtually untrodden before the year 2000. However, since then, the literature witnessed literally hundreds of studies [20, 21, 22, 23, 24, 25, 26, 27, 28] due to several factors, including: (1)the rise of machine learning techniques in natural language processing and information retrieval; (2) access to datasets for training machine learning techniques because of the World Wide Web and specifically review-aggregation Websites, and; (3) realizing the huge applications in industry that the area started to offer [3]. This rapid growth of sentiment analysis and, more importantly its coincidence with the explosive popularity of the social media, have made the sentiment analysis the central point in the social media research [1]. Research on sentiment analysis has been investigated from different perspectives.

Perhaps the most popular perspective is to categorize these studies into three levels, document level, sentence level, and entity and aspect level [1] described as follows:

_ Document level: The aim here is to determine the overall sentiment of an entire document. For example given a product review, the task is to determine whether it expresses positive or negative opinions about the product. This level looks at the document as a single entity, thus it is not extensible to multiple documents.

_ Sentence level: This level of analysis is very close to subjectivity classification and the task at this level is limited to the sentences and their expressed opinions. Specifically, this level determines whether each sentence expresses a positive, negative or neutral opinion.

_ Entity and aspect level: Instead of solely analyzing language constructs (e.g. documents, paragraphs, sentences), this level (a.k.a feature level) provides finer grained analysis for each aspect(or feature) i.e., it directly looks at the opinions for different aspects itself. The aspect-level is more challenging than both document and sentence levels and consists of several sub-problems. It finds different

available sentiment. Sentiment analysis methods could be categorized into two groups, language processing based and application oriented methods. We describe the state-of-the-art approaches in each category and highlight their contributions. Then we conclude this section with a brief overview on visual analytics approaches in sentiment analysis.

PROPOSED METHODOLOGY

- **Data Collection**

Customers usually express their sentiments on public forums like the blogs, discussion boards, product reviews as well as on their private logs – Social network sites like Facebook and Twitter. Opinions and feelings are expressed in different way, with different vocabulary, context of writing, usage of short forms and slang, making the data huge and disorganized. Manual analysis of sentiment data is virtually impossible. Therefore, special programming languages like 'R' are used to process and analyze the data.

- **Text Preparation**

Text preparation refers to the filtering of the extracted data before analysis. It includes identifying and eliminating non-textual content and content that is irrelevant to the area of study from the data.

- **Sentiment Detection**

At this stage, each sentence of the review and opinion is examined for subjectivity. Sentences with subjective expressions are retained and that which conveys objective expressions are discarded. Sentiment analysis is done at different levels using common computational techniques like Unigrams, lemmas, negation and so on.

- **Sentiment Classification**

Sentiments can be broadly classified into two groups, positive and negative. At this stage of sentiment analysis methodology, each subjective sentence detected is classified into groups-positive, negative, good, bad, like, dislike.

- **Presentation of Output**

The main idea of sentiment analysis is to convert unstructured text into meaningful information. After the completion of analysis, the text results are displayed on graphs like pie chart, bar chart and line graphs.

Carrying out sentiment analysis is an important task for all the product and service providers today.

PROCEDURE OF SENTIMENT ANALYSIS

Step 1: Get some sentiment examples

As for every supervised learning problem, the algorithm needs to be trained from labeled examples in order to generalize to new data.

Step 2: Extract features from examples

Transform each example into a feature vector. The simplest way to do it is to have a vector where each dimension represents the frequency of a given word in the document.

Step 3: Train the parameters

This is where your model will learn from the data. There are multiple ways of using features to generate an output, but one of the simplest algorithms is logistic regression. Other well-known algorithms are Naive Bayes, SVM, Decision Trees and Neural Networks, but I'm going to use logistic regression as an example here.
In the simplest form, each feature will be associated with a weight. Let's say the word "love" has a weight equal to +4, "hate" is -10, "the" is 0 ... For a given example, the weights corresponding to the features will be summed, and it will be considered "positive" if the total is > 0, "negative" otherwise. Our model will then try to find the optimal set of weights to maximize the number of examples in our data that are predicted correctly.

If you have more than 2 output classes, for example if you want to classify between "positive", "neutral" and "negative", each feature will have as many weights as there are classes, and the class with the highest weighted feature sum wins.

Step 4: Test the model

After we have trained the parameters to fit the training data, we have to make sure our model generalizes to new data, because it's really easy to overfit. The general way of regularizing the model is to prevent parameters from having extreme values.

Going further

One of the big cons of our model is that it doesn't take into account the order of words in the document, since the feature vector is showing only the frequency of the words. For example, the sentences "good guy beats bad guy" and "bad guy beats good guy" have the same feature representations but have a different sentiment.
One way to overcome this problem is to generate more features, like the frequency of n-grams or the syntactic dependency between words. But my favorite model by Socher et al. takes a completely different approach. It uses the syntactic structure of the document to build up a vector representation by combining recursively the vector representations of words.

APPENDICES (RESULTS, CODE, SNAPSHOTS)

CODE:

```
import nltk
import numpy as np
from sklearn.utils import shuffle
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.linear_model import LogisticRegression
from bs4 import BeautifulSoup


ps = PorterStemmer()

positive_reviews = BeautifulSoup(open('positive.review').read())
positive_reviews = positive_reviews.findAll('review_text')

negative_reviews = BeautifulSoup(open('negative.review').read())
negative_reviews = negative_reviews.findAll('review_text')
diff = len(positive_reviews) - len(negative_reviews)
idxs = np.random.choice(len(negative_reviews), size=diff)
extra = [negative_reviews[i] for i in idxs]
negative_reviews += extra

def my_tokenizer(s):
    s = s.lower()
    tokens = nltk.tokenize.word_tokenize(s)
    tokens = [t for t in tokens if len(t) > 2]
    tokens = [ps.stem(t) for t in tokens]
    tokens = [t for t in tokens if t not in set(stopwords.words('english'))]
    return tokens

word_index_map = {}
current_index = 0
orig_reviews = []
```

```
positive_tokenized = []
negative_tokenized = []
for review in positive_reviews:
    orig_reviews.append(review.text)
    tokens = my_tokenizer(review.text)
    positive_tokenized.append(tokens)
    for token in tokens:
        if token not in word_index_map:
            word_index_map[token] = current_index
            current_index += 1

for review in negative_reviews:
    orig_reviews.append(review.text)
    tokens = my_tokenizer(review.text)
    negative_tokenized.append(tokens)
    for token in tokens:
        if token not in word_index_map:
            word_index_map[token] = current_index
            current_index += 1
def tokens_to_vector(tokens, label):
    x = np.zeros(len(word_index_map) + 1)
    for t in tokens:
        i = word_index_map[t]
        x[i] += 1
    x = x / x.sum()
    x[-1] = label
    return x

N = len(positive_tokenized) + len(negative_tokenized)

data = np.zeros((N, len(word_index_map) + 1))
i = 0
for tokens in positive_tokenized:
    xy = tokens_to_vector(tokens, 1)
    data[i,:] = xy
    i += 1

for tokens in negative_tokenized:
    xy = tokens_to_vector(tokens, 0)
    data[i,:] = xy
    i += 1

orig_reviews, data = shuffle(orig_reviews, data)
```

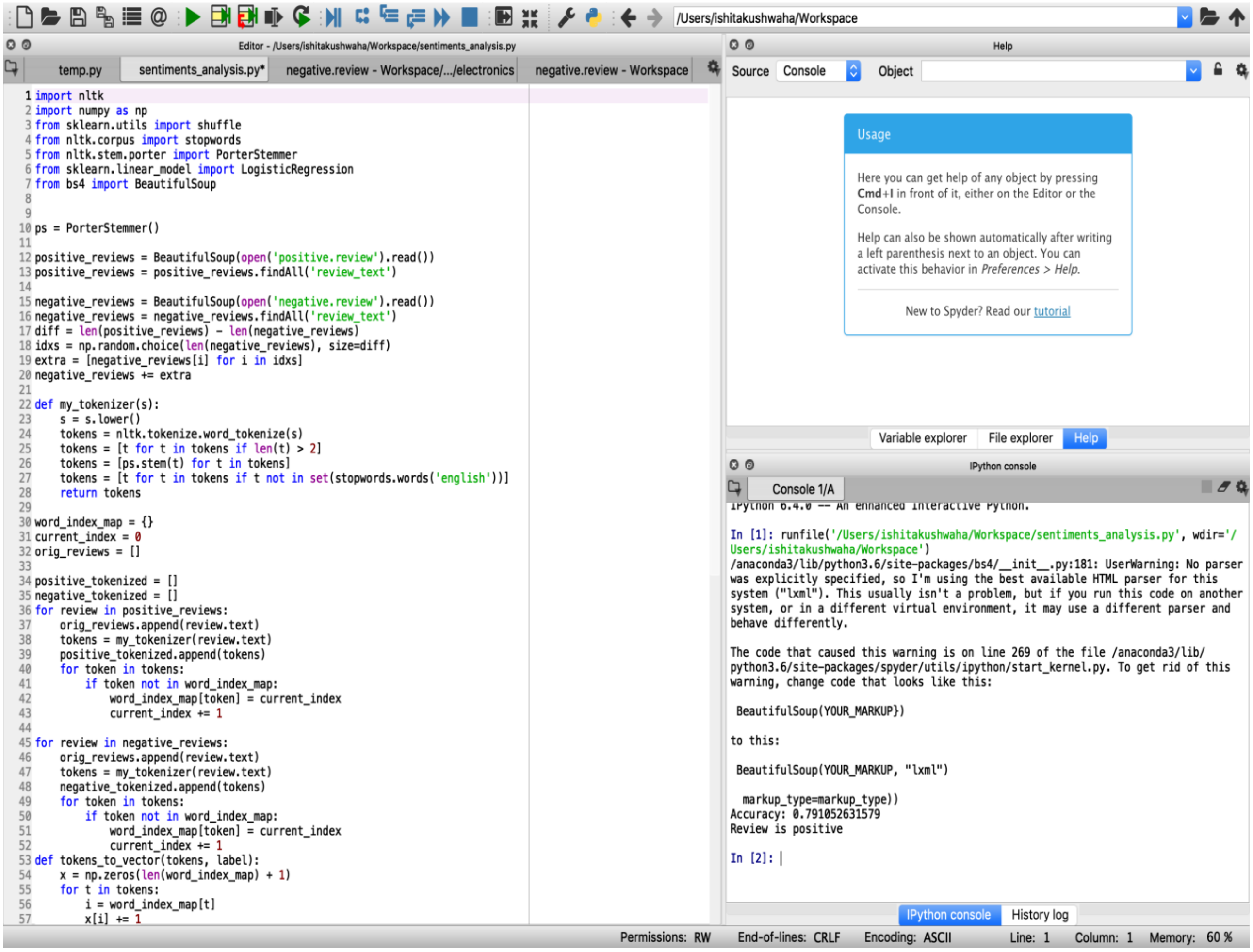
```
x = data[:,:-1]
y = data[:,-1]

Xtrain = x[:-100,]
Ytrain = y[:-100,]

model = LogisticRegression()
model.fit(Xtrain, Ytrain)
print("Accuracy:", model.score(Xtrain, Ytrain))

preds = model.predict(x)
p = model.predict_proba(x)[:,1]
prob = p.sum()/len(p)
if(prob>0.6):
   print("Review is neutral")
elif(0.4<prob<0.6):
   print("Review is positive")
else:
   print("Review is negative”)
```

OUTPUT:


/Users/ishitakushwaha/Workspace
Editor - /Users/ishitakushwaha/Workspace/sentiments_analysis.py
temp.py
sentiments_analysis.py*
negative.review - Workspace/.../electronics
negative.review - Workspace
import nltk
import numpy as np
from sklearn.utils import shuffle
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.linear_model import LogisticRegression
from bs4 import BeautifulSoup
ps = PorterStemmer()
positive_reviews = BeautifulSoup(open('positive.review').read())
positive_reviews = positive_reviews.findAll('review_text')
negative_reviews = BeautifulSoup(open('negative.review').read())
negative_reviews = negative_reviews.findAll('review_text')
diff = len(positive_reviews) - len(negative_reviews)
idxs = np.random.choice(len(negative_reviews), size=diff)
extra = [negative_reviews[i] for i in idxs]
negative_reviews += extra
def my_tokenizer(s):
s = s.lower()
tokens = nltk.tokenize.word_tokenize(s)
tokens = [t for t in tokens if len(t) > 2]
tokens = [ps.stem(t) for t in tokens]
tokens = [t for t in tokens if t not in set(stopwords.words('english'))]
return tokens
word_index_map = {}
current_index = 0
orig_reviews = []
positive_tokenized = []
negative_tokenized = []
for review in positive_reviews:
orig_reviews.append(review.text)
tokens = my_tokenizer(review.text)
positive_tokenized.append(tokens)
for token in tokens:
if token not in word_index_map:
word_index_map[token] = current_index
current_index += 1
for review in negative_reviews:
orig_reviews.append(review.text)
tokens = my_tokenizer(review.text)
negative_tokenized.append(tokens)
for token in tokens:
if token not in word_index_map:
word_index_map[token] = current_index
current_index += 1
def tokens_to_vector(tokens, label):
x = np.zeros(len(word_index_map) + 1)
for t in tokens:
i = word_index_map[t]
x[i] += 1
Help
Source
Console
Object
Usage
Here you can get help of any object by pressing Cmd+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer
File explorer
Help
IPython console
Console 1/A
In [1]: runfile('/Users/ishitakushwaha/Workspace/sentiments_analysis.py', wdir='/Users/ishitakushwaha/Workspace')
/anaconda3/lib/python3.6/site-packages/bs4/__init__.py:181: UserWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("lxml"). This usually isn't a problem, but if you run this code on another system, or in a different virtual environment, it may use a different parser and behave differently.
The code that caused this warning is on line 269 of the file /anaconda3/lib/python3.6/site-packages/spyder/utils/ipython/start_kernel.py. To get rid of this warning, change code that looks like this:
BeautifulSoup(YOUR_MARKUP})
to this:
BeautifulSoup(YOUR_MARKUP, "lxml")
markup_type=markup_type))
Accuracy: 0.791052631579
Review is positive
In [2]:
IPython console
History log
Permissions: RW
End-of-lines: CRLF
Encoding: ASCII
Line: 1
Column: 1
Memory: 60 %

IPython console

Console 1/A

```
IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/ishitakushwaha/Workspace/sentiments_analysis.py', wdir='/
Users/ishitakushwaha/Workspace')
/anaconda3/lib/python3.6/site-packages/bs4/__init__.py:181: UserWarning: No parser
was explicitly specified, so I'm using the best available HTML parser for this
system ("lxml"). This usually isn't a problem, but if you run this code on another
system, or in a different virtual environment, it may use a different parser and
behave differently.

The code that caused this warning is on line 269 of the file /anaconda3/lib/
python3.6/site-packages/spyder/utils/ipython/start_kernel.py. To get rid of this
warning, change code that looks like this:

 BeautifulSoup(YOUR_MARKUP})

to this:

 BeautifulSoup(YOUR_MARKUP, "lxml")

  markup_type=markup_type))
Accuracy: 0.791052631579
Review is positive

In [2]:
```

<u>LIST OF STOPWORDS</u>

!!
?!
??
!?
`
``
"
-lrb-
-rrb-
-lsb-
-rsb-
,
.
:
;
"
'
?
<
>
{
}
[
]
+
-
(
)
&
%
$
@
!
^
#
*
..

...
'll
's
'm
a
about
above
after
again
against
all
am
an
and
any
are
aren't
as
at
be
because
been
before
being
below
between
both
but
by
can
can't
cannot
could
couldn't
did
didn't
do
does
doesn't
doing
don't
down
during

each
few
for
from
further
had
hadn't
has
hasn't
have
haven't
having
he
he'd
he'll
he's
her
here
here's
hers
herself
him
himself
his
how
how's
i
i'd
i'll
i'm
i've
if
in
into
is
isn't
it
it's
its
itself
let's
me
more

most
mustn't
my
myself
no
nor
not
of
off
on
once
only
or
other
ought
our
ours
ourselves
out
over
own
same
shan't
she
she'd
she'll
she's
should
shouldn't
so
some
such
than
that
that's
the
their
theirs
them
themselves
then
there
there's

these
they
they'd
they'll
they're
they've
this
those
through
to
too
under
until
up
very
was
wasn't
we
we'd
we'll
we're
we've
were
weren't
what
what's
when
when's
where
where's
which
while
who
who's
whom
why
why's
with
won't
would
wouldn't
you
you'd

you'll
you're
you've
your
yours
yourself
yourselves
###
return
arent
cant
couldnt
didnt
doesnt
dont
hadnt
hasnt
havent
hes
heres
hows
im
isnt
its
lets
mustnt
shant
shes
shouldnt
thats
theres
theyll
theyre
theyve
wasnt
were
werent
whats
whens
wheres
whos
whys
wont

wouldnt
youd
youll
youre
youve

CONCLUSION

Sentiment analysis or opinion mining is a field of study that analyzes people's sentiments, attitudes, or emotions towards certain entities. It tackles a fundamental problem of sentiment analysis, sentiment polarity categorization. Online product reviews from Amazon.com are selected as data used for this study. A sentiment polarity categorization process has been proposed along with detailed descriptions of each step. Experiments for both sentence-level categorization and review-level categorization have been performed. We have found the positive and the negative reviews for the products and then trained them for the overall prediction. The overall score can be positive, negative or neutral depending on the number of reviews obtained.

REFERENCES

[1] Bing Liu. Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[2] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the 2nd International Conference on Knowledge Capture, K-CAP '03, pages 70–77, New York, NY, USA, 2003. ACM.

[3] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2):1–135, January 2008.

[4] Sanjiv Das and Mike Chen. Yahoo! for amazon: Extracting market sentiment from stock message boards. In Asia Pacific Finance Association Annual Conf. (APFA), 2001.

[5] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pages 341–349, 2002. Industry track.

[6] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan, sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pages 79–86. Association for
Computational Linguistics, 2002.

[7] Richard M. Tong. An operational system for detecting and tracking opinions in on-line discussion.
In Proceedings of the Workshop on Operational Text Classification (OTC), 2001.

[8] Peter D. Turney, semantic orientation applied to
Unsupervised classification of reviews. CoRR, cs.LG/0212032, 2002.

[9] Janyce Wiebe. Learning subjective adjectives from corpora. In Henry A. Kautz and Bruce W. Porter, editors, AAAI/IAAI, pages 735–740. AAAI Press / The MIT Press, 2000.

[10] Gary Carlson. Review of "subjective understanding, computer models of belief systems by jaime g. carbonell"; umi research press, ann arbor michigan copyright 1981.

[11] Yorick Wilks and Janusz Bien. Beliefs, points of view environments. In Proc. and multiple the International NATO Symposium on Artificial and Human Intelligence, pages 147–171, New York, NY, USA, 1984. Elsevier North-Holland, Inc.

[12] Marti Hearst. Direction-based text interpretation as an information access refinement. In Paul Jacobs, editor, Text-Based Intelligent Systems, pages 257–274. Lawrence Erlbaum Associates,
1992.
[13] Alison Huettner and Pero Subasic. Fuzzy typing for document management. In ACL 2000
Companion Volume: Tutorial Abstracts and Demonstration Notes, pages 26–27, 2000.
[14] Warren Sack. On the computation of point of view. In Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 2), AAAI'94, pages 1488–, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence.
[15] JanyceWiebe and Rebecca Bruce. Probabilistic classifiers for tracking point of view. InWorking Notes of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation, 1995.
[16] Janyce M. Wiebe. Identifying subjective characters in narrative. In Proceedings of the 13th Conference on Computational Linguistics - Volume 2, COLING '90, pages 401–406, Stroudsburg, PA, USA, 1990. Association for Computational Linguistics.
[17] Janyce M. Wiebe. Tracking point of view in narrative. Comput. Linguist.,20(2):233–287, June 1994.
[18] Janyce M. Wiebe, Rebecca F. Bruce, and Thomas P. O'Hara. Development and use of a gold standard data set for subjectivity classifications. In Proceedings of the 37th Annual Meeting
Of the Association for Computational Linguistics on Computational Linguistics, ACL'99, pages 246–253, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics.
[19] JanyceWiebe andWilliam J. Rapaport. A computational theory of perspective and reference in narrative. In Jerry R. Hobbs, editor, ACL, pages 131–138. ACL, 1988.
[20] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! ICWSM, 11:538–541, 2011.
[21] Guang Qiu, Bing Liu 0001, Jiajun Bu, and Chun Chen. Expanding domain sentiment lexicon through double propagation. In Craig Boutilier, editor, IJCAI, pages 1199–1204, 2009.
[22] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment in twitter events. Journal of the American Society for Information Science and Technology, 62(2):406–418, 2011.
[23] Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. User-level sentiment analysis incorporating social networks. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1397–1405. ACM,
2011.

[24] Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. Exploiting social relations for sentiment analysis in microblogging. In Proceedings of the sixth ACM international conference on Web search and data mining, pages 537–546. ACM, 2013.
[25] Jichang Zhao, Li Dong, Junjie Wu, and Ke Xu. Moodlens: an emoticon-based sentiment analysis system for chinese tweets. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1528–1531. ACM, 2012.
[26] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Unsupervised sentiment analysis with emotional signals. In Proceedings of the 22nd international conference on World Wide Web, pages 607–618. International World Wide Web Conferences Steering Committee, 2013.
[27] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment specific word embedding for twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers, pages 1555–1565, 2014.
[28] Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. In The Semantic Web–ISWC 2012, pages 508–524. Springer, 2012