Social Media Sentiment Analysis Using Twitter Dataset

ABSTRACT

With the advancement of web technology and its development, there is a huge amount of data present on the web for internet users and a lot of data is also generated. The Internet has become a platform for online learning, opinion exchange and opinion sharing. Social networking sites like Twitter and Facebook are rapidly gaining popularity as they allow people to share and express their views on topics, discuss with different communities or post messages across the world. There has been a lot of work in the sentiment analysis of Twitter data. This survey mainly focuses on sentiment analysis of Twitter data, which is useful for analyzing information in tweets where opinions are very unstructured, heterogeneous, and positive and negative. or neutral in some cases. In this paper, we will analyze sentiment of social media using the Twitter Dataset

Keywords: Twitter, Sentiment analysis (SA), Opinion mining, Lemmatization, Machine learning, Long Short Term Memory (LSTM).

Introduction

Nowadays, the internet era has changed the way people express their views and opinions. This is now mostly done through blog posts, online forums, product review sites, social media, etc. Today, millions of people use social networking sites like Facebook, Twitter, etc. to express their feelings, opinions and share views on their daily lives. Through online communities, we gain an interactive medium where consumers inform and influence others through forums. Social media generates large volumes of emotional data in the form of tweets, status updates, blog posts, comments, reviews, and more. Furthermore, social media offers opportunities for businesses by providing a platform to connect with their customers for promotional purposes. People mainly depend on user-generated online content to a large extent for decision making. For example. If someone wants to buy a product or use a service, they first check online reviews and discussions on social media before making a decision. The amount of user-generated content is too large for the average user to analyze. Hence the need to automate this, various sentiment analysis techniques are widely used.

Sentiment Analysis (SA) tells users whether a product is satisfying or not before purchasing. Marketers and businesses use this analytics data to understand their products or services so that they can be delivered according to user requirements. Documented information retrieval techniques mainly focus on the processing, study, or analysis of actual existing data. Facts have an objective component but there are other textual contents that show a subjective nature. These contents are mainly opinions, feelings, evaluations, attitudes, and emotions, which are the core of sentiment analysis (SA). It offers many exciting opportunities for the development of new applications, mainly due to the huge growth of information available on online sources such as blogs and social networks. For example, recommendations on items provided by a recommendation system can be predicted by taking into account considerations such as positive or negative opinions about such items using SA.

Methodology

Sentiment analysis can be defined as an automated process of extracting attitudes, opinions, opinions, and emotions from text, speech, tweets and database sources through language processing. natural language (NLP). Sentiment analysis involves classifying opinions in the text into categories such as "positive" or "negative" or "neutral". It is also known as subjective analysis, polling and evaluative extraction.

The words opinion, feel, opinion, and belief are used interchangeably, but there is a difference between them.

Opinion	A controversial conclusion (because different experts have different opinions)
View	Subjective opinion
Belief	Intentional acceptance and intellectual consent
Sentiment	Opinions represent one's feelings

Sentiment analysis is a term that encompasses many tasks such as emotion mining, sentiment classification, subjective classification, opinion synthesis, or opinion spam detection, among others. It aims to analyze feelings, attitudes, opinions, emotions, etc. of people related to items such as products, individuals, objects, organizations, and services. Mathematically, we can represent an opinion as a percentile (o, f, so, h, t), where: o = object; f = feature of object o; so = orientation or polarity of opinion about the f characteristic of object o; h = opinion holder; t = time the opinion was expressed.

Object	An entity can be a person, event, product, organization, or subject
Feature	An attribute (or part) of the object on which the evaluation is performed.
Opinion orientation or polarity	The orientation of an opinion about a characteristic f indicates whether the opinion is positive, negative, or neutral.
Opinion holder	An opinion holder is a person or organization or entity expressing an opinion.

In recent years, several researchers have done a lot of work in the field of "Twitter Sentiment Analysis". In its early days, it was used for binary classifications that assigned opinions or ratings to dipole classes, such as positive or negative.

[1] proposed a model to classify tweets as objective, positive, and negative. They created a Twitter database by collecting tweets using the Twitter API and automatically captioning those tweets with emojis. Using this dataset, they developed a sentiment classifier based on the polynomial Naive Bayes

method using features such as N-gram tags and POS. The training set they used was less effective because it only contained tweets with emojis.

[2] deployed two models, a Naive Bayes bigram model and a Maximum Entropy model to classify tweets. They found that the Naive Bayes classifier performed much better than the Maximum Entropy model.

[3] proposed an emotion analysis solution for Twitter data using remote monitoring, where their training data consisted of tweets with emojis acting as big tags. They build models using Naive Bayes, MaxEnt, and Support Vector Machines (SVM). Their feature spaces include unigram, bigram, and store. They conclude that SVM is superior to other models and unigram is more efficient in terms of features.

[4] designed a two-stage automated sentiment analysis method for ranking tweets. They classify tweets as objective or subjective, then in the second stage, subjective tweets are classified as positive or negative. The feature space used includes retweets, hashtags, links, punctuation, and exclamation points along with features like pre-word polarization and selling points.

[5] used Twitter streaming data provided by the Firehouse API, making every user post public in real time. They experimented with naive polynomial Bayes, stochastic gradient descent, and Hoeffding trees. They concluded that the SGD-based model, when used with an appropriate learning rate, is better than the rest used.

[6] developed a 3-dimensional model to classify sentiments into positive, negative, and neutral classes. They experimented with models such as: unigram model, feature-based model, and tree-core model. For the tree-core-based model, they represent tweets as a tree. The feature-based model uses 100 features, and the unigram model uses more than 10,000 features. They concluded that the features that combine the anterior polarities of words with their pronunciation tag (pos) are the most important and play a key role in the classification task. The tree-core-based model outperforms the other two.

[7] proposed an approach to use user-defined hashtags in tweets as sentiment classification using punctuation, single word, n-gram and patterns as feature types are then combined into a single feature vector for sentiment classification. They used the K-Nearest Neighbor strategy to label sentiment by constructing a feature vector for each example in the training and testing sets.

[8] used Twitter API to collect Twitter data. Their training data falls into three different categories (camera, film, mobile). The data is labeled as positive, negative and no opinion. Posts containing filtered comments. The Unigram Naive Bayes model was implemented, and the simplified Naive Bayes independence assumption was used. They also remove unnecessary features using mutual information and chi-square feature extraction methods. Finally, the direction of a tweet is predicted. i.e., positive, or negative.

[9] presented variations of the Naive Bayes classifier to detect the polarization of tweets in English. Two different variations of the Naive Bayes classifier were built, namely Baseline (trained to classify tweets as positive, negative, and neutral) and Binary (using polar and categorical vocabulary) are positive and negative. Neutral tweets are ignored). Features taken into account by taxonomists are lemma (noun, verb, adjective, and adverb), polar and multiple lexicons from different sources, and valence shifts.

[10] used the bag-of-words method for sentiment analysis, where the relationship between words is not considered at all and a document is represented as a set of simple words. To determine the sentiment for the entire document, the sentiment of each word was determined, and these values were unified with several aggregate functions.

[11] used the word Network vocabulary database to determine the emotional content of a word in different dimensions. They developed a distance measure on the WordNet and determined the semantic poles of adjectives.

[12] used a synthetic framework for sentiment classification achieved by combining different feature sets and classification techniques. In their work, they used two types of feature sets (partial speech information and word relationships) and three basic classifiers (Naive Bayes, Maximum Entropy and Support Vector Machines). They applied synthetic approaches such as fixed matching, weighted matching and meta-classifier matching to classify sentiments and achieve better accuracy.

[13] highlights effective challenges and techniques for extracting comments from Twitter tweets. Spam and extremely diverse languages make it difficult to get comments in Twitter.

A general model for sentiment analysis is as follows:

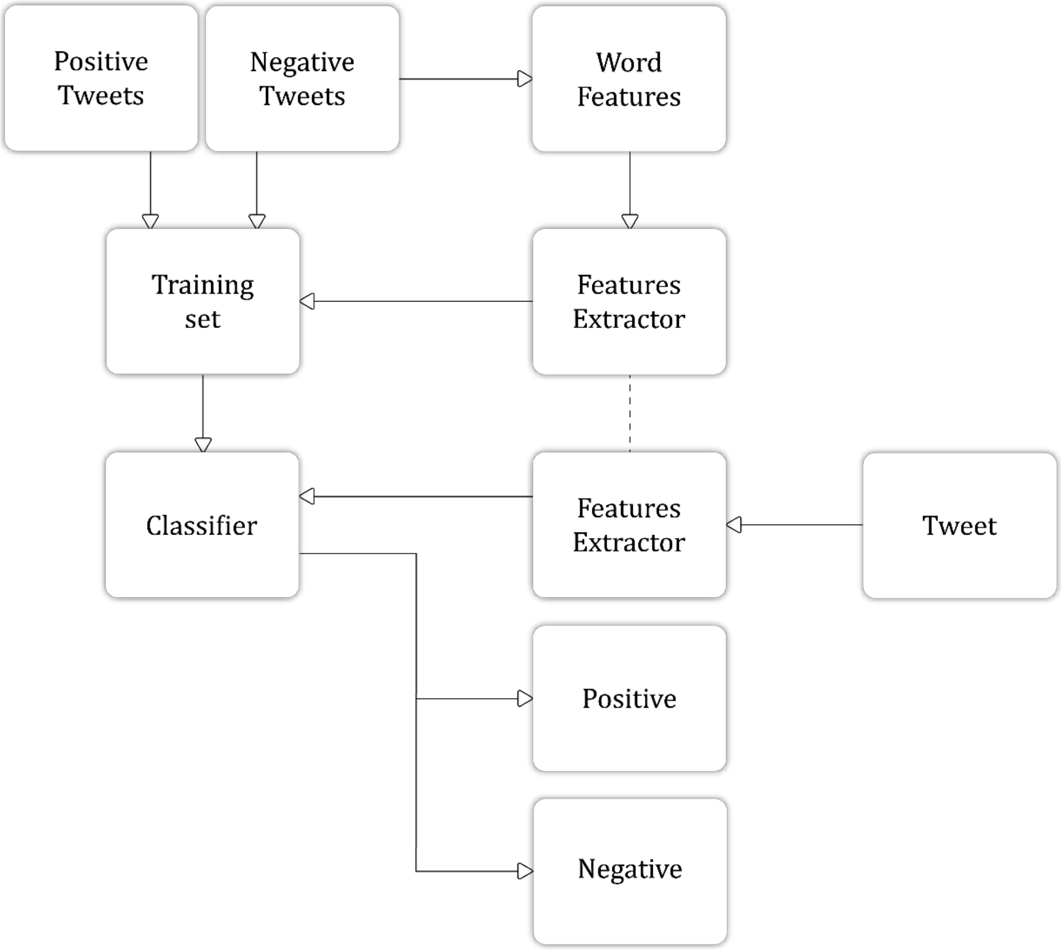

Figure – 1: Sentiment Analysis Architecture

Pre-processing of the dataset

A tweet containing multiple opinions on data expressed in different ways by different users. The Twitter dataset used in this survey work has been labeled into two classes viz. cathode and anode and thus sentiment analysis of the data becomes easy to observe the influence of different characteristics. Polar raw data is prone to inconsistencies and redundancies. Tweet preprocessing includes the following points:

- Remove all URLs (e.g., www.xyz.com), hash tags (e.g., #subject), target (@username)
- Correct spelling: Strings of repeating characters must be handled
- Replace all emojis with their emotions.
- Remove all punctuation, symbols, numbers
- Remove stop words
- Expand abbreviations (we can use a dictionary of acronyms)
- Delete non-English tweets

Table 1: Publicly Available Datasets for Twitter.

HASH	Tweets	http://demeter.inf.ed.ac.uk	31,861 Pos tweets 64,850 Neg tweets, 125,859 Neu tweets
EMOT	Tweets and Emoticons	http://twittersentiment.appspot.com	230,811 Pos& 150,570 Neg tweets
ISIEVE	Tweets	www.i-sieve.com	1,520 Pos tweets,200 Neg tweets, 2,295 Neu tweets
Columbia university dataset	Tweets	Email: apoorv@cs.columbia.edu	11,875 tweets
Patient dataset	Opinions	http://patientopinion.org.uk	2000 patient opinions
Sample	Tweets	http://goo.gl/UQvdx	667 tweets
Stanford dataset	Movie Reviews	http://ai.stanford.edu/~amaas/data/sentiment	50000 movie reviews
Stanford	Tweets	http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip	4 million tweets categorized as positive and negative
Spam dataset	Spam Reviews	http://myleott.com/op_spam	400 deceptive and 400 truthful reviews in positive and negative category.
Soe dataset	Sarcasm and nasty reviews	http://nlds.soe.ucsc.edu/iac	1,000 discussions, ~390,000 posts, and some ~ 73,000,000 words

Feature Extraction

The preprocessed data set has many special properties. In feature extraction method, we extract facets from the processed data set. This aspect is then used to calculate the negative and positive polarity in a sentence, which is useful in determining the opinions of individuals using patterns like unigram, bigram [15]

Machine learning techniques claim to represent the main features of a text or document to process. These key features are considered as feature vectors used for the classification task. Some examples of features that have been reported in the documentation are:

1. **Words And Their Frequencies**: Unigram, bigram and n-gram samples with their frequency counts are considered as features. There have been many studies on using word presence rather than frequency to better describe this characteristic. [16] showed better results when using presence instead of frequency.
2. **Parts Of Speech Tags**: Parts of speech such as adjectives, adverbs, and certain groups of verbs and nouns are good indications of subjectivity and sentimentality. We can generate syntactic dependency models by parsing or dependency trees.
3. **Opinion Words and Phrases**: Besides specific words, a number of emotional phrases and expressions can be used as traits. Example: costs someone an arm and a leg.
4. **Position Of Terms**: The position of a term in the text can affect the difference between the term and the overall feeling of the text.
5. **Negation**: Negation is an important but difficult characteristic to interpret. The presence of a negative often shifts the polarity of opinion. For example: I am not happy
6. **Syntax**: Syntactic patterns such as phrases are used by many researchers as traits to learn subjective patterns.

Supervised learning is an important technique for solving classification problems. Classifier training facilitates future predictions for unknown data.

1. **Naive Bayes Classification:** It is a probabilistic classifier and can learn the pattern by examining a set of classified documents [9]. It compares content with word lists to classify documents according to their correct category or class. Let d be tweet and c * be a class assigned to d, where

   $$C^* = \arg mac_c \, P_{NB}(c|d); \;\; P_{NB}(c|d) = \frac{(P(c))\sum_{i=1}^{m} p(f|c)^{n_{i(d)}}}{P(d)}$$

   From the above equation, "f" is a "feature", the feature number (fi) is denoted n_i(d) and present in d represents a tweet. Here m denotes zero. of features. The parameters P(c) and P(f|c) were calculated using the maximum likelihood estimation and smoothing was used for unseen objects. For training and classification using the Naïve Bayes Machine learning technique, we can use the Python NLTK library.
2. **Maximum Entropy Classification**: In the Maximum Entropy Classifier, no assumptions are made about the relationships between the extracted objects from the dataset. This classifier always tries to maximize the entropy of the system by estimating the conditional distribution of the class label. The maximum entropy even handles feature overlap and is like the logistic regression which finds the distribution over classes. The conditional distribution is defined as MaxEnt makes no independent assumptions for its features, unlike Naive Bayes.

   $$P_{ME}(c|d,\lambda) = \frac{exp[\sum_i \lambda_i f_i(c,d)]}{\sum_c \exp\,[\sum_i \lambda_i j_i(c,d)]}$$

   Where c is the class, d is the tweet and λ_i is the weight vector. The weight vectors determine the importance of an object in the classification.

3. **Support Vector Machine:** Vector machines support data analysis, decision bounds determination, and kernel use for calculations performed in the input space [15]. The input data are two sets of vectors of size m each. Then all the data represented by a vector is classified into a class. Then we find a margin of between the two layers, which is far from any document. The distance determines the classifier's margin, maximizing the margin reduces undecided decisions. SVM also supports classification and regression, which is very useful for statistical learning theory, and it also helps to accurately recognize the factors that need to be considered in order to understand it successfully.

Approaches for sentiment analysis

The machine learning-based method uses the classification technique to classify the text into classes. There are mainly two types of machine learning techniques:

1. Unsupervised learning: It does not include a category and they do not provide exact targets with and are therefore based on grouping.

2. Supervised learning: It is based on a labeled dataset and hence the labels are fed to the model in the process. These labeled data sets are trained to yield meaningful outputs when encountered during decision making.

   The success of these two learning methods mainly depends on selecting and extracting the specific set of features used for affective detection. The machine learning method applied to sentiment analysis mainly belongs to supervised classification. In machine learning technique, two datasets are required - Training Set, Test Set.

   Several machine learning techniques have been built by to classify tweets into classes. Machine learning techniques such as Naive Bayes (NB), Maximum Entropy (ME), and Support Vector Machines (SVM) have achieved great success in sentiment analysis.

   Machine learning starts with collecting a set of training data. Then we train a classifier on the training data. When the supervised classification technique was chosen, an important decision for the was to choose a feature. They can tell us how the documents are presented. The most common features used in the sentiment classification are:

   - Term presence and their frequency
   - Part of speech information
   - Negations
   - Opinion words and phrases

For supervised techniques, Support Vector Machines (SVM), Naive Bayes, Maximum Entropy are some of the most commonly used techniques.

While semi-supervised and unsupervised techniques are introduced when it is not possible to have an initial set of labeled documents/assessments to classify the remaining items.

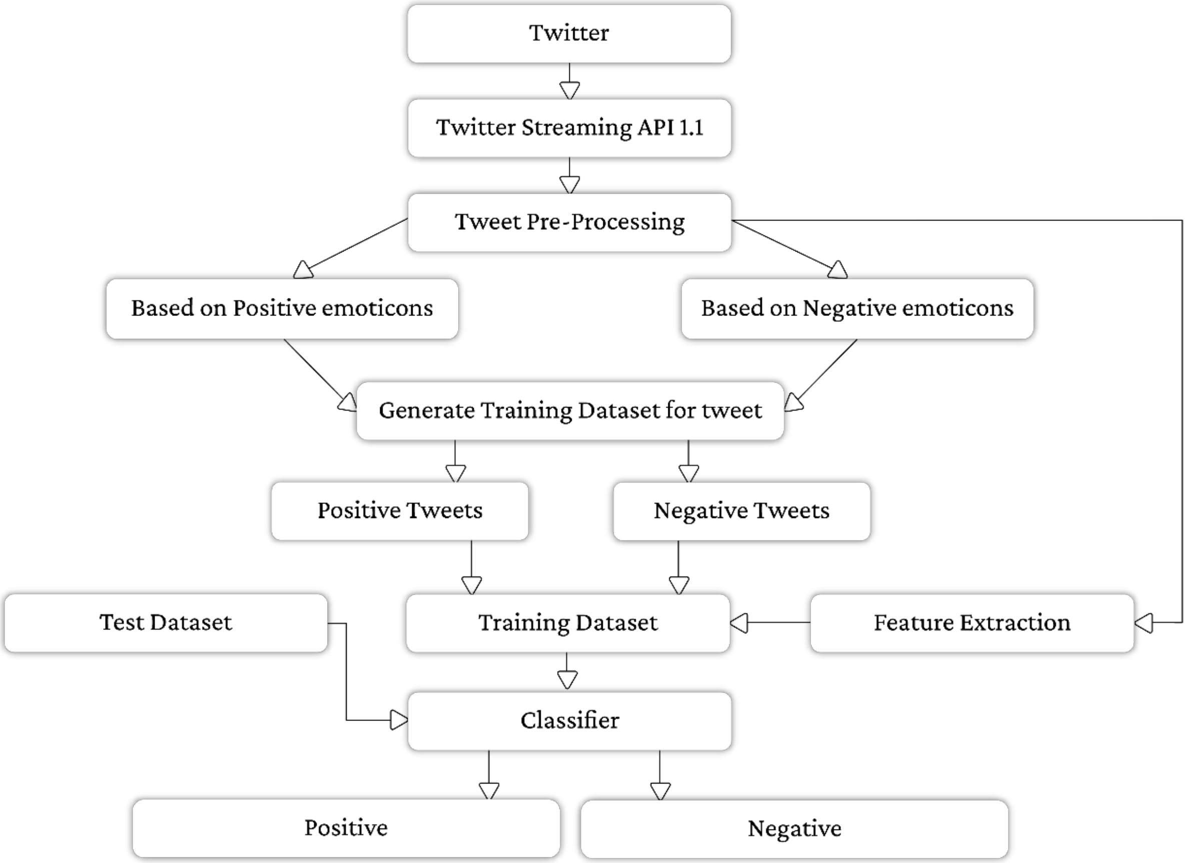

Figure – 2: Sentiment Classification Based On Emoticons

Lexicon-Based Approaches

The lexical-based method [15] uses a sentiment dictionary with opinion words and matches them with data to determine polarity. They assign sentiment scores to opinion words that describe how positive, negative, and objective the words in the dictionary are. Vocabulary-based approaches rely primarily on affective vocabulary, that is, a collection of known, pre-compiled emotional terms, phrases, and even idioms developed for traditional communication genres, such as Vocabulary Seeking Opinion.

There are two sub-categories for this approach:

1. Dictionary-based: It is based on the use of commonly collected and annotated terms (seeds). This set is enriched by looking up synonyms and antonyms from the dictionary. An example of this dictionary is the WordNet, which was used to develop a synonym called SentiWordNet.
   Cons: Can't handle domain and context specific orientations.

2. Corpus-Based: The corpus approach aims to provide dictionaries related to a particular domain. These dictionaries are created from a set of input opinion terms developed through the search for related words through the use of statistical or semantic techniques.

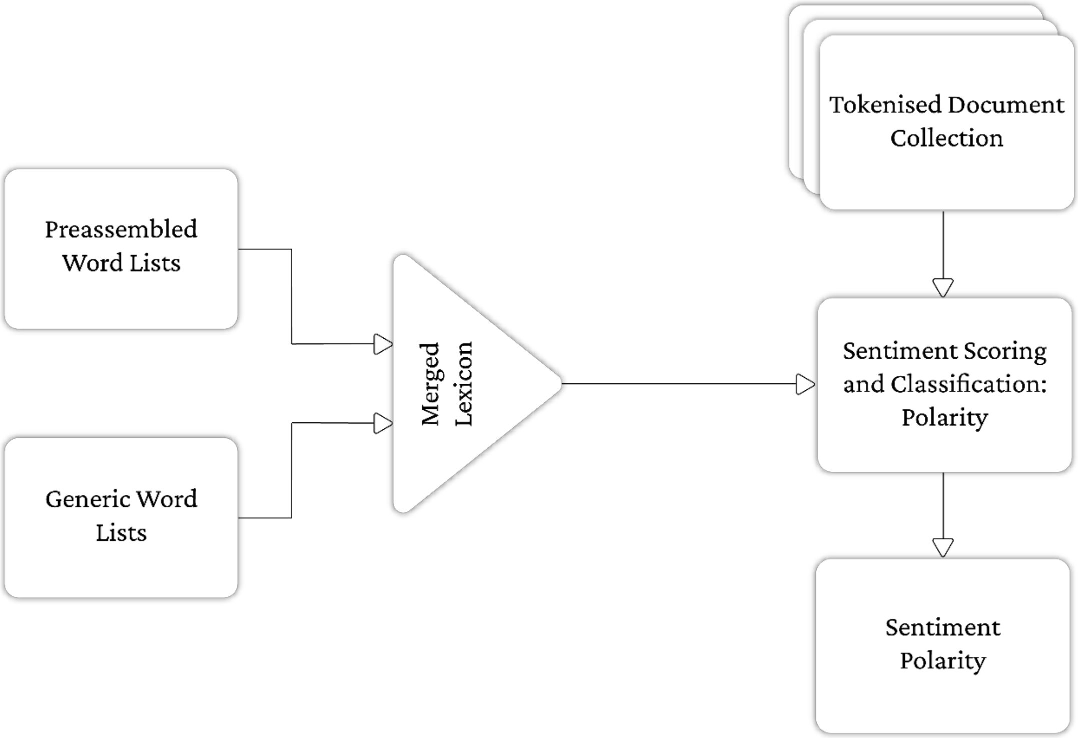
Preassembled Word Lists
Generic Word Lists
Merged Lexicon
Tokenised Document Collection
Sentiment Scoring and Classification: Polarity
Sentiment Polarity

Figure – 3: Lexicon-Based Model

- Statistical Method: Latent Semantic Analysis (LSA).
- Semantic-based methods such as the use of synonyms and antonyms or word-synonym relationships such as WordNet could also represent an interesting solution.

Following performance measures such as accuracy and recall, we provide a comparative study of existing opinion mining techniques, including machine learning, word-based approaches, and more. vocabulary, multi-domain and multilingual approaches, etc., as shown in Table 2.

Table 2: Publicly Available Datasets for Twitter.

	Method	Data Set	Accuracy	Author
Machine Learning	SVM	Movie reviews	86.40%	Pang, Lee [16]
	CoTraining SVM	Twitter	82.52%	Liu [14]
	Deep learning	Stanford Sentiment Treebank	80.70%	Richard [15]
Lexical based	Corpus	Product reviews	74.00%	Turkey
	Dictionary	Amazon's Mechanical Turk	---	Taboada [17]
Cross-lingual	Ensemble	Amazon	81.00%	Wan,X. [18]
	Co-Train	Amazon, ITI68	81.30%	Wan,X. [18]
	EWGA	IMDb movie review	>90%	Abbasi,A.
	CLMM	MPQA, NTCIR, ISI	83.02%	Mengi
Cross-domain	Active Learning	Book, DVD, Electronics, Kitchen	80% (avg)	Li, S
	Thesaurus			Bollegala[22]
	SFA			Pan S J [14]

Sentiment Analysis Tasks

Sentiment analysis is a challenging interdisciplinary task that includes natural language processing, web crawling, and machine learning. This is a complex task and can be divided into the following tasks, namely:

- Subjective Classification
- Sentiment Classification
- Complimentary Tasks
    - Object Holder Extraction
    - Object/Feature Extraction

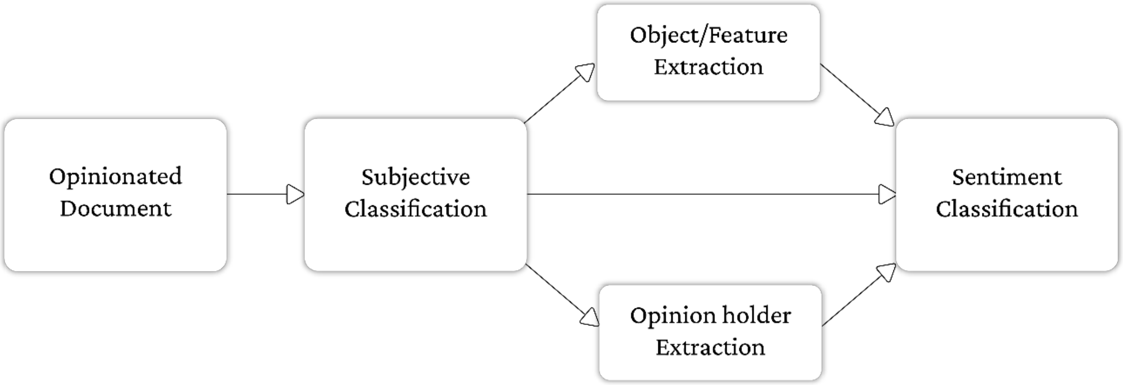
Opinionated Document
Subjective Classification
Object/Feature Extraction
Opinion holder Extraction
Sentiment Classification

Figure – 4: Sentiment Analysis Tasks

Levels of Sentiment Analysis

The tasks described in the previous section can be performed at several levels of detail.

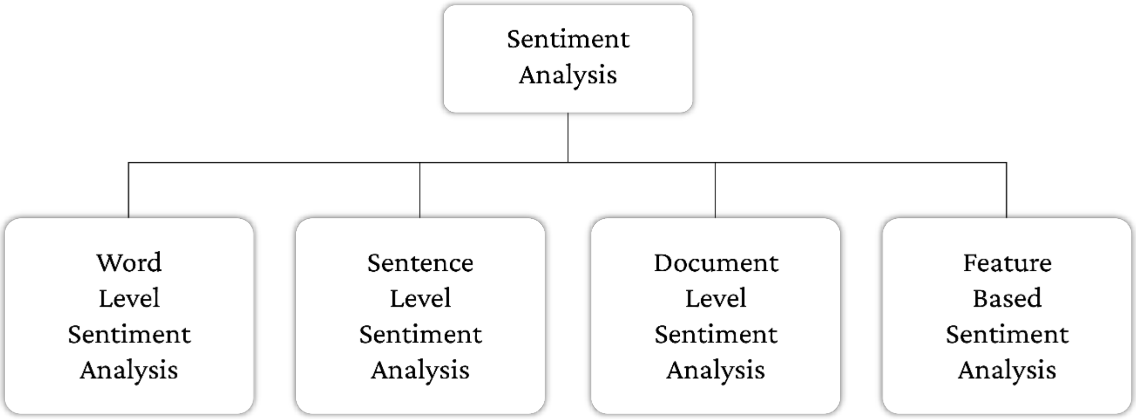
Sentiment Analysis
Word Level Sentiment Analysis
Sentence Level Sentiment Analysis
Document Level Sentiment Analysis
Feature Based Sentiment Analysis

Figure – 5: Levels of Sentiment Analysis

Document level refers to marking individual documents with their sentiment. At the document level, the entire document is classified as either positive or negative at the grade level. General Approach is to find the poles of each phrase or word in and combine them to find the pole of document & other approach is Complex linguistic phenomena such as resolution of co-references, pragmatics, etc. The various tasks involved in this are: Task: Emotions Categorize entire document Categories: Positive, negative, and neutral Assumption: Each document focuses on a single audience (this is not true in discussion posts, blogs, etc.) and contains comments by a single opinion holder

Sentence-level analysis deals with tagging individual sentences with their respective emotional poles. Sentence level classification classifies sentences into levels of positive, negative, or neutral. General Approach is to find the sentimental orientation of each word in the phrase/sentence, then combine them to determine the sentiment of the phrase or the whole phrase & other approach is examining the discursive structure of text, the various tasks involved in this are Task 1: Determine Subjective/Objective Sentence Type: Objective and Subjective Task 2: Categorize Sentiment Sentence Type: Positive and Negative Hypothesis: A sentence contains only one opinion may not always be true.

Aspect Level or Functional Level This involves labeling each word with their emotion and identifying the entity to which the sentiment is directed. Sentiment classification problems at the aspect or function level with identifying and extracting product features from source data. Techniques such as dependency parsing, and discourse construction are used for this purpose. Various Tasks involved in this are Task 1: Identify and extract the characteristics of an object that have been commented on by an opinion holder (e.g., reviewer) Task 2: Determine if the opinion about the object is negative, positive, or neutral Task 3: Look up synonyms for the object.

Most recent works have used pre-sentence-word polarization to classify sentiments at the sentence level and the document level. Classification of feelings from mainly using adjectives as characteristics but adverbs, the two automatic word-level sentiment annotation methods are Dictionary-based approach & The corpus-based approach.

Data Preparation

Since this dataset is quite large, during the exploration we start by generating sub-datasets during training, to speed up the testing steps. We also re-index the dataframe, to make it easier to add columns later in the process.

Here we explore the data, check if it is balanced and check for patterns in missing rows. This can usually be done automatically with pandas config. Sections in this title include:

- Basic visualization - We can display some basic statistics about the data using pandas and also display a few entries from the dataset, to see the sample points we will be working with.
- Automated data exploration with pandas configuration - Pandas-profiling is a library used to automate data discovery. This gives us a good overview of the dataset, which we can use to inform our further work.
- Checking balances in output categories - We want to check the balance of the output column (Sentiment), so we don't train a model that always predicts the output. This model can be highly accurate, but we won't learn anything about trends in the data other than numbers from a general point of view. You can also think about balancing test data, but keep in mind that real world data can hardly be well balanced and experimental data is like real world data

  Sentiment ratios change d' by about 0.15 to 0.3, which is usually a good balance, so we are unlikely to see a scenario where only one class is predicted. However, we will be watching to see if our training accuracy gets around 0.3, which could be a sign of this problem.

We can create an indexer to convert the perception from label to index and vice versa. This is very useful for understanding our predictions later. Then we convert the "Opinion" column in the training data into labels, we will learn how to predict. After that we work with the text in the "Tweet_Content" column, extract all the information we can and convert it into a format usable for the neural network that we will train later. This involves removing words with little meaning (keyword removal) and grouping words with the same meaning regardless of details such as time (lemmatization). Next, we use encryption to encode the presence of words in the matrix, similar to hot encoding. This is called the "bag of words" method. Sections in this title include:

- Stop Word Removal and Lemmatization with NLTK - Here, we first split each string into its individual words, before checking if these strings - Contains text, yes in keyword list If no text in word, meaning is, has only digits or punctuation (or other characters), or the word is a stop word (words like "with", "a", "the"), the word is removed from the string. We also perform lemma in this step, where we convert words back to their original form, so tenses and other details can be omitted in our final model (a negative past tense statement is always negative).
- Tokenization - We generate tokens for the most common words in the dataset, so that we can represent the presence of words in our generated corpus (n most common words) with a list of integers.
- Adding the Tokenized Strings to the DataFrame

The data is now almost ready for a trained model, but some final preparation will need to take place. For example, we need to remove columns that we do not intend to use, such as the "Tweet_Content" column, from which useful information has been extracted. We also split the data into a training and test set so that we can evaluate the performance of our model without touching the retained data. We do this because if we constantly examine this retained data, it loses its usefulness as invisible “real world” data. Sections in this heading include Clear unused data, Test stream separation.

- Dropping Unused Data - Here we remove unnecessary columns from the DataFrame. They have no use value (Tweet ID) or useful information has been extracted (Tweet Content). We're also removing "y" or the dependent variable here, so we don't accidentally train on it.
- Test-Train Split - Here, we use SciKit-Learn's built-in functionality to split our data into a test set and a train set, with the appropriate labels. We use a constant random state to make this repeatable.

Model Construction and Training

Finally, it's time to build our model. In this case, we use a neural network built with Keras. Then we train it with our data in the training dataset and validate using the test dataset.

- Model Construction - Here we define the neural network that we will train to predict the output. This model is built with the following classes:
  - Embedding - The embedding layer is a very important part of this process, as it allows us to learn the meaning of words, based on the context of other words around them. In effect, this class places each word in a vector space, in which we will learn where and what words can be used around it.
  - Bidirectional - These layers are improvements to cyclic neural networks (RNNs), where data flows in a certain order so that we can learn from the order of the text. In two-dimensional layers, this happens in both directions, so we can try to find out the meaning of words in context both forward and backward. We use LSTMs (long-term memory nodes) in this network, so our network can "remember" the previous context. LSTMs perform better than RNNs for applications such as word processing because they solve the problem of gradients disappearing by allowing gradients to pass through unchanged.
  - Dense - The last layer of the network is a dense layer. It just means that all the nodes in this layer are connected to all the nodes in the previous layer, so the input of any LSTM unit can

be included. Here we use softmax exit function and 4 nodes to generate the probability for each potential exit (each possible sentiment).

- Training - We then adapted this model to our data, using backpropagation, over 15 epochs. We can see an increase in model accuracy over different epochs, both on training and test datasets. Now that we have trained the model, we can see its accuracy with a confusion matrix. This allows us to see predictions for Tweets with different real values. From that, we can see that we predict some classes better than others. Now that we are satisfied with our model, we can train using the full dataset and predict the retained test data. This involves performing all of our transformations on both this training dataset and the retained test data. Fortunately, we can reuse the above code to achieve this, so a bit more explanation is needed.

There are a few potential reasons for this. First, we don't really know how the Tweets in the workout data are classified. Maybe a bag-of-words classifier was used, which means we can just reproduce it. Next steps here might be to look at the training and test data to see if that's the case and see if the models perform better on the changed data.

In addition, we need to know the limitations of our model. Two-way LSTM models often require longer pieces of text to learn the context of words, which is not possible in a Tweet. Where bidirectional LSTM models really shine is in longer pieces of text and perform much better than word-for-word classifiers for tasks like evaluation classification. As a result, we generally don't expect great performance from this type of model on Tweets, where word bags can often work very well, due to this lack of context.

Conclusion

It turns out that we can classify Tweets pretty well with a bag of "standard" word classifications. Here we can assume that this is largely due to a lack of context in Tweets, meaning that more complex sentence structures are difficult to display.

If we have a connection to any of the entities mentioned in the classifier, we may want to configure it to automatically detect people's opinions about our products or services. This can provide very valuable insights into what people are thinking without having to manually extract data anymore.

Testing more advanced methods, such as a pre-trained Bidirectional Encoder Representation from a Transformer (BERT) model may yield better results than what we get here

Reference

[1] A.Pak and P. Paroubek. „Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp.1320-1326

[2] R. Parikh and M. Movassate, "Sentiment Analysis of User- Generated Twitter Updates using Various Classi_cation Techniques",CS224N Final Report, 2009

[3] Go, R. Bhayani, L.Huang. "Twitter Sentiment Classification Using Distant Supervision". Stanford University, Technical Paper,2009

[4] L. Barbosa, J. Feng. "Robust Sentiment Detection on Twitter from Biased and Noisy Data". COLING 2010: Poster Volume,pp. 36-44.

[5] Bifet and E. Frank, "Sentiment Knowledge Discovery in Twitter Streaming Data", In Proceedings of the 13th International Conference on Discovery Science, Berlin, Germany: Springer,2010, pp. 1-15.

[6] Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, "Sentiment Analysis of Twitter Data", In Proceedings of the ACL 2011Workshop on Languages in Social Media,2011 , pp. 30-38

[7] Dmitry Davidov, Ari Rappoport." Enhanced Sentiment Learning Using Twitter Hashtags and Smileys". Coling 2010: Poster Volume pages 241{249, Beijing, August 2010

[8] Po-Wei Liang, Bi-Ru Dai, "Opinion Mining on Social MediaData", IEEE 14th International Conference on Mobile Data Management, Milan, Italy, June 3 - 6, 2013, pp 91-96, ISBN: 978-1-494673-6068-5, http://doi.ieeecomputersociety.org/10.1109/MDM.2013.

[9] Pablo Gamallo, Marcos Garcia, "Citius: A Naive-Bayes Strategyfor Sentiment Analysis on English Tweets", 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, Aug 23-24 2014, pp 171-175.

[10] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in Proceedings of the 40th annual meeting on association for computational linguistics, pp. 417–424, Association for Computational Linguistics, 2002.

[11] J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke, "Using wordnet to measure semantic orientations of adjectives," 2004.

[12] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," Information Sciences: an International Journal, vol. 181, no. 6, pp. 1138–1152, 2011.

[13] ZhunchenLuo, Miles Osborne, TingWang, An effective approach to tweets opinion retrieval", Springer Journal on World Wide Web, Dec 2013, DOI: 10.1007/s11280-013-0268-7.

[14] Liu, S., Li, F., Li, F., Cheng, X., &Shen, H.. Adaptive co-training SVM for sentiment classification on tweets. In Proceedings of the 22nd ACMinternational conference on Conference on information & knowledge management (pp. 2079-2088). ACM,2013.

[15] Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment Treebank." Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP). 2013.

[16] Pang, B.and Lee, L. "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts". 42nd Meeting of the Association for Computational Linguistics[C] (ACL-04). 2004, 271-278.

[17] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., &Stede, M.."Lexicon based methods for sentiment analysis". Computational linguistics, 2011:37(2), 267-307.

[18] Wan, X.."A Comparative Study of Cross-Lingual Sentiment Classification". In Proceedings of the 2012 IEEE/WIC/ACMInternational Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01 (pp. 24-31). IEEE Computer Society.2012

[19] Bollegala, D., Weir, D., & Carroll, J.. Cross-Domain Sentiment Classification using a Sentiment Sensitive Thesaurus. Knowledge and Data Engineering, IEEE Transactions on, 25(8), 1719-1731,2013