Sentiment and Emotion Analysis: A Literature Review

Machine learning and dictionary-based algorithms for sentiment and emotion analysis

Argyrios Vartholomaios
Computer Science Department
Aristotle University
Thessaloniki, Greece
asvartho@csd.auth.gr

Dimitrios Agtzidis
Computer Science Department
Aristotle University
Thessaloniki, Greece
agtzdimi@csd.auth.gr

Dimitrios Papadopoulos
Computer Science Department
Aristotle University
Thessaloniki, Greece
papadopod@csd.auth.gr

ABSTRACT

Could it be feasible to identify and classify users' emotional state based on their activity on social networks? In recent years, an extensive effort has been done to determine the users' sentiment opinion incentivizing the growth of numerous profitable advertisement policies or estimations of the general point of view in various subjects. The huge diversity of the data context (videos, images, etc) as well as the objective definition of the text analysis poses an intriguing challenge for implementing accurate and robust algorithms to classify users' emotions. The scope of this survey, is to collect a subset of the recently published articles related to the emotion classification and perform an analysis in the techniques that were mainly used to calculate their results. An effort will be done to understand and separate the different algorithms in an aggregated level regarding the approaches that were followed. The main aggregated layers that has been distinguished could be summarized as: i) Algorithms using Machine Learning approaches to classify sentiment or emotions as multi-Class / multi-Label problem, ii) Algorithmic approaches based on existing lexicons that add a semantic layer to escalate the accuracy of the algorithms or even perform a custom semantic layer to exceed the abstraction of the existing lexicons and define relative concepts to the application that will be utilized.

KEYWORDS

Sentiment Detection, Emotion Detection, Emotion Models, Sentiment Analysis, Machine learning

1 INTRODUCTION

Emotion as a human characteristic is a very complicated aspect of anyone's life regulating the behavioural patterns and actions that determine our everyday routine. Expressing the emotional state could be done by various means in different environments that bound our society. The most common way to express someone's emotions is by different face expressions as well as by conversation. Though, the evolution of the social media and networks emerged a new way to express the emotional status of someone through the social media platforms posts and interactions through texts and annotations (likes, retweets, etc.)

Although the social networking platforms give the capability to express your emotions with 'audio', 'images', 'videos' the 'text' is defined as the most commonly used interaction in social platforms. Though, due to the objectiveness and different ways of expressing the same emotion by each person makes the sentiment analysis and emotion detection in texts a really hard problem. Moreover, the expressed emotions in a single sentence may contain multiple emotions that should be identified by the corresponding algorithm.

Assuming as emotional states the following: "Angry", "Happy", "Sad", "Excitement" the aforementioned algorithms should measure a way to classify the sentence "I cannot believe that was happened to me", according to the emotions. The challenge here is that all the mentioned emotions regarding of being completely contradicting could be related to the input sentence. Furthermore, even if we define a different approach to classify the above sentence into 'Positive', 'Neutral', 'Negative' the issue still remains as it could be classified both as 'negative' or 'positive'. Therefore, automatic identification demands an intelligent way to extract all the vital information from the input texts and classify them accordingly.

To achieve that there are 2 major sectors that has showcased promising results. The machine learning approaches that will confront this problem as multi-label classification problem. The main concept of this approach is to utilize a huge amount of data that are already annotated or will be annotated on the fly and classify any new sentence given, based on the training set that the machine learning model was created. The second approach, is the utilization of lexicons containing a huge number of indicatives words regarding the emotional state that it describes. The lexicons can be used as really powerful tool as they were created by people with certain expertise in the emotion analysis and could identify difficult cases. An example lexicon that is commonly used for sentiment analysis in the algorithms which will be further discussed in the following sections is 'SentiWordNet'. Additionally, lexicons have been expanded to cover apart from the traditional linguistic features more innovative forms of expressions such as emoticons.

In our review paper we present research that has been conducted on algorithms using one or both of the aforementioned techniques and we display an analysis of the different implementation with multiple KPIs that will indicate the accuracy and the performance of each one of them. The general architecture that was followed for sentiment and emotion classification can be seen in **Figure 1.**

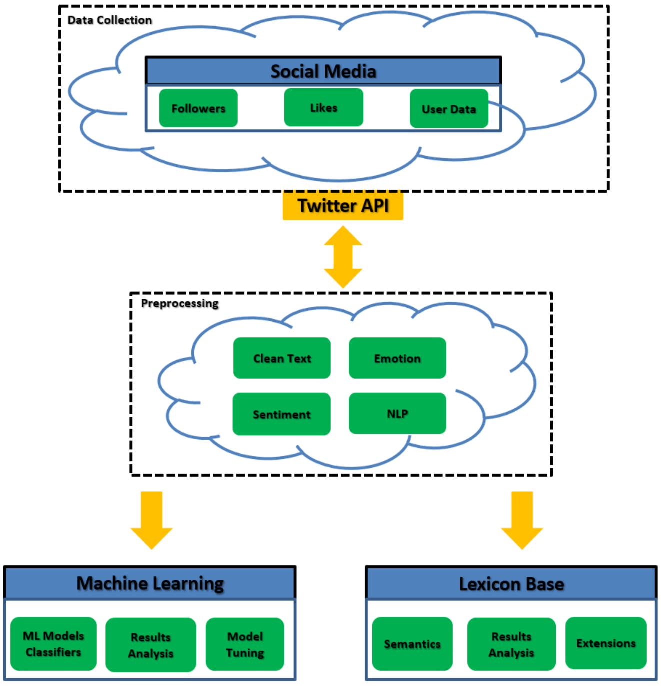

Figure 1: Data Flow Architecture

For the remainder of this research review we present several attempts to address the problem of sentiment and emotion analysis. In Section 2 we state the importance of sentiment analysis and opinion mining as well as the methods employed in the task. Furthermore, in Section 3 we explore the field discoveries of 12 research papers [4,5,6,7,8,10,11,12,21,23,24] by grouping their achievements in two major categories according to the approach followed. For each method we present the domain, analyze the model and demonstrate the evaluation method used. Lastly, we conclude this review by adding our remarks and suggestions for further exploration on the domain of sentiment analysis.

2 PROBLEM DEFINITION

Since the late 2000's the rise of social media boosted the user data generation exponentially. According to [1] as of 2019 there are approximately 3 billion social media users that generate content by interacting with each other via popular social media platforms such as Facebook, Twitter, Tumblr, Reddit and so on. Social web has offered an outlet for people to publish their opinions, express their views and promote themselves and their ideas.

As of recent, opinion mining has been a popular subject of research with the rise of microblogging platforms [2]. Especially since opinionated personal commentary is difficult to acquire outside product reviews or direct customer feedback.

Opinion mining, is the process of deriving the opinion or attitude of a speaker, also known as sentiment analysis the method of 'computationally' determining whether a piece of writing is positive, negative or neutral. Equally popular is the task of sentiment dissemination that further analyzes an emotional response by attributing labels such as "Angry", "Happy", "Sad", "Excitement". Sentiment analysis is instrumental in deploying targeted marketing strategies and promotion of ideas mainly in the following sectors:

- **Business:** In marketing field companies use it to develop their strategies, to measure brand recognition and brand loyalty, to understand customers' feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don't buy some products. Furthermore, understanding the current emotional status behind certain topics helps organizations correctly align their proposition, messaging, or product to match trending opinion.
- **Politics:** In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well! According to Forbes the Obama administration employed sentiment analysis during the 2012 presidential election [3] and ever since politicians and brands have continued to improve their tools to glean clearer insights to understand the public's attitude and opinion.
- **Public Actions:** Sentiment analysis also is used to monitor and analyze social phenomena, such as hate speech detection or spotting potentially dangerous situations and determining the general mood of the blogosphere.

The research behind the task of sentiment analysis is classified in two major approaches. One approach is based on machine learning which involves extracting the sentiment by learning from previously annotated data. In this category we include methodologies that involve neural networks directly or by training sentiment-specific embeddings. The second approach requires the usage of lexicons that contain information of the polarity of a (positive, negative, neutral) emotions. Variations of this method can be performed in unsupervised learning without prior training [4]. Finally, combined features from the machine learning and the semantic-oriented approach were found to be quite successful when applied as a hybrid approach.

There are many sources for collecting the required data for the sentiment classification task. By far the most popular and easily accessible source is Twitter [4,5,8,11,12,14,17] which provides a comprehensive API. Additional sources are Amazon product reviews, Reddit posts and other online communities such as the Experience Project (EP) [10] Lastly a popular technique is training word embeddings such as in Word2Vec, a Google's project that offer embedding trained in English Wikipedia articles. Finally, most dictionary methods are based in publicly available dictionaries like WordNet that produces word associations known as synsets used in many natural language processing tasks (NLP) such as part-of-speech tagging, sentiment analysis, information retrieval and text summarization

Table 1: Overview of the literature review with the type of algorithm and the reference number

Method	Reference
Dictionary-based	[4, 5]
Machine Learning	[8, 10, 11, 12, 21, 23, 24]
Hybrid	[6, 7, 15]

.

3 MODELS

In this section we showcase the work of 12 research papers in the domain of sentiment analysis. We proceed to group the papers in two major categories according to the method used. Firstly, we present the lexicon-based approaches and secondly the methods that employ machine learning techniques.

3.1 Lexicon based methods

The authors in [4] applied a lexicon-enhanced pipeline that leverages the sentiment of product reviews by combining content-free features along with existing content-specific features used in machine learning approaches. Content-free features include lexical, syntactic, and structural features, whereas content-specific features consist of important keywords and phrases on certain topics, such as word n-grams. The dataset used for the experiments consisted of user product reviews acquired by mining sources such as epinions.com and applied on Blitzer's multi-domain sentiment data [2] a set of four publicly available testbeds. In total, 307 negative reviews and 1,499 positive reviews were collected of which 307 positive and 307 negative were combined with 1000 positive and 1000 negative reviews of the Blitzer's dataset.

The proposed sentiment classification method involves the creation of three types of feature vectors. The first feature vector (F1) consists of content-free features, the second (F2) contains content-specific features, whereas the third (F3) hosts the sentiment-specific features. Features F1 and F2 are produced from the machines learning approaches, and F3 is from a semantic-oriented approach. In total 250 F1 features were utilized of which 87 are lexical features, 158 syntactic features (150 function words and eight punctuation marks), and 5 structured features. Furthermore, the F2 features consists of frequent unigrams and bigrams produced after removing the semantically empty stop words.

Finally, F3 features were extracted by applying a number of natural language processing techniques. Firstly part-of-speech tagging was applied to each data collection. Afterwards, a dictionary-based method was implanted utilizing the SentiWordNet dictionary to extract the sentiment of each word. SentiWordNet is a publicly available lexical resource, based on WordNet, where each synset s is associated with three sentiment scores that measure the positivity, negativity and neutrality of the word in a 0 to 1 scale. Because each synset consists of multiple words the authors calculated the average of each polarity score according to the part of speech. The final score is calculated taking in account the objectivity and subjectivity of each word by setting the objectivity threshold at 0.5 in the 0 to 1 scale. Thus, words that score high on objectivity (>0.5) are ignored as they are not used to promote emotions, whereas words equal or below 0.5 are considered subjective. If the positive score is greater than the negative, the words is classified as positive and vice versa. Words with equal negativity and positivity scores are excluded from the F3 feature vector.

The authors conducted their experiments using four different combinations of the three feature vectors described above:

1. Feature set F1 include content-free features.
2. Feature set (F1 + F2) consists of content-free and content-specific features.
3. Feature ser (F1 + F3) consists of content-free and sentiment features
4. Feature set (F1 + F2 + F3) consists of content-free, content-specific, and sentiment features.

Additional features sets were created by applying the Information Gain (IG) feature selection heuristic in F1 + F2 and F1 + F2 + F3 feature sets.

The dataset for the experiments consists of user reviews on 5 kind of products, digital cameras, books, DVDs, electronics and kitchen appliances and was split in 90-10 ratio. The proposed feature sets were tested using an SVM classifier after performing a 10-fold cross validation. The evaluation of the model was measured using the overall accuracy, average precision, average recall, and average F-measure for all five testbeds. The average evaluation metrics of all 5 datasets against the best performing features can be seen in **Table 2**.

Table 2: Average performance of feature selection across all review datasets [4]

Feature set	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
F1+F2+F3	80.5	80.58	80.56	80.57
Selected F1+F2+F3	81.31	81.67	81.27	81.51

In [5] the authors attempt to capture the wisdom of the crowds through a lexicon-based approach on micro-blogging data. Chatzakou et al. proposes a model that capitalizes on the opinions expressed in the Twitter platform and by measuring their intensity proceeds to map them to the six primary emotions. Their method is based on WordNet-Affect, a lexicon that is an extension of WordNet and contains synsets suitable to represent affective concepts correlated with affective words.

In their research Chatzakou et al. proceeded to construct vectors that represent tweets and emotions as sets $T = \{t_1, \ldots, t_m\}$ and $E = \{e_1, \ldots, e_l\}$ respectively. Another set $ER_i = \{r_{i1}, \ldots, r_{ip}\}$ was formed by the representative synsets of affective words (e_i) taken by WordNet-Affect lexicon. Additional synsets of the representative words were added to ER_i set as well, thus expanding the list of representative words. Moreover, for each tweet a set ET_i was created that contained words in the tweet that stemmed from words in the ER_i set. Finally, a set $EL_i = \{el_{i1}, \ldots, el_{ir}\}$ was created by the linguistic emotion representations known as emoticons.

The SentiWordNet dictionary was used in this research as well, to measure the sentiment score of each word. In addition, a list of emotion intensifier words was taken into account to properly measure the emotion. Words such as “quiet”, “hardly”, and “very” were added to this list and attributed a specific score $(intens_j)$. The overall score of an emotional word in a tweet (et_{ir}) is calculated by the following formula:

$$SCI(et_{ir}) = \left(1 + score\left(intens_j\right)\right) * score(et_{ir})$$

where $score(et_{ir})$ is the sentiment score of word et_{ir}, taken by SentiWordNet dictionary. Furthermore, to account for valence sifters in a tweet (words such as “not”, “cannot”, “never”, “no”, etc.) the score $SCI(et_{ir})$ is calculated as:

$$SCI(et_{ir}) = 1 - score(et_{ir})$$

Finally, the sentiment score of each emoticon (el_i) was based on a “sentislang” dictionary provided by the university of Maryland that contains the 55 more popular tokens scored in a 0 to 1 scale. The overall score of a tweet t_i was also weighted by the term frequency of each emotional word et_{ir} which is denoted as $tf(et_{ir})$. Given all the parameters discussed the final score $SC(t_i, e_i)$ between tweet t_i and emotion e_i is calculated by the formula:

$$SC(t_i, e_i) = \frac{\sum_{\forall et_{ir} \in ET_i} tf\left(et_{ir}, e_j\right) * SCI(et_{ir})}{\sqrt{\sum_{\forall et_{ir} \in ET_i} tf\left(et_{ir}, e_j\right)^2}}$$

The product of Chatzakou et al. was the EmoGrabber algorithm that combines in a comprehensive pipeline text preprocessing, the dictionary approach presented above and a clustering algorithm that groups the tweets according to their emotional context. The clustering is performed using KMeans which groups together tweets with similar score regarding specific primary emotions.

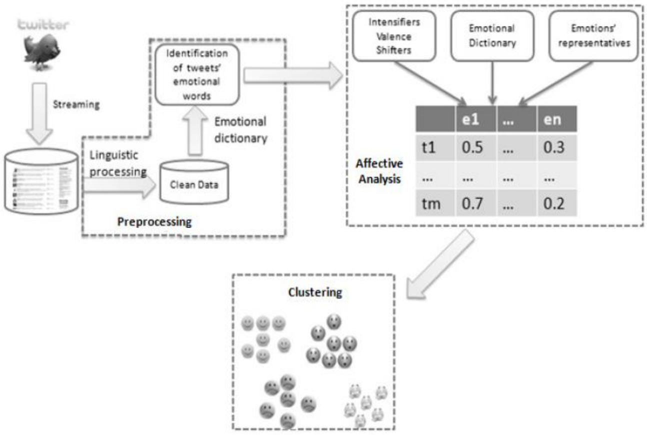

Figure 2: The EmoGrabber pipeline [5]

The evaluation of EmoGrabber was performed in a dataset consisting of 9500 tweets that were posted around Christmas in the area of London. The evaluation of the application was conducted by exploring the correlation of the emotions captured. Thus, negative emotions such as anger and disgust would have closer correlation.

Table 3: Correlation matrix of captured emotions [5]

	Anger	Disgust	Fear	Joy	Sadness	Surprise
Anger	1.000	0.347	0.050	-0.125	0.045	-0.023
Disgust	0.347	1.000	0.049	-0.093	0.036	-0.036
Fear	0.050	0.049	1.000	-0.034	0.051	-0.002
Joy	-0.125	-0.093	-0.034	1.000	-0.031	0.233
Sadness	0.045	0.036	0.051	-0.031	1.000	-0.049
Surprise	-0.023	-0.036	-0.002	0.233	-0.049	1.000

In the research work [6], a different approach was used for the sentiment analysis. The suggested implementation contains different techniques in order to obtain a semantic representation of the input text in two different layers i) the word layer, gaining insights per word inside a text and the sentence layer where aggregation techniques will be applied. Hence, the two different layers will be transformed into input vectors and will be utilized in an RTNN (Recursive Neural Tensor Network) which is widely acknowledged as a strong classifier for identifying meaning in sentences. The main objectives of this work are to present a sequence for creating and combining both text layers (word, sentence) in a single algorithm called MULTISPOT. Additionally, a different technique is showcased that is based on the overall context of the text document itself as a center-based aggregation method. Hence, in data processing phase the feature vectors were constructed based on:

Word-Based: The unigram Bag of Words and bi-gram models of Naïve Bayes used to gain the word representation

Sentence-Based: Classifiers were used to gain a semantic aggregation of the sentences based on the objectivity and the semantic analysis of each word

Those feature vectors were used as input in the RTNN to gather the sentiment results. The second approach i.e center-based is highly related on capturing a semantic dictionary per sentence for the whole document. Hence, the implementation proposed to cluster the sentences based on their semantic representation, in order to obtain a mapping of each sentence and the nearest cluster. After encoding the high-level features that was created the vectors will be fed to the RTNN for the sentiment analysis. The evaluation of this model produced some promising results. Below a table is presented with the best outcome of the model's results having as features both the word & sentence base semantics as well as the center-based feature of the text content. The comparison below showcases also the different results acquired from the BoW and Bigram models.

Table 4: Evaluation of the MultiSpot method using NB-bigram features and BoW [6]

-	Accuracy (%)	Recall (%)	F1 (%)
Bigrams + Features	91.6	91.33	91.58
BoW + Features	89.48	89.19	89.45

Another approach was proposed in [7] that in its unsupervised algorithm embraces the SentiWordNet lexicon for sentiment analysis. The SentiWordNet lexicon regulates the sentiment of each word by integrating a certain function in a set of words that will constitute the corresponding sentence. Each of the words that the SentiWordNet lexicon contains is mapped to a specific weight for each term.

To further enhance the lexicon-based approach the authors implemented a technique to add words that were not contained on the original lexicon for the training of the model. The purpose of this, was the inability of the lexicon-based techniques to identify content related terms of the text that is being processed. In their approach they taking into account as ‘‘excellent’’ to be an extremely positive word and ‘‘poor’’ to be an extremely negative word. Alongside with the above words, also the synonyms were used to pose a positive and a negative list. NAVA (Noun, Adjective, Verb, Adverb) NLTK POS tagger was utilized so that every sentence was tokenized into those specific POS tags. For each remaining word, Google was utilized to determine a certain defined metric as the score of its word about its positivity or negativity. A score was generated for each word(t) using the following rule:

$$score(t) = \log\left(\frac{hits(t \wedge excellent) * hits(poor)}{hits(t \wedge poor) * hits(excellent)}\right)$$

After generating the score for each word it is classified either as positive if the score is above zero, otherwise it is classified as negative.

To validate their findings the authors conducted test cases using the SentiWordNet lexicon alone and with the defined score metric that they have implemented. A showcase of the test cases can be seen in **Table 4**. The dataset that was used to obtain the accuracy was a large number of twitter feeds.

Table 5: performance comparison between SentiWordNet Lexicon approach and Score Metric [7]

-	Accuracy (%)
SentiWordNet Lexicon	91.6

SentiWordNet + Score Metric	89.48

3.2 Machine learning based methods

As for the machine learning approaches, we initially explore [8], a research on Sentiment-Specific Word Embeddings (SSWE) trained and used on Twitter data. The model presented here was based on an earlier word embedding learning algorithm $C\&W$ [9]. In $C\&W$ model Collobert et al., replaces the center word in a given ngram with a random word w^T and produces a corrupted ngram. During the training the original ngram is expected to prevail over the corrupted and receive a higher score. The objective function for training $C\&W$ is:

$$loss_{cw}(t, t^r) = \max\left(0{,}1 - f^{cw}(t) + f^{cw}(t^r)\right)$$

Where t is the original ngram, t^r is the corrupted ngram, $f^{cw}(.)$ is a one-dimensional scalar representing the language model score of the input ngram. The $C\&W$ is applied through a feed forward neural network whose architecture consists of four layers $lookup \rightarrow linear \rightarrow hTanh \rightarrow linear$. The inputs of the neural network are the original and corrupted ngrams and the output is the language model score:

$$f^{cw}(t) = w_2(a) + b_2$$

where L is the lookup table of word embedding, w_1,w_2, b_1, b_2 are the parameters of linear layers. Additional parameters are denoted as:

$$a = hTanh(w_1 L_t + b_1)$$

$$hTanh(x) = \begin{cases} -1, & x < -1 \\ x, -1 \leq x \leq 1 & \\ 1, & x > 1 \end{cases}$$

Similarly, to $C\&W$ the authors in [8] construct two Sentiment-Specific Word Embedding models, expanding the original in an attempt to capture the sentiment in a text. In $SSWE_h$, the model applies a sliding window for the ngrams across a sentence, that predicts the sentiment polarity based on each ngram with a shared neural network. In the feed forward neural network, the distributed representations of higher layer are interpreted as features describing the input. Thus, the continuous vector of top layer is utilized to predict the sentiment distribution of text. Assuming there are K labels, the dimension of top layer in C&W model is modified as K and a softmax layer is added upon the top layer. The cross-entropy error of the softmax layer is calculated as:

$$loss_h(t) = -\sum_{k=\{0,1\}} f_k^g(t) * \log\left(f_k^h(t)\right)$$

The $SSWE_h$ is trained by predicting the positive ngrams as [1,0] and the negative as [0,1]. However, since this is too strict of a constrain an additional model is proposed that relaxes this behavior. $SSWE_r$ predicts the sentiment in a more stochastic way by considering the sentiment polarity of a tweet as positive if the positive score is higher than the negative and vice versa. This relaxed model shares almost the same architecture as $SSWE_h$ ignoring the softmax layer. The hinge loss in $SSWE_r$ is measured as:

$$loss_h(t) = \max\left(0, 1 - \delta_s(t) f_0^r(t) + \delta_s(t) f_1^r(t)\right)$$

where f_0^r and f_1^r are the predicted positive and negative scores respectively and $\delta_s(t)$ is the indicator function:

$$\delta_s = \begin{cases} 1 \\ -1 \end{cases} if \begin{array}{l} f^g(t) = [1{,}0] \\ f^g(t) = [0{,}1] \end{array}$$

The comparison of the architecture of the three models is illustrated in **Figure 3**.

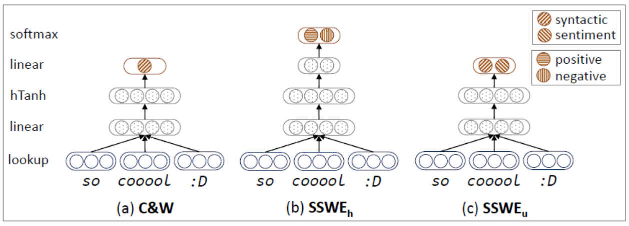

Figure 3: The C&W model and the proposed neural networks (SSWEh and SSWEu) [9]

Finally, the authors of [8] propose a unified embedding model $SSWE_u$ that captures the semantic information from $SSWE_r$ and $SSWE_h$ as well as the syntactic information of $C\&W$. Given an original (or corrupted) ngram and the sentiment polarity of a sentence as the input, $SSWE_u$ predicts a two-dimensional vector for each input ngram.

During training of $SSWE_u$ the original ngram should produce higher language model score $f_0^u(t)$ than the corrupted ngram $f_0^u(t_r)$, and the sentiment score of original ngram $f_1^u(t)$ should be more consistent with the polarity annotation of sentence than corrupted ngram $f_0^u(t_r)$.

The embeddings produced by the models described above were used as input to min, $average$ and max convolutional layers in a supervised framework for Twitter sentiment classification. Each convolutional representation employs the unigram, bigram and trigram embeddings. The result of each convolutional layer z_x, where $x \in \{min, max, average\}$ is:

$$z_x(tw) = [w_x < L_{uni} >^{tw}, w_x < L_{bi} >^{tw}, w_x < L_{tri} >^{tw}]$$

where w_x is the convolutional function of z_x, $< L >^{tw}$ is the concatenated column vectors of the words in the tweet for the unigrams, bigrams and trigrams.

The evaluation of the embeddings was conducted by comparing SSWE with other feature embeddings on the SemEval 2013 benchmark dataset using the LibLinear (Fan et al., 2008) classifier and using Macro-F1 metric.

Table 6: Macro-F1 on positive/negative classification [9]

Method	Macro-F1
DistSuper + unigram	61.74
DistSuper + uni/bi/tri-gram	63.84
SVM + unigram	74.50
SVM + uni/bi/tri-gram	75.06
NBSVM	75.28
RAE	75.12
NRC (Top System in SemEval)	84.73
NRC - ngram	84.17
SSWEu	84.98
SSWEu+NRC	86.58
SSWEu+NRC-ngram	86.48

Another research using a machine learning approach is presented in [10] where the authors develop a model based on a semi-supervised training of Recursive Autoencoders (RAE), that learns vector representations of phrases and sentences along with their hierarchical structure from unsupervised text. An *autoencoder* is a neural network that learns to copy its input to its output. It consists of a hidden layer that represents the encoder of input, and can be divided in two main parts: an encoder that translates the input into the encoding, and a decoder that translates the code to an approximate reconstruction of the original input. For an autoencoder, performing the copying task perfectly would simply duplicate the signal, and this is why autoencoders usually are restricted in ways that force them to reconstruct the input approximately, propagating only the most relevant aspects of the data to the copy. The goal of training a RAE is to minimize the reconstruction error. Lastly, RAE models can be illustrated by unraveling the recursion in a binary tree.

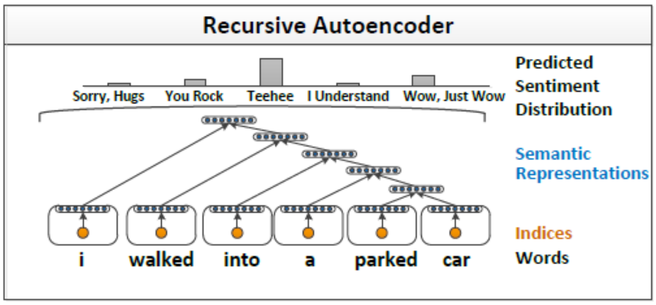

Figure 4: Illustration of the recursive autoencoder architecture which learns semantic vector representations of phrases [10]

The proposed RAE model performs the sentiment label distribution without the use of lexicons or a bag-of-words representation. Instead, it exploits the hierarchical structure using compositional semantics trained in both labeled and unlabeled domain data and on labeled sentiment data. Finally, the model is not just limited to a positive/negative differentiation and it predicts a multidimensional distribution over complex and interconnected sentiments.

The model is trained on data acquired by the Experience Project at www.exprerienceproject.com that contains anonymous personal confessions. The confessions are labeled with a set of five reactions by the users, these reactions are:

i. “You rock” - expressing approval
ii. “Tehee” - amusement
iii. I understand - compassion
iv. Sorry, hugs - sympathy
v. Wow, just wow - displaying shock

The model predicts both the label with the most votes as well as the full distribution over the sentiment categories.

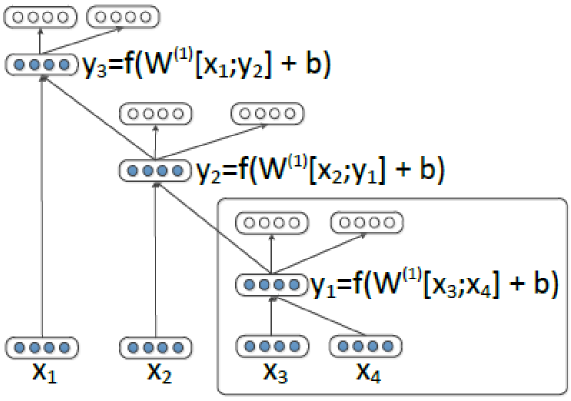
$y_3=f(W^{(1)}[x_1;y_2] + b)$
$y_2=f(W^{(1)}[x_2;y_1] + b)$
$y_1=f(W^{(1)}[x_3;x_4] + b)$
x_1
x_2
x_3
x_4

Figure 5: Illustration of RAE to a binary tree. The nodes which are not filled are only used to compute reconstruction error. A standard autoencoder (in box) is re-used at each node of the tree [10].

Figure 5, displays the architecture of a RAE, each child in the binary tree is either an input word vector $x = (x_1, \dots, x_m)$ or a nonterminal node in the tree. Given this representation the parent node is computed from the children of the previous step. For instance, the parent vector y_1 is computed by the children c_1, c_2 that stands for the word vectors x_3 and x_4. In order to assess how well the parent node represents the children a reconstruction is attempted:

$$[c_1'; c_2'] = W^{(2)}p + b^{(2)}$$

The model is trained by minimizing this reconstruction error using the Euclidean distance:

$$E_{rec}([c_1; c_2]) = \frac{1}{2}\left|\left|[c_1; c_2] - [c_1', c_2']\right|\right|^2$$

The unsupervised RAE model is applied to perform a structural prediction assuming there is no input vector x. The goal is to minimize the sentence reconstruction error of all vector pairs in a tree. Assuming, $A(x)$ is the set of all possible trees that can be built from an input sentence x the reconstruction error is:

$$RAE_\theta(x) = argmin_{y \in A(x)} \sum_{s \in T(y)} E_{recc}([c_1; c_2]_s)$$

where $T(y)$ is a function that returns triplets of a tree indexed by s of all the non-terminal nodes in a tree. This tree is constructed by feeding into the autoencoder word vectors starting with the pair of the two first words in the sentence and taking into account the potential parent node and the reconstruction error.

The RAE model was trained on English Wikipedia unlabeled corpus and using the 49% of the Experience Project labeled dataset. The model was tested in the task of predicting the class popularity and by measuring the sentiment distribution against other feature creation methods.

Table 7: Popularity prediction of RAE [10]

Method	Accuracy
Random	20.0
Most Frequent	38.1
Baseline 1: Binary BoW	46.4
Baseline 2: Features	47.0
Baseline 3: Word Vectors	45.5
RAE	50.1

An interesting methodology was proposed in the research work of [11] where the dataset was not only containing the user defined text (posts) but also the replies that it had received. In order to create and demonstrate their methodology they collected manually a sample of twitter posts with their replies. Another intriguing approach they wanted to research was to calculate the sentiment and emotion analysis in two different ways:

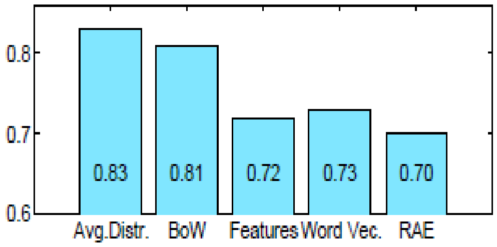

Figure 6: Average KL-divergence between gold and predicted [10]

Generalized approach: In this approach they collected twitter data based on specific topics and used the text as abstract without adjusting a weight from where it had been generated

User Based approach: In this approach an effort has been done to apply some weighting on the gathered input and create a user specific content that will take into account variables that are not differentiate in the general approach

To gather this kind of information some worth mentioning difficulties has been raised without thoroughly providing a solution. Some of the difficulties were:

i. The different language used on multiple twitter comments as a reply to the initial post
ii. The different context used as a reply (e.g memes pictures)
iii. Advertisement companies commenting on highly popular hashtags

After the data collection, the authors standardized a sequence of steps to preprocess the data removing stop words and cleaning the input. One of the features that was proposed to be taken into account for the classification task was also the POS tagging and specifically they kept only the NAVA.

For the classification task, they have integrated a Naïve-Bayes approach mentioning that it had the best results from other classifiers and lexicon-based solutions. The approach followed for Naïve Bayes algorithm was to apply cross validation with different k-folds on the preprocessed text to gain different views generated and tested from the input. Therefore, the classification has been done both on NAVA text and the complete preprocessed text. The extracted features that was fed to the classifier included words per tweet and reply. This dataset was used twice, both for sentiment and emotion analysis.

The author presented some worth mentioning outcomes from the evaluation phase of their algorithm. As we can observe in the below table using full-text instead of only NAVA features increase the accuracy of the model. Also, it is observed that increasing the *k* in cross-validation gradually improves the accuracy as well. Moreover, an intriguing observation was the comparison between accuracies of the classifier in sentiment and emotion analysis. In sentiment classification, the multi class issue contained three classes i.e positive, negative, neutral and on the other hand the emotion analysis contained seven classes i.e anger, disgust, fear, joy, sadness, surprise and neutral. As the number of classes in emotion analysis has bigger margin it is noticed that the accuracy gradually downturned a little bit compared to the sentiment classifier.

Table 8: Sentiment and emotion classification accuracy (First 3 Columns Sentiment, last 3 Columns Emotion) [11]

Text type	NB	SVM	Random Forest	NB	SVM	Random Forest
Full-Text	66.86	23.32	55.23	47.34	14.48	35.66
NAVA words	61.15	23.32	52.01	43.24	14.48	37.26

Table 9: Sentiment and emotion classification accuracy for Naïve Bayes [11]

k-fold	Full Text Sentiment	NAVA Sentiment	Full Text Emotion	NAVA Emotion
3-fold	62.69	55.96	44.37	40.45
5-fold	63.71	58.33	46.32	41.29
10-fold	66.86	61.15	47.34	43.24

Another approach to identify sentiment was introduced in [12]. The authors proposed a convolutional neural network constituted by a single layer followed by a non-linear activation function, max pooling and a softmax classification layer. The dataset that they have used to train their model was mostly twitter posts. The main aim of the algorithm was to create compact embeddings from each word per twitter post and create a sentence matrix where each of the column *i* will contain information for the embedding w_i at the corresponding position in a sentence:

$$\mathbf{S} = \begin{bmatrix} | & & | \\ \mathbf{w}_1 & \dots & \mathbf{w}_{|s|} \\ | & | & | \end{bmatrix}$$

The defined steps of the implementation can be concluded as:

Creation of word embeddings utilizing a neural language model (Word2vec Mikolov et al., 2013). Word embeddings is the largest and most crucial feature of the implementation. Therefore, to train the matrix of the embeddings a skip gram model with window size 5 and filtering words with frequency less than 5 was applied into 50M of twitter posts.

In the first step the word embeddings will contain a very solid structure regarding the important semantic aspects of the text's content. Although, these embeddings lack on what can be appropriate for the sentiment analysis. At this step the authors used a distant supervision approach (Go et al., 2009) to further tune the embeddings.

The produced word embeddings and relative parameters that will adjust their weights will be used to initialize the neural network and trained on a supervised corpus from Semeval-2015.

The aforementioned algorithm was introduced and took part in the Semeval-2015 competition, ranking 1st in the phrase-level task and 2nd on the message level task.

The following table summarizes the performance of this model in different datasets with three different initialization parameters. The first parameter schema is a random initialization that will produce a medium performance. This is an outcome related to the generally small size of the training set. In the second case word2vec embeddings that were pre-trained were used and increase significantly the performance of the model. The final initialization schema, uses distant supervised corpus to further enhance the pre-trained embeddings. As a result, this model captures the sentiment context of the different words and boost the accuracy of the model.

Table 10: Results accuracy based on initialization schemas: Random word embeddings, word2vec embeddings, Distant (all parameters from a network trained on a distant supervised dataset). [12]

Dataset	Random	Unsup	Distant
LiveJournal'14	63.58	73.09	72.48

SMS’13	58.41	65.21	68.37
Twitter’13	64.51	72.35	72.79
Twitter’14	63.69	71.07	73.60
Sarcasm’14	46.10	52.56	55.44

A supervised approach using machine learning techniques was introduced in [15]. Specifically, for the features’ creation of the algorithm a BoW representation was used where a tweet serves as a bag of its words tokenized, without taking into consideration relations and grammar connections. To further filter the bag of words the authors filtered out irrelevant words and converted the dataset to unigrams/bigrams/trigrams or combination of them. From the created N-Grams the authors selected the more representative one to be used as the feature vectors that will be used for the classifier. Although, no mention was made on how the whole preprocessing has happened and mostly described the sequence steps that a machine learning algorithm should follow but not what they have actually implemented. As their classifier Multinomial Naïve Bayes was used after applying Laplace Smoothing to the feature vectors to remove zero probabilities that may occur because of the zero numbers in the N-Grams. The authors conducted several test cases to evaluate their model. Their findings were that generally users express their emotional state or sentiment by using mostly hashtags, so capturing the hashtag text significantly increases the accuracy of the model. The hashtag text alongside with other features were used for the MNB classifier and the results can be observed on **Tables 10, 11.**

Table 11: Accuracy of emotion dataset using different features [15]

-	Accuracy (%)
Bigram	71.23
Unigram & Bigram	95.3
POS	92.9

Table 12: MNB classifier for unigram feature [15]

-	Accuracy (%)	Recall (%)	F1 (%)
Anger	99.48	97.98	98.72
Fear	94.95	96.72	95.82
Joy	92.19	96.42	94.25
Love	95.90	95.90	95.90
Sad	86.89	98.35	92.26
Surprise	99.50	97.59	98.53

On [16] work, a machine learning approach is used to build a sentiment classification model, trained on different vectorized representations of textual data. More specifically the proposed vectorization techniques are word embedding based, lexicon-based feature extraction and a hybrid of the previous methods.

The lexicon-based method uses a lexicon containing terms of a specific language that carry emotion, annotated typically from human experts or machine learning algorithms, in a certain number of dimensions. The term annotation procedure of a lexicon can be done considering their subjectivity or their polarity and even thought the proposed framework can be followed using common sentiment lexicons, the use of lexicons that include emotion dimensions is preferred as it can improve classification accuracy in sentiment detection. In their work they used two lexicons, one in English and one in Greek. For the English they chose a large and rich regarding the number of dimensions lexicon called EmoLex (*NRC Word-Emotion Association Lexicon)*. Emolex contains 14,182 terms, binary annotated as positive or negative but also accounts for the eight primitive emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, and trust) suggested in Plutchik (1994). As many of the terms included in this lexicon do not carry any emotional meaning, thus offering no value with respect to sentiment classification, these terms were removed leaving the lexicon with 6,468 terms. Similarly, regarding the Greek language, the authors choose to use a lexicon designed by (Tsakalidis, Papadopoulos, & Kompatsiaris, 2014) containing 2315 terms in Greek annotated regarding subjectivity, polarity and the six primitive emotions of Ekman (1992) (happiness, sadness, anger, fear, disgust, surprise). Due to pour term coverage of 59% upon the documents, the authors created eGreekSentLex (with 4658 terms), by expanding the lexicon with synonyms and bringing the coverage to 74%.

Next step, for the vectorization of the data, three different techniques were tested.

- The first one is a Bag of Words (BoW) approach with vectors of size equal to the number of the different terms contained in the lexicon.
- The second one is building vectors that hold the average emotion with a vector of size equal to the number of different emotions.
- The last one is constructing e vector a concatenation of the previous two.

In regard to reducing vector dimensionality the authors excluded from the lexicon, terms that were present under a certain number of times in the data. To handle negation, they built a list of negation terms in both English and Greek and scanned the data for the presence of these terms, in order to apply one of the following techniques to a certain number of words that followed a negation word.

- Reverse the feature values of that word.
- Double the size of the constructed vectors in order to create space for equal number of dimensions represented their negative counterparts.

As for the Greek language due to the strong tendency to have numerous spelling mistakes on user created online source, extra processing was necessary. This involved a lemmatization step of the texts that also fixed the regularly missing intonation.

Even though the lexicon approach to feature extraction effectively captures the overall sentiment of a large piece of text, it cannot capture the refined characteristics or complex emotions such as irony. Therefore, word embeddings can be utilized in order to increase the algorithm's ability for semantic and syntactic understanding of the text data. A popular embedding approach is Word2Vec (Mikolov et al., 2013). Using deep learning and learning from large corpus of documents, captures the context of words based in their co-occurrence in the text. Word2Vec can be used pre-trained on its large-scale data, trained from the start or by training the already trained model in order to update its weights and thus help it gain more insight in a certain domain. After preprocessing without lemmatization the texts, the documents were separated to their sentences and applied Word2Vec to create a vector representation for every word of the sentence. Afterwards, they derived a vector for every sentence, as an average of all word vectors of the sentence and did the same with the sentence vector in order to produce the documents vectors.

Finally, the authors used a hybrid approach to feature extraction combining both aforementioned techniques, by concatenating their vectors to one.

Regarding evaluation datasets, the authors used (Agathangelou et al., 2014) dataset of electronic product reviews of a popular Greek e-shop. The reviews have a score ranging from 1 to 5, so the ones and two considered negative and the fours and fives, positive. After oversampling the negative class (minority class) they created the final dataset MOBILE-PAR consisting of 1976 reviews for training and 3329 for testing. They have also created a second dataset MOBILE-SEN extracting 1768 reviews from the same e-shop and manually annotating them as positive or negative. For the English another two datasets have been used. The first is a popular movie review dataset MOVIES (Pang & Lee, 2005), consisting from 10,662 sentences, annotated as positive or negative by its authors. Lastly the research team used the Large Movie Review Dataset (IMDB) of 25,000 training and 25,000 testing reviews.

As for the evaluation of the above framework, [15] utilized k-fold validation on the dataset that were not already split, while used one SVM with linear and another one with RBF kernel. In order to find the best parameter values for C and gamma, grid search was used. The performance of the models was measured with the accuracy and F-score metrics.

Table 13: Datasets used for experimentation. The number of training and test documents corresponds to a single fold (when applicable). The percentage in the parentheses corresponds to the positive class.[15]

Dataset	Language	Training documents (pos%)	Test documents (pos%)	Folds
MOVIES	English	9596 (50.0)	1066 (50.0)	10
IMDB	English	25,000 (50.0)	25,000 (50.0)	-
MOBILE-SEN	Greek	2520 (50.0)	280 (50.0)	10
MOBILE-PAR	Greek	1976 (51.2)	3329 (84.6)	-

Table 14: Comparison between SVM-Linear vs SVM-RBF on MOBILE-SEN dataset. [15]

Vectorization/corpus	SVM-Linear accuracy (%)	SVM-RBF accuracy (%)
Lex1	72.79	74.88
Lex2	73.32	75.00

Lex3	63.39	66.43
MOB-GR	71.79	69.76
WIKI-GR	69.79	70.83
WIKI-MOB-GR	70.89	70.60
W2V+Lex1	74.36	75.71
W2V+Lex2	74.93	76.79
W2V+Lex3	71.43	73.21
***W2V+Lex1**	78.00	75.95
***W2V+Lex2**	78.57	77.62
***W2V+Lex3**	71.79	72.14

Table 15: Best achieved accuracy and corresponding SVM parameters for the lexicon-based vectors. [15]

Dataset	Vectorization scheme	C	Accuracy (%)	F-score
MOVIES	Lex1	0.01	52.29	0.2096
	Lex2	0.1	64.17	0.6062
	Lex3	10	60.20	0.5775
IMDB	Lex1	0.1	82.70	0.8300
	Lex2	0.1	83.00	0.8300
	Lex3	0.1	70.80	0.7100
MOBILE-SEN	Lex1	0.1	72.29	0.6954
	Lex2	1	73.32	0.7000
	Lex3	1	63.39	0.6339
MOBILE-PAR	Lex1	10	76.70	0.8570
	Lex2	100	78.10	0.8680
	Lex3	10	58.10	0.6950

In terms of performance, it has been observed that the SVM-Linear was one order of magnitude faster than the SVM-RBF which held true for all tested datasets. As far as accuracy is concerned the was no major difference between the two.

- Lex1. Mixed Representation (comprising BoW-REVERSE and Average emotion-REVERSE representations)
- Lex2. Mixed Representation (comprising BoW-DOUBLE and Average emotion-DOUBLE representations)
- Lex3. Average emotion-REVERSE representation.

Between the three vectorization techniques it seems that Lex2 is the best one for sentiment classification where as Lex1 remains a good option when there are limitations about vector dimensionality.

For the emerging based vectors, the authors compered three word embedding-based vectors. One was to train a Word2Vec model on the train datasets, another involved training it on larger generic corpus and the last a combination of the previous two. Meaning that a Word2Vec model was trained at first with large generic data and then updated by training with the training data.

Regarding the hybrid approach the authors observed an improvement on accuracy around 5.25% to 5.5% in the Greek corpora in comparison to lexicon representations and 7.68 to 10.2% in comparison to Word2Vec representations. Regarding English there was an increase in accuracy of 5.7% to 10.32% and 1.6% to 10.17 % respectively. As it is shown the hybrid vectorization brings better results that lexicon and Word2Vec approaches alone.

Although the proposed methodology does not surpass state of the art approaches in terms of accuracy, it proved to give good consistent results, examined in two dissimilar languages. Furthermore, it keeps the computational cost quite low, which is useful when performance is an issue.

As sentiment analysis constitutes a very important part of the emerging business intelligence, it is just as important to be able to understand the targets of certain sentiments included in some form of text. This field is called aspect extraction and shows increasing popularity. For example, for the following product review ‘’I bought this laptop some months ago. No problem overall, but its keyboard feels really cheap.’’ the company that produces the laptop would find useful to know, except from the sentiment of the public towards a certain product, the specific feature that drives that opinion.

Table 16: Impact of linguistic patterns on the SenEval 2014 dataset [21]

Domain	Classifiers	Recall (%)	Precision (%)	F-score (%)
Laptop	LP	62.39	57.20	59.68
Laptop	CNN	76.31	86.46	81.06
Laptop	CNN+LP	78.35	86.72	82.32
Restaurant	LP	65.41	60.50	62.86
Restaurant	CNN	85.01	87.42	86.20
Restaurant	CNN+LP	86.10	88.27	87.17

Aspect-based opinion mining [21], can be divided into explicit aspects and implicit aspects. The first refer explicitly to the target feature of the opinion and the latter are implied through the context. The previous example has the explicit aspect ‘’keyboard’’ whereas in a review like ‘’I preferred the laptop as it is small and has great autonomy.’’, ‘’small’’ refers implicitly to size and ‘’autonomy’’ to battery life.

In [21], the authors utilize a deep learning approach with convolutional neural networks to tackle the problem of aspect extraction. In order to train the CNN in sequential data they used an algorithm proposed by Collobert et al. [9] which trains the network through back-propagation. The representation used for the words was 300-dimensional embeddings created through a Word2Vec model that was trained on 100-billion-word corpus from Google News. They also created embeddings from a dataset of Amazon product reviews that was designed by McAuley and Leskovec [22]. The dataset has 34,686,770 reviews from 2,441,053 products. The features of each word consisted from its embedding vector and its part of speech tag. The architecture of the network contained the following seven layers. One input layer, two convolution layers, two max-pool layers and a fully connected layer with softmax output [22].

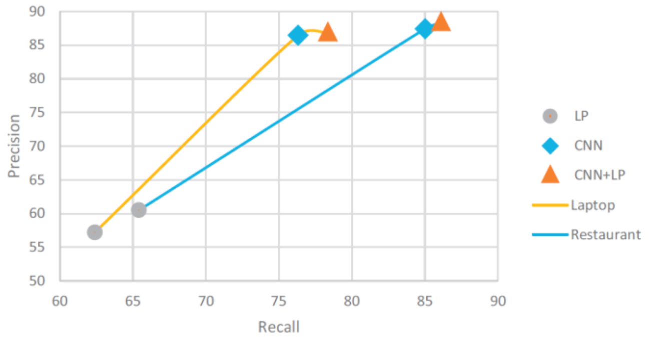

Figure 7: Comparison of the performance of CNN, CNN-LP and LP.

Furthermore, the authors created a set of linguistic patterns (LPs) in order to extract aspects, using language rules and the POS tags of each word. They gathered the final set of aspect terms getting aspect terms classified by the CNN and aspect terms found by the LPs. Finally, they removed any aspect that broke the last of the LPs rules which was not to accept stop-words as aspects.

The datasets used were the aspect-based sentiment analysis dataset created by Qiu et al. [25] and the SemEval 2014 dataset[1] Their particular method outperformed state of the art approaches by 5%–10%.

Table 17: Random features vs. Google embeddings vs. Amazon embeddings on the SemEval 2014 dataset. [21]

Domain	Feature	F-score (%)
Laptop	Random	71.21
Laptop	Google embeddings	77.32
Laptop	Amazon embeddings	80.68
Restaurant	Random	77.05
Restaurant	Google embeddings	83.50
Restaurant	Amazon embeddings	85.70

The results show that Amazon embeddings performed better in contrast to Google embeddings as the latter is more generic, thus, not containing review related vocabulary.

Table 18: Feature analysis for the CNN classifier. [21]

Domain	Features	Recall (%)	Precision (%)	F-score (%)
Laptop	WE	75.20	86.05	80.68
Laptop	WE+POS	76.31	86.46	81.06
Restaurant	WE	84.11	87.35	85.70
Restaurant	WE+POS	85.01	87.42	86.20

Also, word embeddings (WE) and part of speech (POS) features were better choice than WE alone.

Finally, regarding the use of LP and CNN approaches for aspect extraction, it is shown that the ensemble model that they proposed outperforms each one alone.

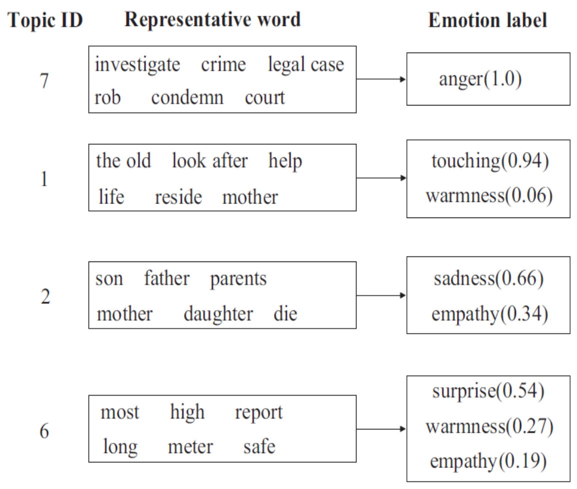
Topic ID
Representative word
Emotion label
7
investigate crime legal case rob condemn court
anger(1.0)
1
the old look after help life reside mother
touching(0.94) warmness(0.06)
2
son father parents mother daughter die
sadness(0.66) empathy(0.34)
6
most high report long meter safe
surprise(0.54) warmness(0.27) empathy(0.19)

Figure 8: Example of social emotion lexicon [23]

The task of emotion mining can be approached with word-level and topic-level models. The word level models consider the individual words to carry the fundamental emotion of the writer. Nevertheless, these methods do not consider that a given word can have many different meanings set in different context. Furthermore, these models are susceptible to word noise, as words that attribute to the overall extracted emotion, may not carry any emotion at all, considering the specific context. As a topic we consider an event an object or a concept that constitutes the target of a certain emotion. Topic-level models show the ability to distinguish different meanings of the same word thus are considered to be an effective alternative to commonly used word-level models.

On [23] the researchers developed two sentiment topic models, the Multi-label Supervised Topic Model (MSTM) and the Sentiment Latent Topic Model (SLTM). The authors used these models to generate lexicons of terms that carry a certain emotion under a certain topic.

In order to test the suggested models, the authors created a dataset of 4570 Chinese news articles. The collected features for each article are URL address, title, publishing date, content and user annotation regarding the following emotions: “amusement”, “anger”, ‘‘empathy’’, ‘‘sadness’’, ‘‘surprise’’, ‘‘touching’’, ‘‘warmness’’. To make sure that the document emotion ratings will not change dramatically, all news articles had been crawled after half a year from their publishing date. During the preprocess phase the title of the article along with its body got intergraded into a single document, and a word segmentation technique followed to the Chinese text. The authors presented results that surpass state of the art approaches in terms of accuracy.

In [24] the authors propose an ensemble model consisting from two statistical models (Naïve Bayes and Maximum Entropy classifiers) and a knowledge-based tool. The authors used such combination of statistical classifiers and knowledge tools, based on the notion that the overall ensemble will benefit from the advantages while avoiding their major disadvantages .The analysis of the text performed on a sentence level in order to enable the system to have a more fine-grained grasp on the multiple different emotional states, that are expressed through the text.

In the preprocess step the text gets broken into sentences and then tokenized in order to be lemmatized or to be removed if is a stop word. Then the representation used is a simple bag-of-words (BoW), where the order of each word in the text is not considered.

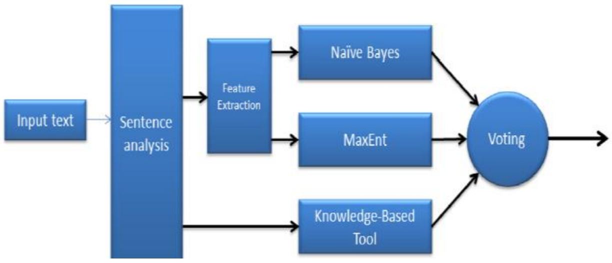

Figure 9: An overview of the ensemble classifier architecture [24]

The combination of the 3 different classifiers is implemented through simple voting where the class to gather most of the classifier votes for each sentence, gets finally selected. This technique is based on the idea that if most of the time the classifiers make independent errors, the majority will still perform the classification right.

In order to evaluate the model, the authors developed a dataset of 750 manually annotated sentences from headlines, twitter posts and news articles from various sources such as CNN, BBC and Euronews. The annotation performed was along the existence and the intensity of the six basic emotions expressed in a scale of 0 to 100 for each one of them. At first the authors assessed the performance of the ensemble model as well of the three classifiers by their one in the classification task of recognizing whether a sentence is emotional or neutral. The results regarding accuracy, precision, sensitivity and specificity were very good and showed that the proposed model indeed achieved better results compared to any of the other by its one. Also, the best results between the three individual classifiers belong to the Naïve Bayes model.

Table 19: Recognizing emotion presence [24]

Metric	Headlines			
	KBtool	N.B.	MaxEnt	Ensemble classifier
Accuracy	0.82	0.87	0.82	0.89
Precision	0.77	0.93	0.89	0.94
Sensitivity	0.92	0.86	0.86	0.9
Specificity	0.66	0.85	0.75	0.9

Metric	Articles			
	KBtool	N.B.	MaxEnt	Ensemble classifier
Accuracy	0.79	0.86	0.82	0.89
Precision	0.72	0.91	0.87	0.93
Sensitivity	0.92	0.9	0.85	0.9
Specificity	0.58	0.81	0.7	0.87

Metric	Tweets			
	KBtool	N.B.	MaxEnt	Ensemble classifier
Accuracy	0.7	0.81	0.77	0.82
Precision	0.68	0.84	0.78	0.85
Sensitivity	0.9	0.87	0.86	0.88
Specificity	0.52	0.67	0.6	0.68

Following that, they assessed the performance of the system regarding the classification between positive and negative polarity on the sentences. Results were similarly good and showed that all models performed better on the structured or emotionally rich texts such as articles and headlines and more poorly on more informal word such as the gathered tweets.

Table 20: Evaluation results of emotional status. [24]

Metric	Headlines			
	KBtool	N.B.	MaxEnt	Ensemble classifier
Accuracy	0.81	0.87	0.84	0.89
Precision	0.85	0.91	0.87	0.90
Sensitivity	0.85	0.84	0.82	0.91
Specificity	0.77	0.90	0.85	0.89

Metric	Articles			
	KBtool	N.B.	MaxEnt	Ensemble classifier
Accuracy	0.8	0.85	0.82	0.87
Precision	0.77	0.89	0.85	0.85
Sensitivity	0.82	0.76	0.80	0.86
Specificity	0.74	0.86	0.85	0.86

Metric	Tweets			
	KBtool	N.B.	MaxEnt	Ensemble classifier
Accuracy	0.71	0.81	0.78	0.83
Precision	0.84	0.88	0.86	0.87
Sensitivity	0.71	0.77	0.87	0.79
Specificity	0.70	0.85	0.77	0.86

5. CONCLUSION

The purpose of this review was to present the trends and methodologies surrounding sentiment and emotion analysis which are amongst some of the most popular aspects of knowledge discovery. Due to the abundance of data generation, opinion mining from microblogging platforms and online review sites is a multi-industry tool that captures the wisdom of the crowds, thus educating decision making, policy creation and marketing strategies.

In our research we examined multiple techniques that can predict the polarity of an opinion or translate it to map onto the primitive emotions, as defined by Ekman or Plutchik. The approaches to the task are mainly divided to lexicon-based methods, machine learning techniques and hybrid methods. Lexicon-based methods analyze the semantic context and sentiment of a phrase by calculating a score or expanding a feature vector based on publicly available lexicons. On the other hand, machine learning models take advantage of the massive amount of data available and train word embeddings that can infer the meaning and structure of a language.

For future work we suggest moving beyond a presentation of the described methods. We recommend a comparative analysis of the examined research using the same dataset on a variation of classifiers.

REFERENCES

[1] J. Clement, "Number of social media users worldwide 2010-2021 | Statista", Statista, 2020. [Online]. Available: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/. [Accessed: 22- Apr- 2020].

[2] J. Blitzer et al., “Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification,” Proc. Assoc. Computational Linguistics, ACL Press, 2007, pp. 440–447.

[3] J. Wertz, "Why Sentiment Analysis Could Be Your Best Kept Marketing Secret", Forbes, 2018. [Online]. Available: https://www.forbes.com/sites/jiawertz/2018/11/30/why-sentiment-analysis-could-be-your-best-kept-marketing-secret/. [Accessed: 10- Apr- 2020].

[4] Y. Dang, Y. Zhang and H. Chen, "A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews," in IEEE Intelligent Systems, vol. 25, no. 4, pp. 46-53, July-Aug. 2010.

[5] D. Chatzakou, V. Koutsonikola, A. Vakali and K. Kafetsios, "Micro-blogging Content Analysis via Emotionally-Driven Clustering," 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, Geneva, 2013, pp. 375-380.

[6] D.Chatzakou, N.Passalis, A.Vakali. MultiSpot: Spotting Sentiments with Semantic Aware Multilevel Cascaded Analysis. Big Data Analytics and Knowledge Discovery (DaWaK), volume 9263, pages 337-350, Springer, 2015

[7] Rout, J.K., Choo, K.R., Dash, A.K. et al. A model for sentiment and emotion analysis of unstructured social media text. Electron Commer Res 18, 181–199 (2018).

[8] Tang, Duyu & Wei, Furu & Yang, Nan & Zhou, Ming & Liu, Ting & Qin, Bing. (2014). Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference. 1. 1555-1565. 10.3115/v1/P14-1146.

[9] Ronan Collobert, Jason Weston, L´eon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12:2493–2537.

[10] Socher, Richard & Pennington, Jeffrey & Huang, Eric & Ng, Andrew & Manning, Christoper. (2011). Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. 151-161.

[11] Kashfia Sailunaz, Reda Alhajj, “Emotion and sentiment analysis from Twitter text”, https://doi.org/10.1016/j.jocs.2019.05.009

[12] Severyn,A, Moschitti, A.(2015,June). Unitn: Training deep convolutional neural network for twitter sentiment classification. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval2015), Association for Computational Linguistics, Denver, Colorado (pp. 464-469)

[13] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint, arXiv: 1301.3781 .

[14] Alex Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. In CS224N Project Report, Stanford.

[15] Maria Giatsoglou, Manolis G. Vozalis, Konstantinos Diamantaras, Athena Vakali, George Sarigiannidis, Konstantinos Ch. Chatzisavvas. 2016. Expert Systems With Applications. Expert Systems With Applications 69(2017) 214-224.

[16] Plutchik, R. (1994). The psychology and biology of emotion (1st). HarperCollins College Publishers

[17] Tsakalidis, A. , Papadopoulos, S. , & Kompatsiaris, I. (2014). An ensemble model for cross-domain polarity classification on twitter. In Proceedings of the Web infor- mation systems engineering –WISE 2014 (pp. 168–177) .

[18] Ekman, P. (1992). An argument for basic emotions. Cognition & Emotion, 6 (3–4), 169–200 .

[19] Agathangelou, P. , Katakis, I. , Kokkoras, F. , & Ntonas, K. (2014). Mining domain-spe- cific dictionaries of opinion words. In Proceedings of the web information systems engineering –WISE 2014 (pp. 47–62) .

[20] Pang, B. , & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the forty-third an- nual meeting on association for computational linguistics (pp. 115–124) .

[21] Soujanya Poria, Erik Cambria, Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. Knowledge-Based Systems 108 (2016) 42-49.

[22] J. McAuley , J. Leskovec , Hidden factors and hidden topics: Understanding rating dimensions with review text, in: Proceedings of RecSys’13. Hong Kong, China, 2013 .

[23] Yanghui Rao, Quing Li, Xudong Mao, Liu Wenyin. 2014. Sentiment topic models for social emotion mining. Information Sciences 266(2014) 90-100.

[24] Isidoros Perikos , Ioannis Hatzilygeroudis. 2016. Recognizing emotions in text using ensemble of classifiers. Engineering Applications of Artificial Intelligence 51 (2016)191–201.