Evaluating the Effectiveness of Linguistic Features for Multi-label Emotion Classification in Reddit Text

Tomisin Adeyemi **Okii Iamvuthipreecha** **Tosin Akinnagbe** **Avé Leone**
New York University
{ota231, ti2060, oma273, aml9182}@nyu.edu

Abstract

Emotional classification in text has many applications in different domains, and social media has become a popular domain for research in this area. In this research paper, we present a study that aims to evaluate the effectiveness of different linguistic feature sets for multilabel emotion classification in Reddit text. We developed lexical, syntactic, and semantic feature sets, then tested the ability of different machine learning models, such as SVM, XGBoost, Random Forests, Logistic Regression, and KNN, to classify emotions based on various combinations of these feature sets. To evaluate the performance of our models, we used several metrics, including micro F1 score, hamming loss, subset accuracy, multilabel accuracy, and a novel metric that we developed, called Balanced Multilabel Performance (BMP) Score. Our findings indicate that the combination of syntactic and lexical features was the most important for emotion classification. Our best-performing models achieved Multilabel Accuracy and Micro F1 scores of 89.03% and 0.5891, respectively. Our study showcases the efficacy of linguistic feature sets and machine learning models for emotion classification in textual data, while emphasizing the need to evaluate model performance using multiple metrics.

1 Introduction

The rapid growth of social media platforms has provided a vast source of textual data which can be used to gain insights into the complexity of human emotions. However, accurately identifying and classifying more granular emotions in unstructured text data remains a challenging task.

A majority of state-of-the-art sentiment analysis tools are limited to binary (positive/negative) sentiment classification. A more nuanced approach would be to identify specific emotions in text. In addition to just identifying one possible emotion from many (multi-output classification), this paper seeks to identify multiple emotions a text can be associated with: this is called multi-label emotion classification. Multi-label classification heavily relies on utilizing different textual features to efficiently capture the emotions of the text. As such, this paper aims to evaluate the significance of different aspects of linguistic features in classifying texts. More specifically, the paper answers the question: **To what extent can lexical, semantic, and syntactic features be used for multi-label emotion classification in Reddit texts?**

To answer this question, we performed an experimental evaluation using Google Researcher's GoEmotions dataset, annotated with multiple emotion labels. We then assess the performance of various feature sets and combinations of them using machine learning multi-label classification models and finally evaluate their effectiveness using a range of example-based and label-based metrics.

2 Related Work

2.1 Emotion Detection in Social Media Text

In recent years, classifying emotion in social media texts has garnered significant attention from researchers. [Pool and Nissim, 2016] delved into Facebook's emotion-based reaction features and classified them into four emotions: anger, joy, sadness, and surprise. To detect emotions, the researchers employed various textual features, such as TF-IDF, word embeddings, n-grams, the WordNet Affect lexicon, and more. Although the best-performing model achieved an F1 Score of 0.469, indicating ample room for improvement, it performed competitively when compared to other similar studies.

In the realm of Reddit, Turcan and McKeown [Turcan and McKeown, 2019] introduced Dreaddit, a corpus annotated for stress levels in Reddit

posts. The researchers gathered data primarily from subreddits centered around topics like abuse, anxiety, and financial instability, amassing a dataset of 187,000 posts, each containing an average of 420 tokens. Using Amazon Mechanical Turk, the comments were annotated as stressed or not stressed, achieving a Fleiss's Kapa agreement of 0.47 among the annotators. Subsequently, the researchers developed supervised models using lexical, syntactic, and social media features to identify stress in comments, resulting in a baseline F-score of 80% for the binary stress classification problem. This work lays the groundwork for future research in this field.

2.2 Multilabel Emotion Detection

In multi-output classification, an instance can be assigned to one of several classes. In contrast, multi-label classification allows an instance to be assigned to one or more classes at the same time. The dataset utilized in this paper is multi-labeled, making it the primary focus of this study.

[Yu et al., 2018] proposed a transfer learning approach to improve the performance of multi-label emotion classification by incorporating sentiment classification. They used a dual attention mechanism to divide the sentence representation into two feature spaces, which capture general sentiment words and emotion-specific words, respectively. They performed experiments on two datasets, resulting in macro-averaged F1 scores of 0.551 and 0.444, and macro-averaged multilabel accuracies of 0.457 & 0.444 on both datasets respectively.

[Zhang et al., 2020] go beyond just textual-based multi-label classification and addresses the issue of multi-label emotion detection in a multi-modal scenario that includes textual, visual, and acoustic modalities. The authors propose a multi-modal sequence-to-set approach to effectively model both the dependence among different labels and the dependence between each predicting label and different modalities in multi-modal multi-label emotion detection. By incorporating text, visual, and audio features, the researchers obtained a multi-label accuracy, hamming loss, and micro F1 score of 0.475, 0.182, and 0.560, respectively. This study is significant because most existing studies on multi-label emotion detection focus on a single modality, such as the textual modality.

2.3 Emotion Detection using Machine Learning

[Chaffar and Inkpen, 2011], adopted a supervised machine learning approach to recognize six basic emotions based on the Ekman model. They tested different feature sets, such as Bag-of-words (BOW), n-grams, and WordNetAffect Lexicon. The SVM classifier performed significantly better than other classifiers, as well on unseen examples, achieving an accuracy of 81.16%

However, different cases of SVM didn't perform as well. [Sailunaz and Alhajj, 2019] tested SVM, random forest and naïve Bayes models to classify emotions based on the Ekman model. The Naive Bayes model outperformed both the Random Forest and SVM model, achieving an accuracy of 47.34% on the emotion classification task.

2.4 Emotion Detection using Deep Learning

[Shrivastava et al., 2019], introduced a new, manually-annotated, corpus which expresses different forms of emotions collected from a TV show's transcript. They used a sequence-based convolutional neural network (CNN) with word embeddings to detect the emotions. An attention mechanism was applied to allow the CNN to focus on words that had more effect on the classification or the part of the features that should be attended to more. They achieved a test accuracy & F-1 score of 80.99 & 0.7248 respectively.

[Chowanda et al., 2021] combined both deep learning and machine learning approaches to explore the effects of different feature sets on emotion classification. 2302 feature sets were explored. The Generalised Linear Model performed the best, with an accuracy score of 0.92 and an F1 score of 0.901.

3 Methodology

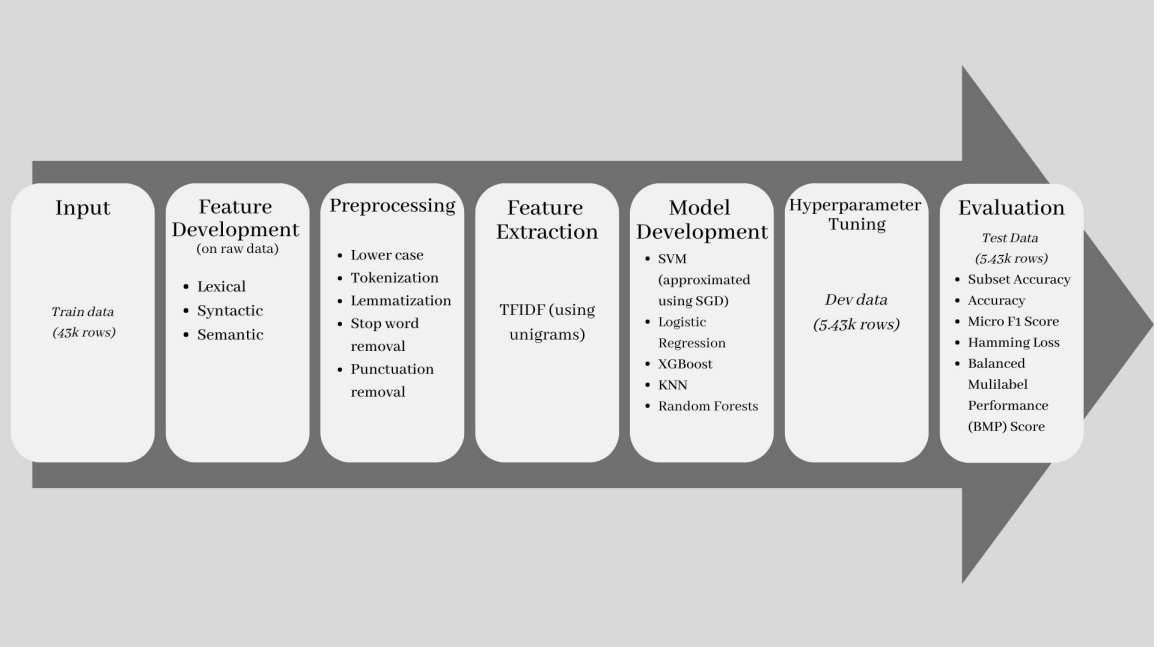
Inspired by [Nandwani and Verma, 2021], Figure 1 outlines the steps taken in our experiment process.

4 Data

4.1 GoEmotions Dataset

For this project, we used the GoEmotions [Demszky et al., 2020] dataset from Google Research, a human-annotated dataset of 58k Reddit comments. The dataset is labeled with 27 emotion categories (plus neutral) and was created to meet the need for a large-scale dataset that covers a wider range of emotions. It is the largest fully annotated English

Figure 1: Experiment process



language fine-grained emotion dataset available, containing 12 positive, 11 negative, 4 ambiguous emotion categories, and 1 "neutral" category. The dataset was built using Reddit comments from 2005 to January 2019, sourced from subreddits with at least 10,000 comments, and excluding non-English and deleted comments. Several other data curation measures were implemented: a lot of offensive language was filtered out, only comments 3 - 30 tokens long were kept, and data was balanced over different subreddit communities.

Demszky et al. consulted psychological literature on emotional expression and recognition in order to create the taxonomy of emotions. The taxonomy was created to meet three objectives [Demszky et al., 2020]: (1) provide comprehensive coverage of emotions expressed in Reddit data, (2) provide coverage of different types of emotional expressions, and (3) limit the number of emotions and their overlap. This allows for fine-grained emotion understanding, even in cases where data may be sparse for certain emotions.

There were 82 unique raters, with three assigned per example, and 2 more added if the 3 raters could not agree on at least 1 emotion label. If raters were not certain about an emotion, they were asked to select Neutral. Out of all of the comments, 54K (around 94% of the total comments) had two or more readers agreeing on at least one label, but only 31% of the comments had three or more raters agreeing on at least one label. Demszky et al. used interrater correlation to determine rater agreement for each correlation.

The study applied Principal Preserved Component Analysis (PPCA) to identify latent dimensions of emotion with high agreement among raters. They found that all 27 PPCs were highly significant, suggesting that the emotions were highly dissociable. The significance of all dimensions is nontrivial, as compared to previous studies that found only a few emotion categories to be significantly dissociable.

4.2 Ekman Model

While the GoEmotions dataset provides a large range of human emotions, the distribution of emotions is very skewed, with the neutral and admiration emotions comprising 42% of the train dataset. Demszky et al. had hierarchies in their taxonomy: the first level being sentiment-based (positive, negative, ambiguous, and Neutral), and the second level based on Paul's Ekman model of emotions [Ekman, 1992]. We decided to map the emotions down to the Ekman level of the hierarchy: it divides the taxonomy into 6 groups + Neutral [1]:

1. Anger (maps to: anger, annoyance, disapproval)
2. Disgust (maps to: disgust)

[1] https://github.com/google-research/google-research/blob/master/goemotions/data/ekman_mapping.json

3. Fear (maps to: fear, nervousness)
4. Joy (maps to: all positive emotions)
5. Sadness (maps to: sadness, disappointment, embarrassment, grief, remorse)
6. Surprise (maps to: all ambiguous emotions)

Table 1 shows some data points from the train set, before and after the original emotions were mapped to the Ekman model:

Text	Emotion	Ekman Emotion
We need more boards and to create a bit more space for [NAME]. Then we'll be good.	desire, optimism	joy
Shit, I guess I accidentally bought a Pay-Per-View boxing match	annoyance, embarrassment	anger, sadness
Maybe that's what happened to the great white at Houston zoo	confusion, realization	surprise
Troll, bro. They know they're saying stupid shit. The motherfucker does nothing but stink up libertarian subs talking shit	anger	anger
My favourite food is anything I didn't have to cook myself.	neutral	neutral

Table 1: Original GoEmotions Labels vs. Ekman Labels

4.3 Splitting the Dataset

The GoEmotions Dataset had already been split into train (80%), development (10%) and test (10%) sets. The number of *labels* per emotion and the number of *data points* with multiple labels is shown in Tables 2 & 3.

Note the distinction between the total number of labels and the total number of data points: in a multi-output classification case, these numbers would be the same, as each data point matches to *1* of many possible outputs. Since this is a multi-*label* case, a single data point can have multiple labels, so the number of labels is higher than the number of data points.

4.4 Development of Linguistic Features

Using the raw, unprocessed data, we designed 3 main feature sets: **Lexical**, **Synctatic** & **Semantic** Features. The lexical features are designed to capture placement of different characters in the text; the syntactic features are designed to capture textual elements related to grammar and parts of speech, and the semantic features are designed to capture the meaning of the text. Table 4 includes a summary of the various linguistic properties used.

	Train	Dev	Test
joy	17410	717	726
neutral	14219	97	123
anger	5579	105	98
surprise	5367	2219	2104
sadness	3263	1766	1787
disgust	793	390	379
fear	726	624	677
Total # Labels	47357	5918	5894

Table 2: The number of labels per emotion in the train, dev & test sets.

# of Labels	Train	Dev	Test
1	39555	4946	4968
2	3763	468	451
3	92	12	8
Total # of Data Points	43410	5426	5427

Table 3: The number of data points labeled with multiple emotions in train, dev & test sets.

4.4.1 Semantic Features

The semantic features for the text are developed using VADER[2] and TextBlob[3].

VADER (Valence Aware Dictionary and sEntiment Reasoner) [Hutto and Gilbert, 2014] is a sentiment analysis tool that has been specifically designed to effectively detect and interpret sentiments expressed in social media. It is a rule-based and lexicon-driven method that utilizes a set of heuristics that take into consideration various linguistic features such as capitalization, punctuation, degree modifiers, contrastive conjuctions, and trigrams, as well as the contextual information in which these features occur. A study conducted by [Hutto and Gilbert, 2014] found that VADER significantly outperforms SVM and Naive Bayes algorithms, achieving an F1 score of 0.96 Twitter data.

[2] https://www.nltk.org/_modules/nltk/sentiment/vader.html

[3] https://textblob.readthedocs.io/en/dev/classifiers.html

Type	#	Features
Lexical	5	• stop words ratio • unique token ratio • punctuation intensity • digit ratio • upper-lower case ratio
Semantic	6	• VADER Scores (positive, negative, neutral, compound) • TextBlob Scores (polarity, subjectivity)
Synctatic	3	• pronoun frequency • noun-verb phrase ratio • tense

Table 4: Summary of Linguistic Features used in Experiments

For the purposes of our experiment, we utilized four key outputs provided by VADER, namely: Positive score (pos), Negative Score (neg), Neutral score (neu), and Compound Score (comp). The Positive score (ranging from 0 to 1) reflects the proportion of positive sentiment words detected, while the Negative score (ranging from 0 to 1) represents the proportion of negative sentiment words detected. The Neutral score (ranging from 0 to 1) represents the proportion of neutral words identified by VADER. The Compound score (ranging from -1 to 1) represents a normalized score that incorporates both the positive, negative, and neutral scores into a single, interpretable score.

TextBlob is a Python (2 and 3) library for processing textual data. In this study, we have leveraged the sentiment property provided by TextBlob[4], which returns both the polarity and subjectivity scores of a given text. The sentiment analysis tool used by TextBlob is based on a lexicon and heuristic approach, which makes use of a dictionary of words along with their associated scores to produce a sentiment score and a subjectivity score for the text. The polarity score is a floating-point value that ranges from -1 (indicating a negative sentiment) to 1 (indicating a positive sentiment). Meanwhile, the subjectivity score is a floating-point value ranging from 0 (indicating an objective text) to 1 (indicating a highly subjective text).

[4] https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.TextBlob.sentiment

4.4.2 Lexical Features

The lexical features for the text are developed manually. Five different properties were developed: Stop Words Ratio, Unique Token Ratio, Punctuation Intensity, Digit Ratio, Upper-Lower Case Ratio.

The stop word ratio is a measure of the proportion of stop words to tokens in a text, and can provide insight into the extent to which a text is composed of non-informative words. The unique token ratio is the proportion of tokens that occur only once in a text to the total number of tokens in the text, which can be an indicator of the lexical diversity of a text. A text with a higher unique token ratio is likely to have a broader range of vocabulary and may be considered more complex or sophisticated. The digit ratio captures the presence of numerical data in a text, calculated as the ratio of digit-containing tokens to the total number of tokens in a text.

The Upper/Lower case ratio is a measure of the number of uppercase characters compared to lowercase characters in a text and can provide information on the writing style and tone of the text. For instance, a text with a higher ratio of uppercase characters may indicate emphasis or shouting, while a text with a higher ratio of lowercase characters may suggest a more casual or informal tone. Finally, the punctuation intensity property describes the average length of each instance of punctuation in a sentence and can be indicative of the level of intensity or emotion conveyed by the text. Given that the text data in this study is from social media, a higher punctuation intensity may indicate the presence of more aggressive or intense emotions. The Punctuation Intensity, PI, of a sentence, s, is calculated as follows:

$$PI(s) = \frac{\sum_{i=1}^{n} len(i)}{n}, \text{ where}$$
$$i = \text{contiguous instance of punctuation}$$
$$n = \text{total number of punctuation instances}$$

4.4.3 Synctatic Features

This study utilized the spaCy Python package [Honnibal and Montani, 2017] to develop syntactic features. Specifically, tenses and parts-of-speech (POS) tags are determined through rule-based algorithms that identify patterns within the text. The calculation of the noun-verb phrase ratio is based

on syntactic dependency relations extracted from spaCy, which employs the Universal Dependencies Version Guidelines [Nivre et al., 2017]. This standardized set of syntactic dependency labels is used to annotate the structural organization of natural language text.

To develop the noun-verb phrase ratio, tokens that are assigned dependencies labeled as nsubj: nominal subject, nsubjpass: passive nominal subject, attr: attribute, dobj: direct object, prep: prepositional modifier, pobj: object of preposition, conj: conjunction, or appos: appositional modifier are considered noun phrases. On the other hand, tokens labeled as verbs, excluding auxiliary verbs, are considered verb phrases.

It is worth noting that rule-based algorithms have been widely adopted in natural language processing (NLP) research [Sidorov et al., 2013] and have shown to be more efficient than complex deep learning techniques in certain cases, such as tense classification. Therefore, rules that describe tenses are employed to check for tense patterns using POS tags. In total, the rule-based algorithm identifies 13 different tenses and one unknown class. For the purpose of machine learning, dummy encoding is used to represent the categorical tense classification as binary values, while the unknown class is excluded to avoid over-determination of the model.

4.5 Feature Extraction

4.5.1 Text Preprocessing

To preprocess and clean the textual data, several pre-processing techniques were applied. Preprocessing is critical stage in data preparation as the data quality significantly impacts many approaches that follow pre-processing [Nandwani and Verma, 2021]. For this project, text pre-processing was done in the following order: first the words were converted to lowercase, then tokenized [5], then lemmatized, then stop words were removed, and finally punctuation was removed.

4.5.2 TFIDF

To convert the textual features into a numerical form, we used the Term Frequency-Inverse Document Frequency (TFIDF) technique, which has been shown to outperform other feature extraction methods like Bag-of-Words (BOW) or n-grams, according to [Nandwani and Verma, 2021].

[5]Tokenization is done using NLTK's tokenizer https://www.nltk.org/api/nltk.tokenize.html

We decided against using Word Embeddings like Word2Vec for feature extraction since they are more commonly used with deep learning models. Our TFIDF calculation for a term t in a document d in the dataset is as follows:

$$tfidf(t,d) = tf(t,d) * idf(t), \text{ where}$$
$$tf(t,d) = 1 + \log(tf) \text{ and}$$
$$idf(t) = \log[\tfrac{1+n}{1+df(t)}] + 1$$
with n = total # of documents in the dataset
$df(t)$ the document frequency of term t,
as explained by [Pedregosa et al., 2011]

We used sublinear tf scaling to balance the weight of terms as the tf increases, and we smoothed the IDF values by adding 1 to both the numerator and the denominator of IDF to prevent zero division on out-of-vocabulary terms. Specifically, we used the TFIDF of unigrams in the dataset, as bigrams and trigrams required unnecessary computational power. We constructed the TFIDF matrix and vocabulary using the training set, and then applied it to the development and test sets to prevent data leakage and mitigate overfitting.

5 Model Development

5.1 Description of Machine Learning Models

Five different classification-based machine learning algorithms were used for our experiments: Support Vector Machines (SVM), Logistic Regression, K Nearest Neighbors (KNN), Random Forests & XGBoost.

SVM is a maximal margin classifier: it makes classifications by finding a hyperplane that provides maximum separation between classes, using a Hinge Loss Function. We approximated SVM using Stochastic Gradient Decent with hinge loss to save computational time. Logistic regression makes predictions by deriving the probability of a particular outcome given a particular threshold. The KNN classifier assigns data points to classes based on closeness to other points. Random Forests make predictions by aggregating multiple decision trees. Finally, the XGBoost classifier is based on a gradient boosting technique similar to random forests: it aggregates multiple decision trees and usually outperforms Random Forests [Hastie et al., 2001].

5.2 Handling Label Imbalance, Multi-labels, & Hyperparameter Tuning

The distribution of labels in the dataset is very imbalanced, as shown in table 2. To account for this

in our models, we added weights inversely proportional to the frequency of a label. Thus, misclassifications for rarer labels were more heavily penalized.

While KNN, Random Forests, & XGBoost handle multi-labeled data natively, Logistic Regression & SVM do not. Thus, we used a transformation based classifier called a Classifier Chain [Pedregosa et al., 2011]. It is an ensemble of k binary classifiers, where k is the number of labels; the output of each classifier is chained to the input of the next classifier, taking label dependencies into account [Herrera et al., 2016b].

Finally, to prevent overfitting, hyperparameter tuning for each model was done on the dev set, and ran only once on the test set.

5.3 Implementation of Baseline & Test Models

A series of five tests were conducted to evaluate the effectiveness of incorporating lexical, semantic, and syntactic features in the classification of emotions. In the first test, only the TFIDF matrix was utilized as a predictor. The subsequent tests involved incorporating all of the features, including the TFIDF matrix, and then excluding the individual sets of syntactic, semantic, and lexical features while maintaining the TFIDF matrix.

5.3.1 Combining (Sparse) TFIDF w/ (Dense) Linguistic Features

The sparsity of the TFIDF matrix posed a challenge when combining it with dense linguistic features. To address this, we used the predicted probabilities from the best-performing baseline model (Random Forests for both the development and test sets), to represent the TFIDF model as dense input. We concatenated these probabilities with the relevant linguistic features being tested. While there may be more effective approaches to combining sparse and dense features, we opted for this method due to time constraints. Further research could explore alternative methods for concatenating sparse and dense input.

5.3.2 Dropping of Linguistic Features

To assess the importance of Syntactic, Lexical & Semantic features in our experiments, we excluded the predictors associated with each feature to determine whether their absence had any noticeable impact on performance. Another approach could have been to use each feature as the sole input to the model. We decided not to go this route to allow the models assess the relative importance of each feature in the context of the other features being tested. For instance, a specific syntactic feature might not be strongly predictive of a particular emotion label, but when combined with certain lexical features, it could contribute to improving the overall performance of the model.

6 Evaluation

The output of each model is a set of labels predicted for each data point i. In traditional multi-output classification, predictions are either correct or incorrect. However, because this is a multi-label classification task - each data point i can have multiple labels - predicted instances of our models can be fully correct, partially correct, or completely incorrect. Herrera et. al [Herrera et al., 2016a], describe multiple ways to evaluate multi-label classifiers. There are two main classes of metrics: Example-based metrics & Label-based metrics.

For example-based metrics, we selected Hamming Loss, Subset Accuracy, & Multi-label Accuracy. For label-based metrics, we selected the Micro F1 score. For ease of comparison, we then combined these metrics to create a new bespoke metric called Balanced Multi-label Performance (BMP) Score, which captures the overall quality of the classification model on the multi-label classification problem.

6.1 Example-based metrics

Example-based metrics evaluate the similarity between the predicted labels and the true labels separately for each instance, then average the performance scores across all instances by dividing by the number of samples.

6.1.1 Hamming Loss

One of the most commonly used metrics in multi-label classification, the hamming loss describes the fraction of labels that are incorrectly predicted [Pedregosa et al., 2011]. It is computed as follows:

$$HL = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}|Y_i \Delta Z_i|$$

Where Y_i refers to the ith instance of the actual label-set, Z_i, refers to the ith instance of the predicted label-set, k refers to the number of labels, and n the total number of data points. Δ measures the symmetric difference between Y_i and Z_i. In

essence, Hamming Loss measures the error made by the classifier as a proportion of the length of the label set, in this case, we have 7 labels.

6.1.2 Subset Accuracy

Subset accuracy is a strict evaluation metric that checks for an exact match between the real label set and the predicted label set. The larger the label set, the lower the likelihood that the classifier produces exactly the correct output [Herrera et al., 2016a]; thus with 7 labels, a relatively low subset accuracy score is expected. It is calculated as follows:

$$SubsAcc = \frac{1}{n}\sum_{i=1}^{n}[\![Y_i = Z_i]\!]$$

6.1.3 (Multilabel) Accuracy

Defined as just *Accuracy* in the multi-label field, it is the proportion between the number of correctly predicted labels and the total number of active labels, in the both real label set and the predicted one [Herrera et al., 2016a]. It is averaged over all the data points, like the other example-based metrics. It is calculated as follows:

$$Acc = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$

6.2 Label-based metrics

Label-based metrics are computed independently for each label, then averaged using either micro- or macro- averaging, hence they focus on the performance of the classifier for each label. In the context of this paper, label-based metrics captures the classifier model's performance across different emotions, adding granularity to our evaluation.

6.2.1 Micro F1 Score

With micro-averaging, metrics (true positives, false positives, & false negatives) for all labels are aggregated, then the metric itself is computed only once [Herrera et al., 2016a]. Thus, the Micro F1 Score calculates metrics globally by counting the total number of true positives, false negatives and false positives across all labels [Pedregosa et al., 2011]. Given $j = (1, \cdots k)$ labels, the Micro F1 is calculated as follows:

$$MiF_1 = \frac{2PR}{P+R}, \text{ where:}$$

$$P = \frac{\sum_{j=1}^{k} TP_j}{\sum_{j=1}^{k}(TP_j+FP_j)} \text{ and } R = \frac{\sum_{j=1}^{k} TP_j}{\sum_{j=1}^{k}(TP_j+FN_j)}$$

[Zhang et al., 2015]

In the macro case, Precision & Recall are done for each label separately then averaged over the number of labels, giving each label equal weights. Thus, because of the uneven distribution of labels in our dataset, we opted for MicroF1 as the predictions where rare labels appear are combined with those where frequent ones appear; the contribution of each label to the final measure is not the same [Herrera et al., 2016a]. Hence, this metric helps determine how well a classifier does in terms of class imbalance.

6.3 Summary Metrics

We decided to combine both our example-based and label-based metrics to create a single summary metric that encapsulates a classifier's performance.

6.3.1 Balanced Multilabel Performance (BMP)

With the hamming loss metric, lower scores indicate better performance, but with the Micro F1, Subset Accuracy & Accuracy, higher scores indicate better performance. Thus, the first step at combining these metrics was finding the inverse of the hamming loss so a higher score indicates a better performance. After normalizing, assigning the hamming loss half the weight, and the other metrics the other half, the Balanced Multilabel Performance (BMP) metric is calculated as follows:

$$BMP = (\tfrac{1}{2} * \tfrac{1}{10*HL}) + (\tfrac{1}{2} * (\textstyle\prod_{i=1}^{n} x_i)^{\frac{1}{n}}),$$
where $x \in \{\text{SUBSACC, ACC, MIF1}\}$

As shown in the formula, the metric comprises of two things: the inverse of the hamming loss and the geometric mean of the other metrics. The geometric mean is used instead of the regular arithmetic mean to favor models that perform well across all metrics, as opposed to favoring a model that performs extremely well on one metric but very poorly on others.

7 Results

Table 5 shows the results for all experiments conducted.

7.1 Comparison of Baseline & All Features Model

The results indicate that the model incorporating all features with TFIDF yielded a higher arithmetic mean for the BMP metric than the baseline model;

however, the baseline model outperformed the comprehensive feature model in terms of the geometric mean. Thus, the baseline model exhibited superior performance across all tests. Notably, the model incorporating all features demonstrated a more pronounced performance spike. This can be attributed to the geometric mean treating each model with equal weight, whereas the regular mean does not. Furthermore, neither model displayed the best BMP performances across all tests, indicating that only select features (specifically, lexical and syntactic features, as will be demonstrated later) are necessary to enhance model performance.

7.2 Linguistic Feature Analysis

Incorporating all features led to the SVM model performing the worst, which is not surprising given that SVM is based on the concept of separability and may struggle with complex data. On the other hand, XGBoost performed the best across all tests, which is not unexpected since it is known for its robustness in handling complex data and has been a popular choice in various classification tasks. It is important to note that the performance of each algorithm was influenced by the particular combination of features used as input. Therefore, further experimentation with different combinations of features is used for a more nuanced analysis, as shown in the next couple of paragraphs.

Interestingly, dropping the semantic features resulted in the best BMP GeoMean performance across all tests, with a close tie to the baseline. This finding indicates that semantic features were actually the **least** helpful for the emotion classification task, which may come as a surprise since positive, negative, neutral, and compound sentiments are often associated with emotions. It is possible that collinearity among the semantic features may have contributed to this outcome. Additionally, these results suggest that **lexical and syntactic features work together best for this emotion classification task**.

When the lexical features were dropped, the BMP performance for both the GeoMean and arithmetic mean decreased, indicating that the lexical features were the **most** helpful in determining emotions. This shows that the use of properties such as the upper/lower case ratio, punctuation intensity, and others aided the multi-label emotion classification task. On the other hand, dropping the syntactic features resulted in the **best** BMP Mean across all tests, indicating that the syntactic features played a lesser role in the emotion classification task. One possible interpretation of this result is that syntactic features might not be as important as lexical features in conveying emotions since emotions are often expressed through the use of specific words rather than grammatical structures.

7.3 Model-based Analysis

Logistic Regression and Support Vector Machines (SVM) yielded poor performances across all tests except for the Baseline model. In Section 5.2, it was noted that both models were adapted to the multilabel classification task using classifier chains. However, classifier chains are known to produce variable results depending on the ordering of the labels [Herrera et al., 2016b], which may have impacted the performance of these models.

On the other hand, K-nearest neighbors (KNN) had the lowest performance among all the baseline models; one possible explanation for this could be the sensitivity of KNN to high-dimensional data, which is known to adversely affect its performance. Random Forests exhibited generally good performance on metrics across all tests and performed the best on the Baseline model; the ensemble nature of Random Forests contributed to their success, as they can reduce overfitting and provide robust classification. Incorporating features into the models led to XGBoost exhibiting the best performance, which, as stated in previous paragraphs, is not surprising given its ability to deal with complex data.

8 Comparison To Previous Results

To compare our work to previous results, we compare our best Micro F1, (Multilabel) Accuracy & Hamming Loss scores to those of other papers that did experiments on the GoEmotions Dataset. Our best accuracy, micro F1 & hamming loss scores are 89.03%, 0.5891 & 0.1097 respectively. Our micro F1 score beats the baseline of 0.51 achieved by the BERT model trained in the GoEmotions paper [Demszky et al., 2020].

[Maheshwari and Varma, 2022] used Deep Learning models for predictions on the Goemotions dataset and achieved accuracy & micro F1 scores of 0.661. Whilst they achieved a higher micro F1 score, our study achieved better multilabel accuracy.

[Huang et al., 2021] designed a new Deep Learning framework for multilabel classification. They

Baseline (Just TFIDF)

	Acc	MiF1	SubsAcc	HL	BMP
Logistic Regression	0.8664	**0.5758**	**0.5067**	0.1336	0.6904
SVM	0.8646	0.5696	0.4979	0.1354	0.6824
KNN	**0.8572**	**0.4455**	0.3495	**0.1428**	**0.6056**
Random Forests	**0.8860**	0.5415	0.4266	**0.1140**	**0.7333**
XGBoost	0.8855	0.4968	**0.3484**	0.1145	0.7044
				BMP GeoMean	0.6818
				BMP Mean	0.6832

TFIDF + All Features

	Acc	MiF1	SubsAcc	HL	BMP
Logistic Regression	0.8500	0.5443	0.4551	0.1500	0.6309
SVM	**0.8292**	**0.4883**	**0.3973**	**0.1708**	**0.5646**
KNN	0.8767	0.5388	0.4452	0.1233	0.7029
Random Forests	0.8895	0.5549	0.4397	0.1105	0.7528
XGBoost	**0.8899**	**0.5891***	**0.4837***	**0.1101**	**0.7704**
				BMP GeoMean	0.6798
				BMP Mean	0.6843

TFIDF + Semantic Dropped

	Acc	MiF1	SubsAcc	HL	BMP
Logistic Regression	**0.8413**	**0.5141**	0.4325	**0.1587**	**0.6011**
SVM	0.8583	0.5481	**0.4809**	0.1417	0.6576
KNN	0.8748	0.5305	0.4373	0.1252	0.6931
Random Forests	**0.8860**	0.5361	**0.4172**	**0.1140**	0.7301
XGBoost	0.8848	**0.5589**	0.4502	0.1152	**0.7370**
				BMP GeoMean	**0.6819**
				BMP Mean	0.6838

TFIDF + Lexical Dropped

	Acc	MiF1	SubsAcc	HL	BMP
Logistic Regression	0.8500	0.5433	0.4572	0.1500	0.6311
SVM	**0.8127**	**0.4447**	**0.3567**	**0.1873**	**0.5196**
KNN	0.8781	0.5487	0.4605	0.1219	0.7128
Random Forests	**0.8903***	0.5637	0.4526	**0.1097***	0.7610
XGBoost	0.8889	**0.5832**	**0.4800**	0.1111	**0.7647**
				BMP GeoMean	**0.6710**
				BMP Mean	**0.6779**

TFIDF + Syntactic Dropped

	Acc	MiF1	SubsAcc	HL	BMP
Logistic Regression	0.8510	0.5458	0.4592	0.1490	0.6342
SVM	**0.8261**	**0.4791**	**0.3895**	**0.1739**	**0.5556**
KNN	0.8778	0.5480	0.4557	0.1222	0.7107
Random Forests	**0.8903***	0.5613	0.4476	**0.1097***	0.7592
XGBoost	0.8892	**0.5841**	**0.4791**	0.1108	**0.7658**
				BMP GeoMean	0.6802
				BMP Mean	**0.6851**

Table 5: Results for Baseline Model, Model with All Features Included, and Model with Semantic, Lexical & Synctatic Features Dropped. A green asterisk, *, indicates the best performance across a specific metric (excluding the BMP metric)

achieved Micro F1 & Hamming Loss scores of 0.5957 & 0.0302 respectively on the GoEmotions dataset, beating our study's results.

Finally, [Alvarez-Gonzalez et al., 2021] experimented with different feature extraction & Machine Learning methods. We will be comparing our results with their TF-IDF & Random Forests/Logistic Regression models. They achieved micro-F1 scores of 0.53 & 0.52 on their logistic regression & random forest models respectively. Our baseline Logistic Regression & Random Forest models beat their results, achieving F-1 scores of 0.5758 & 0.5415 respectively.

9 Future Work & Concerns

This research has contributed significantly to our understanding of multi-label emotion classification. However, there is still ample room for future studies to expand upon the findings. As noted in the Related Works section, Deep Learning models have been used for multi-label classification tasks. Hence, future studies can focus on comparing the performance of different deep learning models for the same task. Additionally, the distribution of emotions in the dataset could be balanced by incorporating random oversampling or undersampling methods, this could possibly lead to higher Micro F1 scores. Since no libraries currently contain implementations of these techniques in a multi-label setting, manual implementation may be necessary.

Furthermore, it is imperative to examine the labels provided in the dataset to ensure we agree with their annotations. One possible approach would be to annotate a subset of the dataset and compare these annotations to the original labelset. To investigate more the features that aid the classification of each emotion specifically, separate models can be built to test for different emotions. Additionally, further experimentation with additional feature sets is also worth considering. As highlighted in Section 5.3, alternative ways to combine dense and sparse features can also be explored.

Lastly, in order to further improve the process of calculating the Balanced Multi-label Performance (BMP), a modification can be made to how the hamming loss is inverted. Rather than inverting the hamming loss and multiplying it by 10 to normalize it, a better approach would be to subtract the hamming loss from 1. This method would result in a more stable and reliable BMP metric, particularly when dealing with extreme values.

$$BMP = (\tfrac{1}{2} * (1 - HL) + (\tfrac{1}{2} * (\textstyle\prod_{i=1}^{n} x_i)^{\frac{1}{n}}),$$
where $x \in \{\text{SUBSACC}, \text{ACC}, \text{MIF1}\}$

By implementing this revised approach, the BMP would be better suited to analyzing normalized metrics and producing more consistent and accurate results.

10 Conclusion

In this paper, we have explored the extent to which lexical, semantic, and syntactic features can be utilized for multi-label emotion classification in Reddit texts. Synctatic, Lexical & Semantic Features were concatenated with TFIDF features and tested against different models. Our results show that semantic features help the least, and lexical features help the most, with the combination of Synctatic & lexical features achieving promising results.

In terms of broader, more practical applications, our study contributes to existing research aiming for a better understanding of how emotions are expressed and classified in social media. The results of this research can lead to improved social media monitoring, more targeted marketing strategies, and aid mental health professionals in helping their patients.

References

Nurudin Alvarez-Gonzalez, Andreas Kaltenbrunner, and Vicenç Gómez. Uncovering the limits of text-based emotion detection. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2560–2583, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.219. URL https://aclanthology.org/2021.findings-emnlp.219.

Soumaya Chaffar and Diana Inkpen. Using a heterogeneous dataset for emotion analysis in text. In Cory Butz and Pawan Lingras, editors, *Advances in Artificial Intelligence*, pages 62–67, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21043-3.

Andry Chowanda, Rhio Sutoyo, Meiliana, and Sansiri Tanachutiwat. Exploring text-based emotions recognition machine learning techniques on social media conversation. In *Procedia Computer Science*, pages 821–828, Online, 2021. Elsevier. doi: https://doi.org/10.1016/j.procs.2021.01.099bv. URL https://www.sciencedirect.com/science/article/pii/S1877050921001320?via%3Dihub.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.372. URL https://aclanthology.org/2020.acl-main.372.

Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200, 1992. doi: 10.1080/02699939208411068. URL https://doi.org/10.1080/02699939208411068.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Francisco Herrera, Francisco Charte, Antonio J. Rivera, and María J. del Jesus. *Case Studies and Metrics*, pages 33–63. Springer International Publishing, Cham, 2016a. ISBN 978-3-319-41111-8. doi: 10.1007/978-3-319-41111-8_3. URL https://doi.org/10.1007/978-3-319-41111-8_3.

Francisco Herrera, Francisco Charte, Antonio J. Rivera, and María J. del Jesus. *Transformation-Based Classifiers*, pages 65–79. Springer International Publishing, Cham, 2016b. ISBN 978-3-319-41111-8. doi: 10.1007/978-3-319-41111-8_4. URL https://doi.org/10.1007/978-3-319-41111-8_4.

Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

Chenyang Huang, Amine Trabelsi, Xuebin Qin, Nawshad Farruque, Lili Mou, and Osmar Zaïane. Seq2Emo: A sequence to multi-label emotion classification model. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4717–4724, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.375. URL https://aclanthology.org/2021.naacl-main.375.

C. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225, May 2014. doi: 10.1609/icwsm.v8i1.14550. URL https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

Himanshu Maheshwari and Vasudeva Varma. An ensemble approach to detect emotions at an essay level. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 276–279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.wassa-1.30. URL https://aclanthology.org/2022.wassa-1.30.

Pansy Nandwani and Rupali Verma. A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, 11, 2021.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-5001.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Chris Pool and Malvina Nissim. Distant supervision for emotion detection using Facebook reactions. In *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES)*, pages 30–39, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/W16-4304.

Kashfia Sailunaz and Reda Alhajj. Emotion and sentiment analysis from twitter text. *Journal of Computational Science*, 36:

101003, 2019. ISSN 1877-7503. doi: https://doi.org/10.1016/j.jocs.2019.05.009. URL https://www.sciencedirect.com/science/article/pii/S1877750318311037.

Kush Shrivastava, Shishir Kumar, and Deepak Kumar Jain. An effective approach for emotion detection in multimedia text data using sequence based convolutional neural network. In *Multimed Tools Appl 78*, page 29607–29639, Online, july 2019. Springer Nature. doi: https://doi.org/10.1007/s11042-019-07813-9. URL https://link.springer.com/article/10.1007/s11042-019-07813-9#citeas.

Grigori Sidorov, Anubhav Gupta, Martin Tozer, Dolors Catala, Angels Catena, and Sandrine Fuentes. Rule-based system for automatic grammar correction using syntactic n-grams for English language learning (L2). In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 96–101, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/W13-3613.

Elsbeth Turcan and Kathy McKeown. Dreaddit: A Reddit dataset for stress analysis in social media. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 97–107, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6213. URL https://aclanthology.org/D19-6213.

Jianfei Yu, Luís Marujo, Jing Jiang, Pradeep Karuturi, and William Brendel. Improving multi-label emotion classification via sentiment classification with dual attention transfer network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1097–1102, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1137. URL https://aclanthology.org/D18-1137.

Dell Zhang, Jun Wang, and Xiaoxue Zhao. Estimating the uncertainty of average f1 scores. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, page 317–320, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450338332. doi: 10.1145/2808194.2809488. URL https://doi.org/10.1145/2808194.2809488.

Dong Zhang, Xincheng Ju, Junhui Li, Shoushan Li, Qiaoming Zhu, and Guodong Zhou. Multi-modal multi-label emotion detection with modality and label dependence. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3584–3593, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.291. URL https://aclanthology.org/2020.emnlp-main.291.