Using Sentiment Multi-label Analysis for MARVEL Character Review

Esma Galijatovic
Technische Universität Graz
Graz, Austria
egalijatovic@student.tugraz.at

ABSTRACT

Sentiment analysis is one of the most important parts of Natural Language Processing (NLP) research. It is mainly used to understand social media contents. Apart from information, text almost always contains emotional content. Emotional part of text can not always be classified as one category but it can often belong to several classes. This project describes a model that predicts whether movie text line belongs to one or more emotional classes. After model is trained over one data-set [5] of movie lines, it is used for character analysis of other data-set [6] - MARVEL movie lines. This part includes exploring what emotions characters encounter through a movie. For character analysis dataset of MARVEL movie lines is used, where most important characters are analysed. This model uses features derived from word and char n-grams, parts-of-speech, word embedding and Opinion Lexicon. In this document information about data-sets, methods used and results is provided.

KEYWORDS

Sentiment analysis, Support Vector Machine, Multi-label Classification

ACM Reference Format:
Esma Galijatovic. 2021. Using Sentiment Multi-label Analysis for MARVEL Character Review. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 RELATED WORK

Sentiment analysis, as one main areas of NLP was actively researched, especially with the growth of social media. In [1] we can see emotion classification very similar to one in this project. Here presence or absence of an emotion is predicted, i.e. if there is anger or no anger in a text. Beside pure sentiment analysis, some also researched intensity of emotions in text [2], where we can see similar feature selection process. In [3] Alm et al. explored automatic classification of sentences in children's fairy tales according to the basic emotions. In this project we focus on multi-label emotion classification, and transfer learning approach to the same problem is shown in [4]. In [4] they extend the Aspect Based Sentiment Analysis (ABSA) methods with multi-label classification capabilities.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 INTRODUCTION

Understanding implicit or explicit sentiments expressed in texts is very common and indeed a complex problem. As we saw in section 2, there is a lot of previous researches that covered similar problems. In this document here we will focus on sentiment/emotion analysis, but we will also use it to analyse characters based on the emotions from their dialogues. Section 3 describes data that was used, namely two datasets that were used: annotated emotion dataset from movie lines, and Marvel Universe movie lines. First one was used for training the model, and the second one is used for character analysis. In section 4 preprocessing techniques of dialogue lines are described, along with explanations why are these techniques important. Section 5 describes feature extraction. Here various techniques were used to improve the quality of predictions.

2.1 Multi-label classification

Multi-label classification emerged from the research of text categorisation problem, where each document may belong to several classes simultaneously, for example predefined topics or in our case emotions. This is an important problem. Examples range from news articles, emails to reviews. For instance, this can be employed to find the genres that a movie belongs to, based on the summary of its plot or to find emotions in movie reviews/comments. In multi-label classification training set consists of instances with associated set of labels, where the goal of a model is to predict all labels correctly.

3 DATA

In this section used data will be described. There are two data-sets used for this project. First is training dataset [7] that consists of movies subtitles, at it is used to create a model which will later be used for prediction of emotions in target dataset [6].

3.1 XED dataset

XED dataset consists of emotion annotated movie subtitles from OPUS, which is a new collection of translated movie subtitles. Authors used Plutchik's 8 core emotions to annotate the data. In this project we only focused on English language subtitles, but there are also annotates movie subtitles in 42 other languages. This English subtitle dataset consists of 17528 lines and eight emotions/categories:

(1) anger
(2) anticipation
(3) disgust

(4) fear
(5) joy
(6) sadness
(7) surprise
(8) trust

Since labels of the data were in following format:

$$sentence1\backslash t\ label1, label2$$

where label1, label2 are numbers corresponding to emotions as previously written, we had to implement one hot encoding and change the format of the data.

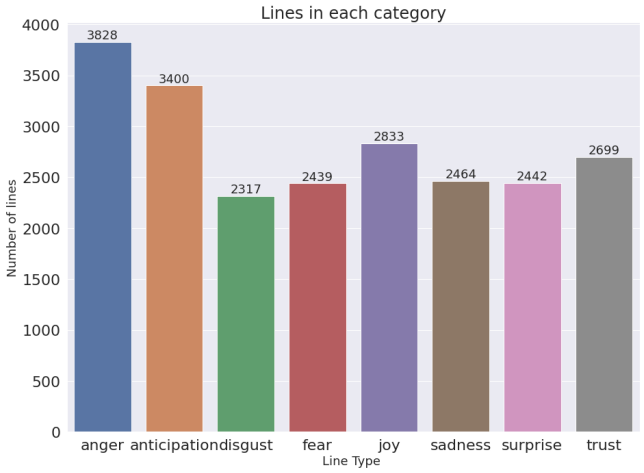

Figure 1: Number of lines in each category

In figure 1 we see distribution of lines through categories. From there we see that classes are almost evenly distributed, except from anger and anticipation that have more instances.

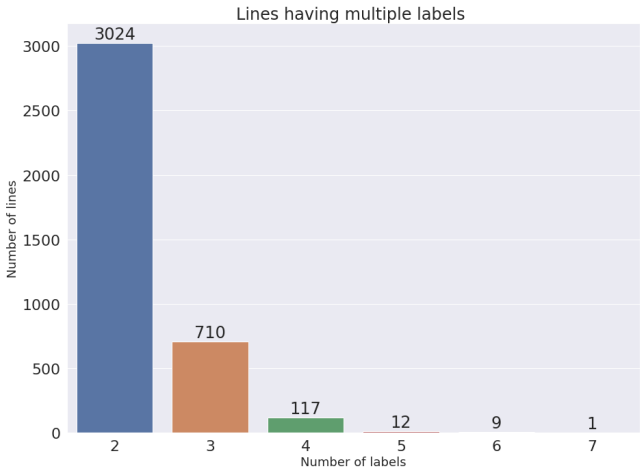

Figure 2: Number of lines that belong to more than one category

As previously mentioned, this lines are multi-labeled which means that every line may belong to more classes, or in other words lines may have several emotions. In 2 we notice that most of the lines that are multi-labeled have two labels, and rarely three or four. Apart from this, there are 13655 lines that only have one label, which is a majority. Average number of words per line in this dataset is 8.99.

3.2 Marvel Universe dataset

This dataset is created from the transcripts of Marvel Universe movies. Transcripts were taken from Fandom's Transcripts Wiki [8]. Creators of the dataset preprocessed in in a way that every line is associated with character name, movie name and a word count. When the predictions are done we can use this data to analyse which emotion prevails in which character lines, and does this also depend on movie the line was said in.

This dataset consists of 652 characters from 11 Marvel movies. For this project, lines from 15 most important characters were extracted, as we can see in figure 3. Importance of characters is based on number of lines they have throughout all movies.

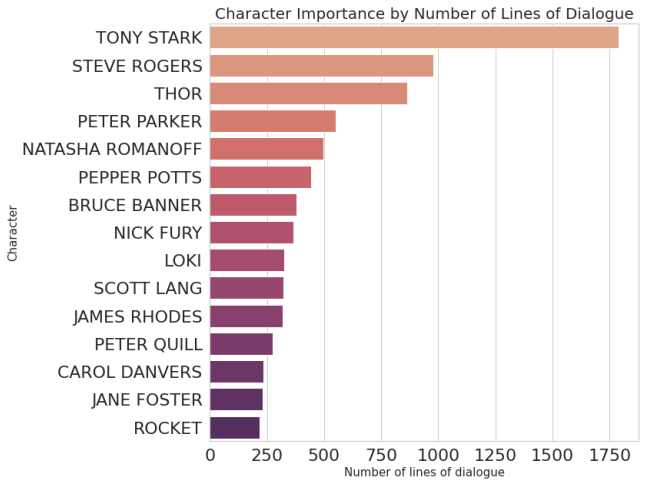

Figure 3: Important characters based on number of lines

After extraction, this dataset consists of 7793 lines, with average of 10.65 words per line, which is slightly higher than in training dataset. On figure 4 we see how are lines distributed through movies. There we also see that there is no big difference between the number of lines in movies. In figure 3 we see that Tony Stark is most important character, which can be confirmed from movies listed since most of the movies has him as a main actor. This character will be used for main analysis of his emotions through movies and in general.

4 PREPROCESSING

In this section, all used preprocessing techniques will be described. As we know, preprocessing is very important part of classification process, especially in NLP. The following techniques were used to get the best results:

- Lemmatisation is process were a word is transformed into its dictionary or base form. This way different forms of a same word are treated the same - this gives better results in

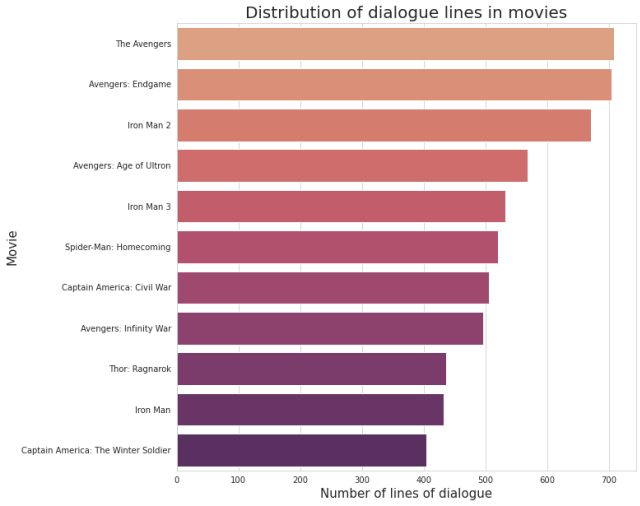

Figure 4: Distribution of lines through movies

NLP. One example is word 'good' and its comparative form 'better'. After lemmatisation, both words will be in base form, which is 'good'. This process takes a lot of time, and is very consuming because it uses POS tags for transformation. Sometimes, instead of lemmatisation, stemming is used, but it is only capable of removing stems or endings of a word, but not transforming it into its base form in cases like above. This technique was also tested but it did not yield in good results. Unfortunately, unlike stemming lemmatisation is very time consuming which is not always acceptable. For MARVEL dataset with only fifteen most important characters took about 20hours for processing to finish (this subset has 7793 lines which summed gives 83015 words in total). In this project *spacy* library for lemmatisation was used.

- Stop-word removal was a next step, which is a very common step in text preprocessing. By removing stop words we remove the low level information, and give more focus to the important information.
- Removing all non-letter characters is the last part of preprocessing. This is also a very common step.

5 FEATURE EXTRACTION

There are following groups of features extracted:

- word and character n-grams
- linguistic features
- sentiment features
- Part-of-Speech features
- word embeddings

In this section we will describe each of these groups of features.

5.1 Word and Character N-grams

As we know word n-grams are used to capture the context in which words are, that is capture the sequence information of consecutive words. Same holds for character n-grams. In final solution, using TfidfVectorizer from *sklearn* [9] library unigrams, bigrams and trigrams were extracted. In appendix A all parameters that were used for this extraction can be found.

5.2 Linguistic features

In this category following features were extracted:

- number of words in preprocessed line
- number of terms in original line
- number of unique terms in original line
- number of characters in preprocessed line
- number of characters in original line
- number of syllables
- FKRA and FRE

Number of words, character, terms, syllables, unique terms are very important features for sentiment/emotion analysis. These features can tell us a lot about how a person is feeling since different emotions make us speak more or less.
Extraction of most of these features is straightforward except from Flesch–Kincaid Grade Level (FKRA) and Flesch Reading-Ease (FRE). These are Flesch–Kincaid readability tests designed to express how difficult a part of text in English is to read. This is very helpful for sentiment analysis since more complex text is, more probable it is that it contains serious content.
Flesch–Kincaid grade level is computed using following formula [10]:

$$FKRA = 0.39\left(\frac{total\ words}{total\ sentences}\right) + 11.8\left(\frac{total\ syllables}{total\ words}\right)$$

Flesh reading ease is computed using following formula [10]:

$$FRE = 206.835 - 1.105\left(\frac{total\ words}{total\ sentences}\right) - 84.6\left(\frac{total\ syllables}{total\ words}\right)$$

5.3 Sentiment features

For extracting sentiment features opinion lexicon [11, 12] was used. This lexicon consists of a set of positive words and set of negative words, and it is implemented in *nltk* [13] Python library. Following features were extracted:

- number of positive words
- number of negative words
- ratio between positive and negative words
- index of first positive word
- index of first negative word

All of these features can tell us a lot of emotions in the line. Also, indexes can help determine if there are two emotions inside one line. These features are very important and play big role in emotion analysis. In [2] similar approach is used. They also used Opinion Lexicon from Liu et. al. [11], but also several other lexicons. Since they handled emotion intensity, they also used those other lexicon to include word strength.

5.4 PoS features

Part-of-Speech (PoS) tagging is a process where each word is assigned its part-of-speech category. In this project not whole lines are tagged but only preprocessed ones. This tags are then grouped as ungirams, bigrams or trigrams using TfidfVectorizer once again. In appendix B are parameters of TfidfVectorizer.

5.5 Word Embeddings

Word embeddings are machine learning algorithms that transform words into vectors, where similar words have similar values [16]. In this project GloVe [15] to convert the lines into a 100-dimensional feature vectors. There are also very famous Word2vec word vectors, but in [15] they explained why such algorithms work and they reformulated Word2vec optimizations as a special kind of factorization for word co-occurrence matrices.
These feature vectors greatly help in sentiment and emotion analysis since words that produce similar sentiment or emotion will have similar vector values. Therefore, it will be easier to determine whether a line belongs to certain emotion class.

6 EVALUATION METRICS

In single-label classification we use simple metrics such as precision, recall, accuracy, etc,. In multi-label classification evaluation metrics usually differ, because a we can not define a miss-classification as strictly wrong or right. If a certain prediction contains a subset of actual labels, it is more accurate than the prediction that does not contain actual labels at all. There are two methods that can be used for evaluation [18]:

- Micro-averaging - in this method all true positives (TPs), true negatives (TNs), false positives (FPs) and false negatives (FNs) for each class are summed up and than average is computed. Here we see computation of precision and recall:

$$Prc^{micro} = \frac{\sum_{c_i \in C} TPs(c_i)}{\sum_{c_i \in C} TPs(c_i) + FPs(c_i)}$$

$$Rcl^{micro} = \frac{\sum_{c_i \in C} TPs(c_i)}{\sum_{c_i \in C} TPs(c_i) + FNs(c_i)}$$

- Macro-averaging - in this method we just take the average of the precision and recall of the system on different sets:

$$Prc^{macro} = \frac{\sum_{c_i \in C} Prc(c_i)}{|C|}$$

$$Rcl^{macro} = \frac{\sum_{c_i \in C} Rcl(c_i)}{|C|}$$

In this project micro and macro averaging methods will be used for evaluation of used models. Apart from this individual class accuracy will be reported.

7 RESULTS

In this section all used approaches for model training will be described. Apart from the approaches described here, usage of deep learning models, namely LSTMs was tested but it unfortunately gave much worse results than the models presented in this section.

7.1 Prepairing Dataset

After the XED dataset was preprocessed and features were extracted, train-test-split was used to prepare dataset for model training. The parameters are:

- random_state = 20
- test_size = 0.2

7.2 OneVsRestClassifier

One-vs-the-rest (OvR) multiclass strategy, also known as one-vs-all is very popular multilabel classifier. This estimator uses the binary relevance method, which involves training one binary classifier independently for each label. This classifier requires 2D target label matrix, where [i, j] == 1 says that sample i belongs to the class j. One-vs-the-rest multiclass strategy will be used to solve this multi-labeled dataset problem in combination with two methods: linear regression model and linear support vector classifier model. These two models will than be evaluated using mentioned evaluation methods - micro and macro averaging and individual class accuracy since OvR strategy allows to have insight in each classifier individually. The implementation of the OvR classifier will be used from sklearn library [17].

7.3 LinearRegression

Implementation of linear regression was used from sklearn library [19]. In figure 5 results of train and test accuracy for each class is shown.

Micro and macro averaging metrics produced following results:

- Micro averaging precision: 0.99845453764918
- Micro averaging recall: 0.8412181712962963
- Macro averaging precision: 0.9984646583962933
- Macro averaging recall: 0.8412603952409143

	category	test_accuracy	train_accuracy
0	anger	0.785117	0.780991
1	anticipation	0.804153	0.808322
2	disgust	0.868763	0.866013
3	fear	0.855495	0.862696
4	joy	0.847419	0.843225
5	sadness	0.854052	0.860532
6	surprise	0.862417	0.863561
7	trust	0.845976	0.845244

Figure 5: Results of Logistic regression model for each class

The results for this model are reasonably good since we know that sentiment and emotion analysis is very hard NLP task. We can also see that micro and macro averaging precision is very high which means that model is not creating a lot of false positives, but rather more false negatives. Although the precision is very high this model rarely predicts positive class (it often does not predict none of the available emotions) - as we see from precision and recall it much more predicts false negatives than false positives.

7.4 LinearSVC

Implementation of Linear Support Vector Classifier was also used from sklearn library [20]. In figure 6 results of train and test accuracy for each class is shown.

	category	test_accuracy	train_accuracy
0	anger	0.677819	0.696762
1	anticipation	0.803865	0.808466
2	disgust	0.760888	0.766424
3	fear	0.844823	0.853826
4	joy	0.843380	0.843369
5	sadness	0.853187	0.861037
6	surprise	0.822325	0.818923
7	trust	0.846265	0.844307

Figure 6: Results of SVC model for each class

Micro and macro averaging metrics produced following results:

- Micro averaging precision: 0.7854812398042414
- Micro averaging recall: 0.8413574286108428
- Macro averaging precision: 0.7876255420332559
- Macro averaging recall: 0.8521901399604622

Individual accuracy results of this model are very similar to the Logistic regression model, although slightly worse in almost all categories. On the other side, precision of this model is lower than in the case of logistic regression model, while recall remained approximately the same.

8 DISCUSSION

In this section we will try to apply trained models to MARVEL characters and see what were chosen characters emotions in movies. First important thing to emphasize is that the models were trained on texts from different movies. Of course the training set is in the movie domain, but we need to understand that each movie has its own topics and characters talk about different things. It is possible that in MARVEL movies characters talk more about technical topics than in other movies and this can effect the efficiency of model. In ideal case we would need annotated data from the same domain as the topics in movie.

Here we will analyse two main characters: Tony Stark and Steve Rogers. On figures 7, 8, 9 and 8 we can see emotion prediction of their lines. As we can see these models give totally different results. Logistic regression predicts more of anger, anticipation and disgust emotions, while SVC predicts more of joy and sadness and trust. Surprise and fear emotions were were not predicted by either of models for either of two characters considered. This can be caused by domain problem discussed previously. Moreover it can be noticed that SVC model has assigned labels to many lines while Logistic Regression model has assigned labels for really small number of lines. This was also discussed in section 7 where it was noticed that this model rarely predicts positives for all of the classes. This is not a good characteristic of a multi-label classifier.

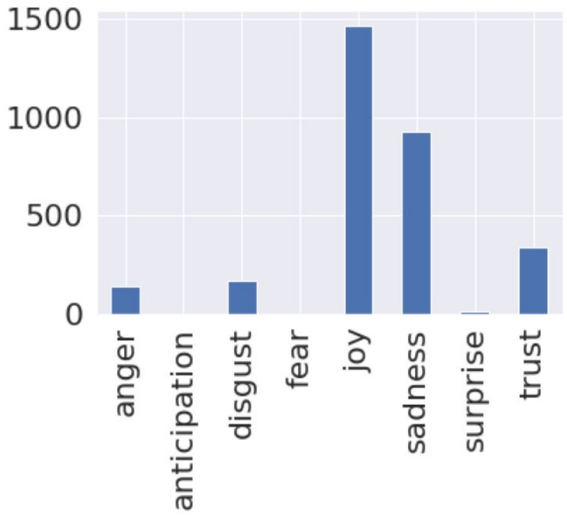

Figure 7: Results of SVC model for Tony Stark

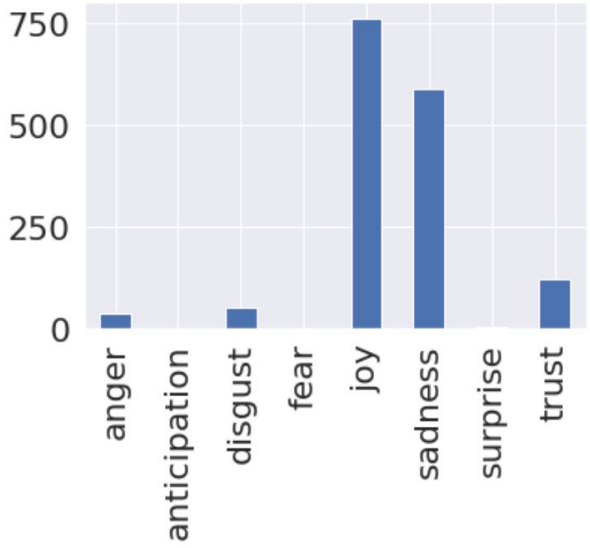

Figure 8: Results of SVC model for Steve Rogers

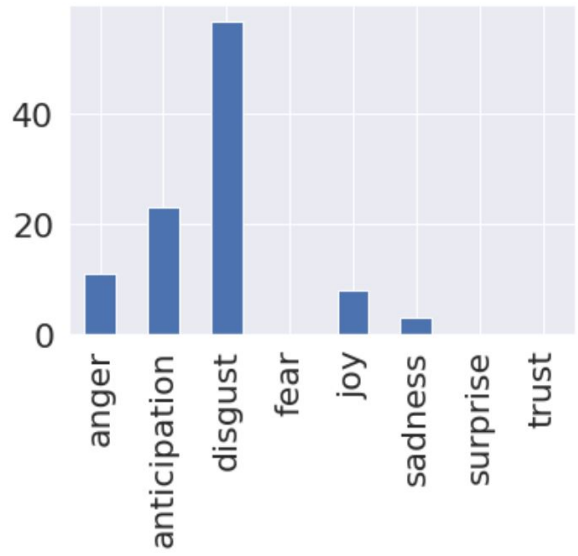

Figure 9: Results of Logistic Regression model for Tony Stark

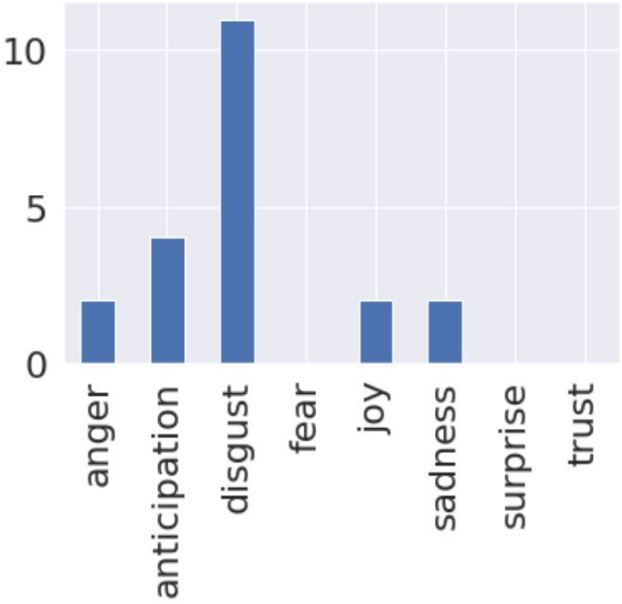

Figure 10: Results of Logistic Regression model for Steve Rogers

9 CONCLUSION AND FURTHER WORK

Character analysis can be very useful and applied to analysis of users on forums, twitter, reviews and for person profiling. In this project training and test data were from slightly different domains so that is why results were not as good as they can be (like if one could have annotated data for emotions of users on social media). In those cases one could add more domain specific features to improve results. Also there are many more multi-label approaches that could be tested and potentially give better results.

REFERENCES

[1] Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. International Conference on Text, Speech and Dialogue pages 196–205.
[2] Gupta, Raj & Yang, Yinping. (2018). CrystalFeel at SemEval-2018 Task 1: Understanding and Detecting Emotion Intensity using Affective Lexicons. 256-263. 10.18653/v1/S18-1038.
[3] Alm, C.O., Roth, D., Sproat, R.: Emotions from text: machine learning for text-based emotion prediction. In: Proc. of the Joint Conf. on Human Language Technology/Empirical Methods in Natural Language Processing (HLT/EMNLP), pp. 579–586 (2005)
[4] Tao, J., Fang, X. Toward multi-label sentiment analysis: a transfer learning based approach. J Big Data 7, 1 (2020). https://doi.org/10.1186/s40537-019-0278-0
[5] Öhman, E., Pàmies, M., Kajava, K. and Tiedemann, J., 2020. XED: A Multilingual Dataset for Sentiment Analysis and Emotion Detection. In Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020).
[6] Marvel Dialogue Dataset: https://github.com/prestondunton/marvel-dialogue-nlp
[7] Movie Dialogues Annotated Dataset: https://github.com/Helsinki-NLP/XED
[8] https://transcripts.fandom.com/wiki/Category:Marvel_Transcripts
[9] https://scikit-learn.org/stable/
[10] https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests
[11] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. ACM SIGKDD pages 168–177.
[12] Bing Liu, Minqing Hu and Junsheng Cheng. "Opinion Observer: Analyzing and Comparing Opinions on the Web". Proceedings of the 14th International World Wide Web conference (WWW-2005), May 10-14, 2005, Chiba, Japan.
[13] https://www.nltk.org/_modules/nltk/corpus/reader/opinion_lexicon.html
[14] https://www.nltk.org/book/ch05.html
[15] Pennington, J., Socher, R. and Manning, C.D. 2014. GloVe: Global Vectors for Word Representation. Empirical Methods in Natural Language Processing (EMNLP) (2014), 1532–1543.
[16] Kawin Ethayarajh. 2019. Word Embedding Analogies: Understanding King - Man + Woman = Queen. (2019).
[17] https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html#sklearn.multiclass.OneVsRestClassifier
[18] https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff
[19] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[20] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html?highlight=svc#sklearn.svm.LinearSVC

A WORD AND CHARACTER N-GRAMS

As mentioned before, for extraction of word and char n-grams TfidfVectorizer vectorizer was used with following parameters:

(1) word n-grams:
- ngram_range=(1,3),
- tokenizer = None,
- preprocessor = None,
- stop_words = None,
- max_features = 300,
- max_df = 0.90

(2) char n-grams:
- ngram_range=(1,3),
- tokenizer = None,
- preprocessor = None,
- stop_words = None,
- max_features = 200,
- max_df = 0.85

B POS N-GRAMS

Following parameter for extracting PoS n-grams using TfidfVectorizer are used:

- tokenizer=None,
- lowercase=False,
- preprocessor=None,
- ngram_range=(1, 3),
- stop_words=None,
- use_idf=False,
- smooth_idf=False,
- norm=None,
- decode_error='replace',
- max_features=50,
- min_df=0.1,
- max_df=0.80