GRASIE
(Grammatical Adapter for Sentiments and Intention in Emails)

Abir Lettifi, Adriana Nicoara, Dalila Ladli, Camille Saran, Colm Rooney, Jorge Vasquez

Abstract

We have all received at least one short tempered, arrogant email which prompted us to reply with a similar attitude; although, in most cases, we end up regretting this unprofessional behavior. Therefore, GRASIE exists! The goal of this project is to give to every person, in a professional environment, the opportunity to avoid displaying their current state of emotions through their emails in order to maintain a professional state. This will be achieved through sentiment analysis (SA) and grammatical error correction models.

1 INTRODUCTION

GRASIE's main characteristics is analyzing the sentiments displayed in an email, suggest another wording and correct any grammatical error. The ideal output is an email that fits the professional criteria while granting the user a more appropriate tone before sending it. GRASIE will be implemented with different email templates that matches the professional relationship between the sender and the receiver in order to provide to the user several corrections templates. More concretely, in addition to a spelling analysis, our application will be able to :

- Detect strong negative sentiments in an email and suggest other wordings.
- Detect and propose some elements of elevated language.
- Identify the use of excessive, familiar language and choose an appropriate vocabulary.

GRASIE will be implemented as a website to allow wider access. Furthermore, our target base is directed towards people with business interactions, who could benefit from systematically having their emails reviewed and improved before sending them.

In the following sections the grammatical correction feature will be discussed alongside with the methods used to train and build the SA model. Afterwards, the construction of the corpus will be presented. Finally, the results of our models will be shared in addition to a data interpretation part and the conclusion.

2 STATE OF THE ART

Sentiment Analysis (SA) is a sub domain of natural language processing that is used to detect any subjectivity, opinion, or sentiment in data. Overall, any SA related work uses labeled data for the model's training, this can be considered one of the biggest issues in this domain since the data needs to be manually annotated, which takes time and exterior professional help. The output of these models is a polarity score that shows whether the sentiments are negative, positive, or neutral. Currently, the two main methods in SA are [4]:

- The lexicon-based method (also known as rule-based or dictionary-based method).
- The machine learning method.

The lexicon-based methods use lexicons which are pre-annotated words usually called Sentiment Bearing Words (SBWs), for example [7] developed their own lexicon to train their model Muse (Memories Using Email), it has 45 emotion categories and around 500 terms. Although lexicon-based methods seem popular among researchers because of how unbound they are to quality and quantity of training data, they get outperformed by modern machine learning based methods [11]. However, the lack of large, annotated corpora makes the use of machine learning methods complicated [12]. Moreover, the few research conducted on email sentiment analysis focus on spam detection, but recently we notice the application of data processing techniques to emails resulting into the appearance of email clustering, email classification, topic modelling and email summarization. For example, [8], introduced the LDA approach to determine the categorical terms, which mixed to email clustering resulted into email categorization. On the other hand, [14] introduced a hybrid text clustering-based classification, which is named ML-EC2. In addition, a supervised machine learning algorithm was used into creating a classifier [12]. In this model, the training data is the Enron email data set. According to the authors, during the labelling phase, the lexicon labelling outperformed the k-mean labelling. Consequently, it allowed them to label the data set and train the classifier while obtaining better results. Ultimately, in order to work on the prediction of any sentiment's intensity degree, deep learning and feature-based models got combined resulting into the appearance of three different deep learning techniques: CNN (convolutional neural network), LSTM (long short-term memory), and GRU (gated recurrent unit). Along with these three, one supervised models SVR was also developed [1].

3 MATERIALS AND METHODS

3.1 Grammatical errors correction

This idea saw the light thanks to a French company called ”Synapse Développement” [1]. 25 years ago, a group of linguists and computer scientists worked together in order to automate language processing and create innovative solutions. From their work, a software was born: the ”cordial.fr” [2]. It is a Windows-based grammar correction and editing software for the French language.

Therefore, for the Grammatical Error Correction part, a python framework called Gramformer[3] was appropriate for this project. It has been trained with a large corpus built from C4 [15] and PIE [3], containing synthetic data. In addition, some python libraries were used and the most relevant are:

- **Transformers** – provides general purpose architectures for Natural Language Understanding (NLU) and Natural Language Generation (NLG) with over 32+ pretrained models in 100+ languages and deep interoperability between Jax, PyTorch and TensorFlow.
- **SentencePiece** – unsupervised text tokenizer and detokenizer mainly for Neural network based text generation systems where the vocabulary size is predetermined prior to the neural model training.
- **Python-Levenshtein** – module that contains functions for fast computation of Levenshtein (edit) distance, edit operations, string similarity, approximate median strings, and generally string averaging, string sequence and set similarity.
- **Lm-scorer** – simple programming interface to score sentences using different ML language models.

To make sure the corrections and highlights recommended are of high quality, the framework Gramformer comes with a quality/score estimator. The score is calculated based on the probabilities of text fragments and calculates how likely a sentence is to appear. The calculation is done for each token: predict the probability distribution of the next token given previous context and the final result is given by the product rule of probability (aka the chain rule formula (1)) .

$$P(S) = \prod_{i}^{n} P(w_i|w_1, w_2, ..., w_{n-1}) \qquad (1)$$

This model works at sentence level, which means any text must be first split into sentences. Therefore, when the user inputs their

[1] https://www.synapse-developpement.fr/qui-sommes-nous

[2] https://www.cordial.fr/meilleur-correcteur

[3] https://github.com/PrithivirajDamodaran/Gramformer

email on the GRASIE website, the first step will be to isolate each sentence.

Since we are dealing with human language data, the *NLTK* library will be used as it has packages for symbolic and statistical natural language processing. It supports classification, tokenization, stemming, tagging, parsing and semantic reasoning functionalities.

The process of splitting text in sentences is known as tokenization by point process, therefore for this step, the function *PunktSentence Tokenizer* from the *NLTK* library will be used. After splitting in sentences, each sentence will be corrected and displayed, together with its grammatical score quality. Next, the new corrected sentences will be outputted to the user as suggestions for them to use.

3.2 Sentiment analysis

Sentiment analysis is a powerful tool that allows the computer to understand the underlying subjective tone of a piece of writing [13]. By using this, the user benefits by having an objective review of their choice of words while also receiving possible improved adjustments[4] to their email. This gives the sender the ability to neutralise their emotive responsive and communicate more professionally to their recipient. This could help avoid conflicts in the workplace as-well as the recipient being appreciative of the coherently messaged email they have received

The first step is loading the data then, passing it through an NLP pipeline to train the model[5]. The role of the pipeline is to reduce the noise inherent in any text written in a natural language, in order to improve the classifier's accuracy and to provide an understandable shape for the computer. The necessary steps implemented using NL Toolkit are:

3.2.1 Tokenization

The data used is in a text file format containing one sentence per line. A tokenization at word level will be done while preserving the one sentence per line form. Consequently, there will be multiple tokens/words, corresponding to the sentence, one line. The tokenization is made that any punctuation and non-words string will be considered as tokens.

3.2.2 Removing noise and stop words

Noise is specific to each project, so what constitutes noise in one project may not be considered as such in another project. The most common words in a language are called stop words. Stop words are important in human communication but they have weak values for machines. A default list of stop words from *nltk* will be used to filter the tokens' list. Additionally, hyperlinks, punctuation and special characters will be removed using regular expression.

3.2.3 Normalizing words

This section alludes to condensing all word forms into a single representation of words. The process is called lemmatization and converts the various forms of a word to its simplest form, known as lemma. Before running a lemmatizer, a determination of the part of speech for each word in the text is made, using the *POS TAG* function from *NLTK*. During this process each word is tagged based on its definition and context.

3.2.4 Preparing data for the model

First, the tokens from each sentence are converted into a dictionary. By this, the multiple occurrences of a token in the same sentence are eliminated. Considering that the data comes from 2 files (positive and negative) saved in 2 variables, we passed them through the processing pipeline in parallel. Because this model is for supervised learning, as one of the final steps, each new obtained data set is assigned the representing label: positive or negative.

The final steps, training and testing, consisted of merging the data from the positive and negative data sets and shuffling them in order to use 87% of the data set for training and the remaining 13% for testing.

The classification workflow for the model looks like:

[4] https://towardsdatascience.com/a-beginners-guide-to-sentiment-analysis-in-python-95e354ea84f6

[5] https://towardsdatascience.com/sentiment-analysis-with-python-part-1-5ce197074184

- Split data into training and testing sets.
- Select a model architecture.
- Train the model with training data.
- Use the testing data to evaluate the performance.
- Use the model with new data to generate predictions.

3.2.5 Predict sentiments using machine learning (ML) classifiers

(this should be in the state of the art)

There are several tools available in python that solve the classification problem:

- TensorFlow.
- PyTorch.
- Scikit-learn.

All these frameworks possess solutions that can be used in areas like computer vision, natural language processing and predictive data analysis. They are categorized as machine learning tools. Despite this, we choose NLTK because it is a specialized tool built especially for natural language processing.

3.3 KeyBERT

KeyBERT[6] is a keyword extraction technique that leverages BERT [5] embedding to create keywords and key phrases that fit the document in the most appropriate way.

From this framework, the **keybert** library will be used, alongside a method that extracts the keywords from a text.

3.4 Templates

The idea of using templates in order to generate the output of GRASIE was inspired by these researches [2] [9].

The templates that GRASIE use are related to multiples topics covered in basic professional environments such as Invitations, Recommendations, Support Requesting, Holiday Requesting, and others.

The templates that GRASIE used for the experimental step were about Invitation and Recommendations. These templates were based in the templates suggested by the web-articles 'Letter of recommendation template'[6] and '9 Best Business Event Invitation Email Examples Ideas'[7].

Example of a template's type we used:

Dear [receiver-name],

I'm very pleased to invite you to the [event] which will take place the [date], we are holding this event in the [place]. I will be very grateful with your participation and your time.

Regards, [sender-name]

The variables [receiver-name], [event], [date], [place] and [sender-name] need to be completed with the information recollected from the original email which is an input for GRASIE.

3.5 Corpus Construction

During the building of the corpus, we searched for data sets that provided data, classified by positive or negative sentiments. This data in turn would be used to train the Sentiment Analysis model. For the Grammatical Error Correction function, there was no need for data, considering that Gramformer provides this function for us. The main sources used during this process are free public data sets: Sentiment Labeled Sentences Dataset [10] and Movie Review Data(2004)[8].

3.5.1 Movie Review Data

Movie Review Data is a data set that provides movie-review data to be used in SA experiments such as: Sentiment polarity data sets, Sentiment scale data sets and Subjectivity data sets. This data set is fitted with sentences (comments of Movie Reviews) already tokenized, with them classified as positive and negative. From this data set two sub data sets have been created to be part of our corpus, these are called 'Sentence Polarity Dataset v1.0' and 'Polarity Dataset v2.0'.

[6] https://resources.workable.com/letter-of-recommendation-template-sample

[7] https://www.virtualedge.org/business-event-invitation-email-examples-ideas/

[8] https://www.cs.cornell.edu/people/pabo/movie-review-data/

3.5.2 Sentiment Labeled Sentences Dataset

This data set was generated in 2015 and contains three different files with reviews for products, movies, and restaurant from three different websites: imdb, amazon and yelp. Each file contains 1000 sentences labelled with positive or negative sentiments, of which are 500 positive and 500 negative. Since each file contains both types of sentences, we separated them into two files, according to sentence polarity.

3.5.3 BC3 Email Corpus

BC3 Email Corpus is licensed under *Creative Commons Attribution-Share Alike 3.0 Unported License.*.The BC3 framework is open-sourced and licensed under the MIT license. It's made of 837 sentences from 261 emails that treat 40 different topics. It's the only corpus that has an Email type form. For our final corpus we used BC3 to get positive or negative sentences. We analyzed 541 of those sentences to extract only the positive and negative one. The neutral sentences were not used in our final corpus. For the analyzing we used 3 online sentence sentiment analysis demo: Monkeylearn, Text2data and Komprehend. Each of them has a free demonstration that we used to analyze the sentiment of the sentence. As a final result, on 541 sentences we got 91 positive sentences and 119 negatives sentences.

As a result, our final corpus will consist of classified data from the 3 data sets presented in the previous subsections, 3.1, 3.2 and 3.3. It will be represented by 2 files, a positive file, with all the sentences labelled with this sentiment, and a negative one.

The training stage contains 3 phases and each of them bring an improvement in the efficiency of the model. In the first phase the model was trained just with movie review data being the only one found at that time. We got some reasonable results, but we wanted to improve even more the model, so we kept going because more data means more training and therefore efficiency. In the second phase we use a corpus created with the first 2 datasets and since this application works with emails, we finally found the third dataset BC3 to get an email type dataset.

Moreover, because internally the application works at the sentence level, we cannot say that we were forced to work only with data coming from emails. The only constraint was to have some objective sentences not in a familiar language which reflects some positive or negative feelings.

3.6 GRASIE methodology for generating an email suggestion

GRASIE receives the input email that will be passed through a processing pipeline. Each step of the pipeline is executed sequentially, this means, the following step will not start its execution while the previous one has not finished yet. To put everything together, the execution order is resumed in the following subsection. Also a graphical representation of the complete process can be seen in Figure 1 .

3.6.1 Gramformer - correct the grammatical errors

The Gramformer model receives the input email and applies the grammatical corrections. As a result, a corrected email is generated, ready to be used as input for the next step.

3.6.2 Sentimental Analysis - classify the text

The pre-trained classifier receives as input corrected email after is is cleaned by removing the noise, in order to give a supposition: positive or negative.

3.6.3 Construction of the New Email Suggestion

In the case where the result of the email's classification is positive, the output consists of the same email but with grammatical corrections. We made this choice because the positive classification indicates that the email is already assertive. However, when the email is classified as negative, an assertive email version of it is then suggested. In order to achieve this task, the templates and KeyBERT framework will be used.

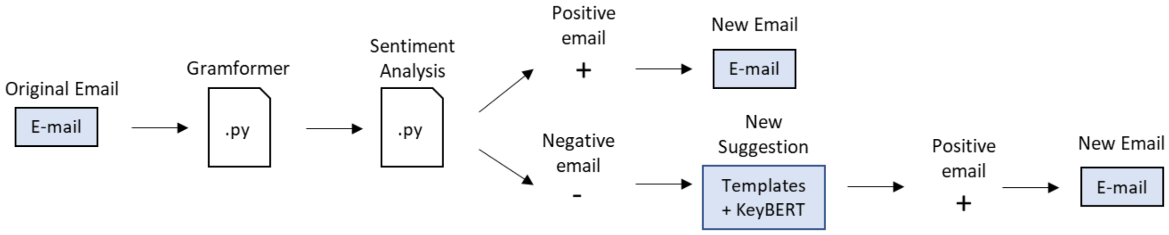


Figure 1: Graphical representation of the methodology

As said before, GRASIE has email templates for multiple professional topics. These templates will have some blank spaces that will be completed with keywords from the original email.

Succeeding GRASIE's identification of the appropriate template that will be used, the keywords, that will fill the blanks, will be identified by using KeyBERT and stored in a dictionary. In addition, to extract the date information from the email, the library called **datefinder**[9] will be used.

In order to know how the template will be filled, the keyword dictionary will be compared with a source dictionary that contains the variables and values that represent the blank spaces from each template. After the comparison between these two dictionaries, the keywords will be filled into the blanks from the template. As a result, an assertive email suggestion will be obtained.

4 RESULTS

4.1 Stage 1 - Movie Review

Training and testing our model with the corpus for SA lead us to some results:

4.2 Accuracy

The accuracy of our model was obtained by evaluating annotated data sets then comparing the results from our model with other known results.

The trained model proved 75% accurate in assessing text with sentiment as sentimental. However, we acknowledge accuracy alone is not a true representation of the functionality of a model of this sort.

4.3 Precision and Recall

Since these are some important statistical parameters we evaluated precision, recall and f-score with the help of some functions from *NLTK*.

The precision and recall measure the exactness and sensitivity of our classifier. Higher precision equates to less false positives, and higher recall means less false negatives.

A false positive is an outcome where the model incorrectly predicts the positive class, a negative sentence such as ”I liked the plot but the movie was terrible” wrongly predicted by our program as positive. While, a false negative is an outcome where the model incorrectly predicts the negative class, for example like ”I have so much money and I don't know what to do with it”, a positive sentence being classed as negative.

Scoring 0.7382 in precision and 0.7716 in recall proves our model is effective in both correctly analysing and identifying sentiment in our test. This statement was further reinforced by the 0.7597 f-score our program outputted (Table 1).

Metric	Results
Accuracy	0.7543
Precision	0.7382
Recall	0.7716
F-measure	0.7597

Table 1: Results Stage 1

4.4 Stage 2

Stage 2 of testing our model resulted in minor improvements for each metric.

[9] https://github.com/akoumjian/datefinder

4.4.1 Testing Data

Stage 2 of evaluating our model, we combined the Movie-review data set along with the labeled sentences data set we found from Yelp, Amazon and IMBD. We choose this data set because of its size and the fact it was already labeled at sentence level for our model. The decision to increase the size of our corpora was a success as it increased the effectiveness of our model. The increase in scores is most likely attributed to our model having a larger data set to train and learn from (Table 2).

Metric	Results
Accuracy	0.7721
Precision	0.7738
Recall	0.7721
F-measure	0.77

Table 2: Results Stage 2

4.5 Stage 3 - BC3 Email data set

Stage 3 was the final stage of training and testing of our model. It was decided for the last addition to our corpora be a data set entirely consisting of emails segmented at the sentence level, this was agreed upon as our model was created for the intention of interacting and perfecting email's. The only inconvenient is that this corpus has a small size.

4.6 Final Results

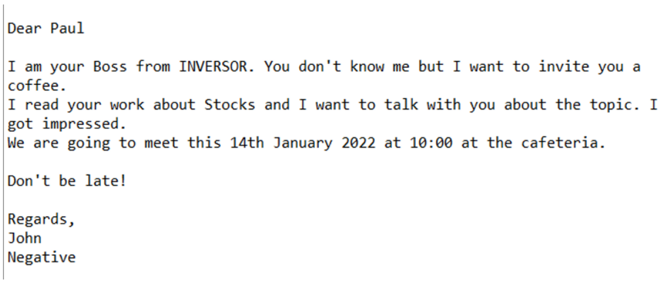
Here is the final evaluation of our model tested and trained with our chosen corpora (Table 3). As we can see in the next table, the performance hasn't changed, most likely because the new data added was not very large.

Metric	Results
Accuracy	0.7721
Precision	0.7738
Recall	0.7721
F-measure	0.77

Table 3: Final Evaluation

4.7 Testing GRASIE

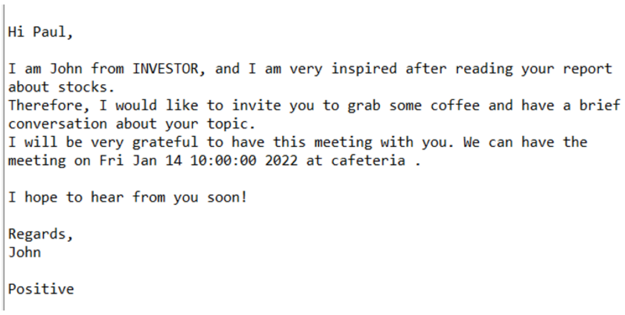
With the results obtained from the evaluation of our SA model, the execution of tests with GRASIE can begin. For this test we are going to use an invitation email. This input followed the first two steps of the GRASIE's process and as a result it was classified as Negative (Figure 2).



```
Dear Paul

I am your Boss from INVERSOR. You don't know me but I want to invite you a
coffee.
I read your work about Stocks and I want to talk with you about the topic. I
got impressed.
We are going to meet this 14th January 2022 at 10:00 at the cafeteria.

Don't be late!

Regards,
John
Negative
```

Figure 2: Input Email Classified as Negative

Because the input email is negative, it is necessary to construct a new email suggestion. For this, the step 3 and 4 of GRASIE's process are executed. As a result, after identifying the keywords from the input email, the template are filled with those keywords to obtain an assertive email just liked represented in (Figure 3).



```
Hi Paul,

I am John from INVESTOR, and I am very inspired after reading your report
about stocks.
Therefore, I would like to invite you to grab some coffee and have a brief
conversation about your topic.
I will be very grateful to have this meeting with you. We can have the
meeting on Fri Jan 14 10:00:00 2022 at cafeteria .

I hope to hear from you soon!

Regards,
John

Positive
```

Figure 3: Output Email Classified as Positive

5 CONCLUSION

According to the results discussed earlier (Figure 4), it is clear that we can improve our website to make it generate automatic emails without using templates. This is an ambitious goal that will be difficult to achieve but we believe it is possible.

5.1 Ongoing problems

The part for grammatical correction that used the Gramformer model was first implemented on Google Collab. After finishing both models, we wanted to put them together locally

Evaluation results			
...	Stage 1	Stage 2	Stage 3
Accuracy	0.7543	0.7721	0.77
Precision	0.7482	0.7738	0.77
Recall	0.7716	0.7721	0.77
F-Measure	0.7597	0.77	0.77

Figure 4: Final Evaluation Results of the 3 Stages

on a computer. The Gramformer model did not work because of a conflict between multiple versions of libraries. We tried to solve the conflict but with no success. This remains an open issue and we are still searching for other options.

Therefore this project for now is just a prototype that can't be used by the public because each model works independently.

5.2 Future improvements

Speaking of ambitious work, these are the potential features that will improve GRASIE's effectiveness and the user's experience:

- Summarization of received emails to help the user avoid wasting time.
- Gain time and choose the priority of the received emails without having the user read them.
- Classification and grouping of received emails based on emotional content.
- Adding emotion analysis to output a more precise feedback of the email.
- Incorporate ML to make the platform self-improving.
- Simplify sentences considered too complex.
- Each tag in the result will be accompanied by a degree of confidence; for example a text can be just 70% positive.
- Detect unsolicited emails and categorize them as spam.

References

[1] Md Shad Akhtar, Asif Ekbal, and Erik Cambria. "How Intense Are You? Predicting Intensities of Emotions and Sentiments using Stacked Ensemble [Application Notes]". en. In: *IEEE Computational Intelligence Magazine* 15.1 (Feb. 2020), pp. 64–75. ISSN: 1556-603X, 1556-6048. DOI: 10.1109/MCI.2019.2954667.

[2] Abdulkareem Al-Alwani. "A novel email response algorithm for email management systems". In: *Journal of Computer Science* 10 (Apr. 2014), pp. 689–696. DOI: 10.3844/jcssp.2014.689.696.

[3] Abhijeet Awasthi et al. "Parallel Iterative Edit Models for Local Sequence Transduction". In: (2019). DOI: 10.18653/v1/D19-1435.

[4] Anaıs Collomb et al. "A Study and Comparison of Sentiment Analysis Methods for Reputation Evaluation". In: (2013), p. 10. URL: https://mobisocial.stanford.edu/papers/informatics11.pdf.

[5] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: (2019), pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[6] Maarten Grootendorst. "KeyBERT: Minimal keyword extraction with BERT." Version v0.3.0. In: (2020). DOI: 10.5281/zenodo.4461265.

[7] Sudheendra Hangal and Monica S Lam. "Sentiment Analysis on Personal Email Archives". en. In: (2012), p. 4. URL: https://mobisocial.stanford.edu/papers/informatics11.pdf.

[8] *Identifying Categorical Terms Based on Latent Dirichlet Allocation for Email Categorization — SpringerLink*. 2018. URL: https://link.springer.com/chapter/10.1007%2F978-981-13-1498-8_38.

[9] Leila Kosseim, Stéphane Beauregard, and Guy Lapalme. “Using Information Extraction and Natural Language Generation to Answer E-Mail”. In: *Natural Language Processing and Information Systems Lecture Notes in Computer Science* (2001), pp. 152–163. DOI: 10.1007/3-540-45399-7_13.

[10] Dimitrios Kotzias et al. “From Group to Individual Labels using Deep Features”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), pp. 597–606. DOI: https://dl.acm.org/doi/10.1145/2783258.2783380.

[11] Heidi Nguyen et al. “Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches”. en. In: 1.4 (2018), p. 23. URL: https://scholar.smu.edu/cgi/viewcontent.cgi?article=1051&context=datasciencereview.

[12] Rayan Salah and Neamat El Gayar. “Sentiment Analysis using Unlabeled Email data”. en-US. In: (Dec. 2019). Number: 2080 Publisher: EasyChair. ISSN: 2516-2314. URL: https://easychair.org/publications/preprint/HkRB.

[13] “Sentiment analysis using product review data”. In: *Journal of Big Data 2, 5* (2015). DOI: https://doi.org/10.1186/s40537-015-0015-2.

[14] Aakanksha Sharaff and Naresh Kumar Nagwani. “ML-EC2: An Algorithm for Multi-Label Email Classification Using Clustering”. en. In: *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)* 15.2 (2020). Publisher: IGI Global, pp. 19–33. URL: https://ideas.repec.org/a/igg/jwltt0/v15y2020i2p19-33.html.

[15] Felix Stahlberg and Shankar Kumar. “Synthetic Data Generation for Grammatical Error Correction with Tagged Corruption Models”. In: (Apr. 2021), pp. 37–47. URL: https://aclanthology.org/2021.bea-1.4.