Data Augmentation vs Multi-modality : Best approach towards fake news detection?

Ronak Nahata (1222310336), Hastin Himanshubhai Modi (1225543982),
Sayantan Sarkar (1224598375), Sai Pravallika Gadiparthi (1225789918),
Sai Krishna Chilvery (1225915511), Anudeep Reddy Dasari (1225435406)

December 8, 2022

Abstract

In recent years, there has been a heightened awareness of the detrimental effects of fake news. Traditional approaches to addressing this issue have relied solely on textual data [13][15], while multimodal approaches [16], which exploit both text and visual information, remain less common. However, multimodal approaches, although effective, are hindered by the need for larger amounts of training data. This study sought to evaluate the hypothesis that augmenting the textual data of the *Fakeddit* [20] dataset would lead to performance improvements in the unimodal approach that are comparable to those of multimodal approaches. The results were then compared to determine if the additional effort of collecting and labeling data in multiple modalities is worthwhile.

1 Introduction and Motivation

The circulation of unreliable information without any veracity, which is generated to echo the views expressed in the media, is referred to as fake news. Social media sites, such as Reddit, Facebook, and WhatsApp, have become primary sources of information, promoting the propagation of inaccurate facts. Due to the time-consuming and laborious nature of manually examining such data, automation of this process is essential to prevent the proliferation of untruths and the potential for conflict.

Unimodal approaches to detect fake news in natural language processing rely on a single source of data, such as text or images, to identify patterns and inconsistencies. This makes them prone to false positives, as they may identify patterns that are not actually indicative of fake news. Multimodal approaches, however, can combine data from multiple sources to identify inconsistencies and accurately detect fake news. By combining data from text, images, audio, and video, multimodal approaches can detect more complex patterns and nuances, enabling them to more accurately identify fake news. Despite its advantages, this technique entails hefty costs from the added labor of gathering and labeling multiple modalities of data not to mention the time taken in training such models.

Data augmentation affords us the chance to assess the tradeoffs associated with constructing machine learning/deep learning models for fake news detection, making evident its connection to the material presented in our course. To the best of our knowledge, this is the first work in literature to explore the question of multimodality vs data augmentation for the Fakeddit dataset.

2 Problem Description

The problem at hand is trying to consider the classification of fake news. For this, we used the Fakeddit dataset to train models using unimodal as well as multimodal approaches.

Since many unimodal and multimodal approaches have been proposed, we aim to investigate the effectiveness of different methods using the Fakeddit dataset as it has various advantages over other datasets:

- Higher number of samples for both text and image-based data
- More variety of data compared to other sources
- Multi-modal dataset, hence, could be used for a fair comparison across different types of models

The dataset contains a total of 878,218 training examples, 92,444 validation examples, and 92,444 testing examples for the unimodal approach. About 64% of the dataset also contains the corresponding images, which are used for the multimodal approach. Hence, for this approach, there is a total of 564,000 training examples, 59,342 validation examples, and 59,319 testing examples.

3 Methodology[1]

We have used both unimodal (text-based) approach and multimodal (both text and image-based) approach for the classification of fake news. Given further are the details of all the models used for both the approaches.

3.1 Unimodal approaches

The performance of five unimodal models is investigated: Naive Bayes, Logistic Regression, Random Forest, SVM, and CNN. Furthermore, we also provide details regarding the preprocessing done before passing the data into the models for the purpose of classification. For all the unimodal models except CNN, we have used the Optuna library [2] to find the optimal hyperparameters and used the implementation provided in scikit-learn [21].

3.1.1 Preprocessing for the unimodal aproaches

We follow a 4-stage process for preprocessing the data:

- Removing punctuations, numbers, and multiple spaces - We only keep the capitalized and small letters. Also removed the extra spaces between the words in a sentence.
- Removing stopwords - We use the stopwords corpus of the NLTK [17] library to remove some commonly used words in the English language such as ‘i’, ‘it’, ‘the’, etc. as they wouldn’t contribute much.
- Lemmatization - We convert all the words to their base form. For example, ”stripes” becomes ”stripe”. We use the Wordnet lemmatizer [7] with NLTK for performing lemmatization.
- Using tf-idf features or GloVe embedding [22] - Term frequency-inverse document frequency is the product of two statistics, term frequency, and inverse document frequency. It aims to define the importance of a word within a document. The formula for finding each is:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

Where $f_{t,d}$ is the raw count of a term in a document, i.e., the number of times that term t appears in a document d.

$$idf(t,D) = log\frac{N}{|\{d \in D : t \in d\}|}$$

Where N is the total number of documents in the corpus and the denominator is the number of documents where the term t appears.

The tf-idf is then calculated as: [25]

$$tfidf(t,d,D) = tf(t,d).idf(t,D)$$

We use the tf-idf features as input to 4 unimodal approaches - Naive Bayes, Logistic Regression, SVM, and Random Forest. The 300-dimensional GloVe embeddings are used as an input to the CNN model.

3.1.2 Naive Bayes

Naive Bayes is a probabilistic supervised classification algorithm that is commonly used for text classification tasks. The main assumption used by the

[1] The code for all the methodologies is present here: `https://tinyurl.com/49wuxu83`

model is that all the features are conditionally independent given the class. We have used the multinomial Naive Bayes provided in the scikit-learn adjusting 2 hyperparameters using Optuna:

- n-gram range - The maximum value of n-gram to be taken into account while calculating tf-idf, for our dataset, we found a maximum value of 2 to be optimal.
- minimum document frequency - The minimum number of documents in which a particular word must appear in order to be considered in the dataset. We found 5 to be the optimal number of documents.

3.1.3 Logistic Regression

Logistic Regression is a supervised classification algorithm that is one of the simplest and fastest classification algorithms since it does not involve any complicated calculations. For Logistic Regression, we have 3 hyperparameters to adjust using the Optuna library:

- n-gram range - Again, we found a maximum value of 2 to be optimal.
- minimum document frequency - Here also, we found the value of 5 to be optimal.
- solver - It is the algorithm to be used for solving the optimization problem. We found newton-cg to be the best for our dataset.

3.1.4 Random Forest

Random Forest falls under the category of bagging algorithms. It can be used for both, classification and regression problems. There are 3 main steps involved in the creation of a random forest:

- Creating a bootstrapped dataset using the original dataset.
- Creating a decision tree with a random subset of features at each node using the bootstrapped dataset.
- Creating multiple decision trees using the above 2 steps and aggregating the decisions from all the trees and choosing the class having the majority of votes.

For Random Forest, we have 5 hyperparameters to adjust using the Optuna library:

- n-gram range - We found a maximum value of 2 to be optimal.
- minimum document frequency - We found the value of 9 to be optimal.
- max depth - the max depth allowed in any tree of the random forest. The optimal value was found to be 6.
- criteria - the criteria to be used for splitting the nodes, the options were gini and entropy. We found gini criteria to give the best results.
- number of estimators - the total number of decision trees to be used for creating the random forest. We found the value of 290 trees to be optimal.

3.1.5 Linear SVM

SVM is a supervised classification algorithm that can be used for both linear and non-linear classification problems. Here, we are using a version of SVM which is used for linear classification. It aims to maximize the margin between positive and negative class examples. We have 3 hyperparameters to adjust using the Optuna library:

- n-gram range - We found a maximum value of 1 to be optimal.
- minimum document frequency - We found the value of 22 to be optimal.
- loss function - we found the squared-hinge loss to be optimal.

3.1.6 CNN

Convolutional Neural Network (CNN) is generally used for image classification tasks. Here, we are using it for text classification using the above-mentioned pre-processing techniques. Figure (2) shows the general architecture of a CNN architecture used for text classification. We have a vocabulary size of 142,270 words. The architecture of the CNN is as follows:

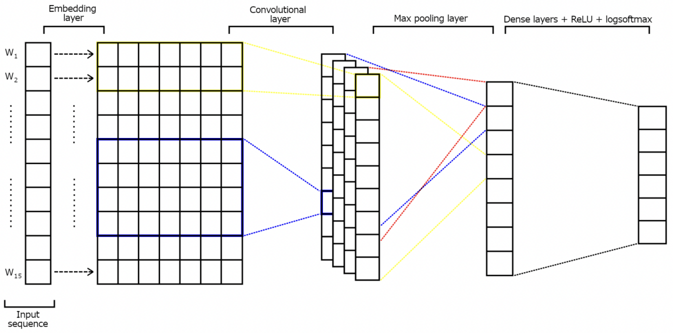

Figure 1: CNN for text classification

- 4 convolution layers with 50 filters, stride of (1, 1), and kernel sizes of (2, 300), (3, 300), (4, 300), and (5, 300)
- 2 fully-connected linear layers

The loss function used is negative log-likelihood loss and the learning rate is 0.003.

3.2 Data Augmentation

Data augmentation for text in natural language processing is important because it helps to increase the size of the training dataset, which helps improve the accuracy of the model. Data augmentation can also help to reduce the amount of time it takes to train a model, as larger datasets can be used, while still maintaining accuracy. Finally, data augmentation can also help to improve the generalizability of the model, as it can reduce the risk of the model over-fitting to the specific training data. Some of the proposed techniques are Easy Data Augmentation Techniques like synonym replacement, random insertion, random swap and random deletion[28], paraphrasing[14], Back-translation[23], Contextual Augmentation and Text Generation.

In this work, we employ Back-translation and Text Generation to augment our data. For both of these techniques, we employ large scale Language Models (LM) for the Machine Translation and text generation tasks. The details of the methods are explored in the following sections but essentially through clever prompt engineering, we were also able to achieve easy data augmentation, paraphrasing and more sophisticated texts all under the purview of LM-based text generation.

3.2.1 Back-Translation

Back translation as a data augmentation technique in natural language processing is a method of creating additional training data from existing data. It works by first translating an existing sentence from a source language into a target language, then translating the target language sentence back into the source language. For this, we employed the neural machine translation framework called MarianNMT[11]. More specifically, we used the pre-trained MarianMT models ***opus-mt-en-fr*** to convert from English to French and ***opus-mt-fr-en*** to convert from French to English. There are also techniques which run back translation in a chain to get more diversity. For example, English → French → English → Spanish → English. But for our experiments we just stuck to one round of chaining.

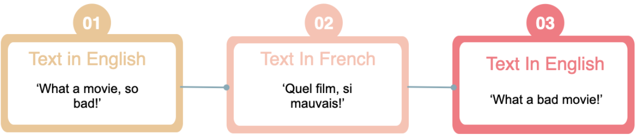

Figure 2: Back-translation flow in a nutshell

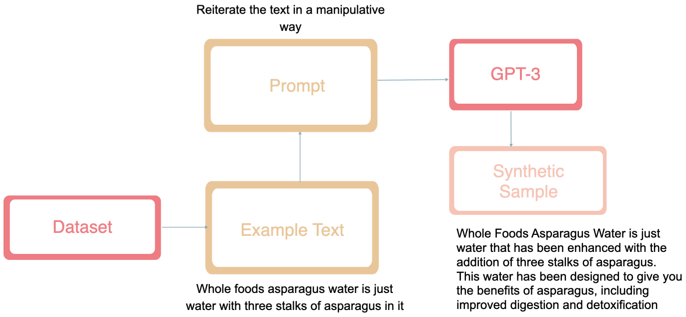
Reiterate the text in a manipulative way
Prompt
GPT-3
Synthetic Sample
Dataset
Example Text
Whole Foods Asparagus Water is just water that has been enhanced with the addition of three stalks of asparagus. This water has been designed to give you the benefits of asparagus, including improved digestion and detoxification
Whole foods asparagus water is just water with three stalks of asparagus in it

Figure 3: Text-Generation flow using GPT-3 in a nutshell

3.2.2 Text-Generation using GPT-3

Generative Pre-trained Transformer 3 (GPT-3) is an autoregressive language model that leverages deep learning to generate text in a fashion resembling that of a human. Presenting it with an initial prompt, GPT-3 builds upon this input to produce a response that follows on in a natural way. [29] For our use case of augmenting fake news, we have to engineer input prompts in a manner that retains the deceitfulness of a fake news while still producing text that is semantically equivalent to the input. In case of news with a not fake label, we also generate prompts in order to perform word-swap, synonym replacement and paraphrasing. We used the OpenAI's API to leverage GPT-3.

3.2.3 Augmented Data

We were able to augment 40000 new samples using both the techniques mentioned above. We augmented 20000 samples using Back-translation with an equal split for fake and non fake labels. Similar was the case with GPT-3 (with a whopping 175 billion parameters) based text completion. We also tried to experiment with various open source free altervatives to GPT-3 from a group named EleutherAI which made GPT-Neo (2.7 billion parameters)[4] and GPT-J (6 billion parameters)[27] publicly available. A subset of samples was generated using all the three models. A comparison of the synthesized samples revealed that the quality of those produced by GPT-3 was far superior to those of GPT-Neo and GPT-J.

3.3 Multimodal approaches

3.3.1 Multi-modal dataset

While the original dataset contains over 1 million samples, only about 64% i.e about 564,000 samples contain image data associated with the post. The images in the data is given in the form of a URL, which needs to be fetched. After studying various text processing models we found that BERT is ideal for our purpose as it considers contextual embedding[9] and choose ResNet50 as it generally performs better than other image classification models[19].

3.3.2 Text and Image preprocessing

- **Text Processing:** Firstly, we examined the length of the titles for all the samples in the data and found that 75% of the titles have a length of fewer than 57 words. So in the case of text processing, we used BERT Transformer [6] with the hyperparameter `max_length` set to 128 i.e shorter sentences are padded to length 128 and longer sentences are truncated to length 128. Before applying the transformer network we perform additional text processing steps[24]. Those include tokenization(a method of dividing a piece of text into smaller components. This cleaned sentence is now passed to the BERT transformer network which returns a list of 128 numbers that represent the sentence.

- **Image Processing** To extract features from images we planned to use a truncated version of an existing image recognition network like ResNet50 [10]. First, we needed to extract the images from the URL for this purpose we used the requests library in python. While extracting the images we filtered out all the images that don't contain color data since most image recognition networks accept images in the form of colored data. To make the input size consistent we resized the extracted images to 224x224. We then passed these processed images to stage 1 of ResNet50 to extract feature vectors of size 1000x1 for individual images.

3.3.3 Multimodal flow and architectures

The outputs of the BERT transformer network and the ResNet50 network are passed through two different trainable dense layers with 128 nodes each. These two feature vectors of size 128 can be aggregated using 3 different methods i.e add, maximum and average. In the 'add' method, as the name suggests we add ith element of the text feature vector with ith element of the image feature vector. In the 'maximum' method, we take the maximum of two ith elements in both feature vectors. In the 'average', we take the average of the feature vectors. The three methods result in a output vector of size 128x1. This is then passed through a Dense model with 4 dense layers each with size 512, 256, 64, 1 respectively. The first three layers were relu activation functions and the last one uses sigmoid activation function. All the layers are regularized with dropout regularization to prevent overfitting. The output of this dense model represents whether the given sample is real or fake. It is then trained on the binary crossentropy loss function with gradient descent having adam optimization.

4 Results

Table 1 compares the performance of the models utilizing only text based classification models on both the augmented and original test data.

Table 2 lists the performance results obtained using the multimodal approach to text classification. The first column denotes the model used to obtain textual features before concatenation with image features obtained from the model mentioned in the second column.

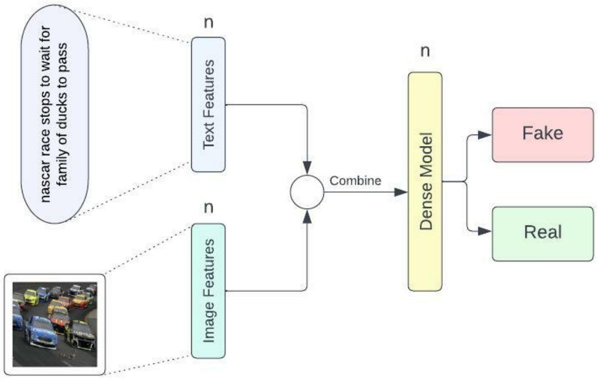

Figure 4: Multimodal Architecture

We have chosen accuracy as the metric for comparison since the split between true and fake samples is extremely close to 50% and hence, accuracy is a good metric to compare the models. As is evident from the tables, augmentation does result in better accuracy but, only by a small amount since we didn't augment many samples (only about 5%). This is because augmentation took a long time and OpenAI has a limit on the amount of API requests a single account can make for free. We believe that a higher amount of augmentation with more resources definitely has the potential to result in a greater increase in accuracy. The multimodal approaches result in a substantial increase in accuracy as was expected since a single training sample now has more information than the unimodal approach. Another point of interest is the accuracy of Random Forest, which is the least of all the methods, which is a bit surprising since we would expect it to be at least as good as the simple models such as Naive Bayes and Logistic Regression. One potential reason for its performance could be a non-optimal choice of hyperparameters which can be solved by increasing the size of the space.

5 Related Work

Deep learning techniques have a wide range of uses since neural networks were revived in the sec-

Accuracy		
Model	with Augmentation	without Augmentation
Naive Bayes	79.6%	79.2%
Logistic Regression	82.9%	82.4%
Random Forest	69.9%	69.5%
LinearSVM	82.5%	82.1%
CNN	85.8%	85.3%

Table 1: Unimodal Approaches

Textual features	Image features	Accuracy
CNN	CNN	87.2%
BERT	ResNet-50	89.4%

Table 2: Multimodal Approaches

ond decade of the twenty-first century.Computer vision and Natural Language Processing (NLP) advancements include deep neural network approaches [18][5]. The majority of these works have relied solely on text-based, unimodal approaches. There have also been attempts at more ambitious architectures that combine different types of data (like text and image) [1][3][30][26][8].

5.1 Multimodal methods for detecting fake news

We examine the most recent research on identifying fake news solely based on its textual content. Kaliyar et al. [12] propose the DeepNet model for a binary classification of fake news. Seven dense layers, one LSTM layer, three convolutional layers, one embedding layer, ReLU for activation, and finally the softmax function for binary classification make up this model. The Fakeddit and BuzzFeed (Kaggle) datasets are used to evaluate the model. The models' binary classification accuracy on the Fakeddit dataset was 86.4%, and their accuracy on the BuzzFeed dataset was 95.2%.

Kirchknopf et al. [13] perform binary classification of fake news using the Fakeddit dataset and four different data modalities. More specifically, the authors make use of the news's textual content, the comments that go along with it, the images, and the remaining metadata from other modalities. 95.5% accuracy is the highest measurement. For binary fake news detection, Li et al. [16] suggest using the Entity-Oriented MultiModal Alignment and Fusion Network (EMAF). The accuracy on the Fakeddit, Weibo, and Twitter datasets was 92.3%, 97.4%, and 80.5%, respectively.

6 Conclusion

We presented a comparison across different approaches for the detection of fake news. Performance improvement can be achieved using data augmentation techniques like back-translation and paraphrasing using GPT-3. Although the increase in performance using data augmentation isn't apparent since the improvement in performance could have been due to randomness, we believe that augmentation using a larger number of samples will definitely result in an unquestionable increase in accuracy. Using a multimodal approach increases the accuracy significantly. We found that using data augmentation could result in better performance and that the multimodal approach performs better than the augmentation approach. Hence, as of now, it is worth the effort of going the extra mile to collect and label image data for the classification. In the future, we could work on augmenting more samples and increasing the accuracy so that we can confidently confirm our hypothesis that augmentation results in better performance. We could also look to experiment with the architectures used for the textual and image features in the multimodal approach.

References

[1] Mahdi Abavisani, Liwei Wu, Shengli Hu, Joel R. Tetreault, and Alejandro Jaimes. Multimodal categorization of crisis events in social media. *CoRR*, abs/2004.04917, 2020.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[3] Kang Il Bae, Junghoon Park, Jongga Lee, Yungseop Lee, and Changwon Lim. Flower classification with modified multimodal convolutional neural networks. *Expert Systems with Applications*, 159:113455, 2020.

[4] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021.

[5] Li Deng and Yang Liu. *Deep Learning in Natural Language Processing*. Springer, 2018.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[7] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[8] António Gaspar and Luís A. Alexandre. A multimodal approach to image sentiment analysis. In Hujun Yin, David Camacho, Peter Tino, Antonio J. Tallón-Ballesteros, Ronaldo Menezes, and Richard Allmendinger, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2019*, pages 302–309, Cham, 2019. Springer International Publishing.

[9] Santiago González-Carvajal and Eduardo C. Garrido-Merchán. Comparing bert against traditional machine learning text classification. 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[11] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[12] Rohit Kumar Kaliyar, Pawan Kumar, Manish Kumar, Meenal Narkhede, Sreyas Namboodiri, and Sneha Mishra. Deepnet: An efficient neural network for fake news detection using news-user engagements. In *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pages 1–6, 2020.

[13] Armin Kirchknopf, Djordje Slijepcevic, and Matthias Zeppelzauer. Multimodal detection of information disorder from social media. *CoRR*, abs/2105.15165, 2021.

[14] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *CoRR*, abs/1805.06201, 2018.

[15] Philippe Laban, Lucas Bandarkar, and Marti A Hearst. News headline grouping as a challenging nlu task. In *NAACL 2021*. Association for Computational Linguistics, 2021.

[16] Peiguang Li, Xian Sun, Hongfeng Yu, Yu Tian, Fanglong Yao, and Guangluan Xu. Entity-oriented multi-modal alignment and fusion network for fake news detection. *IEEE Transactions on Multimedia*, 24:3455–3468, 2022.

[17] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028, 2002.

[18] Niall O′ Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Adolfo Velasco-Hernández, Lenka Kr-

palkova, Daniel Riordan, and Joseph Walsh. Deep learning vs. traditional computer vision. *CoRR*, abs/1910.13796, 2019.

[19] Sheldon Mascarenhas and Mukul Agarwal. A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification. 1:96–99, 2021.

[20] Kai Nakamura, Sharon Levy, and William Yang Wang. r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. *arXiv preprint arXiv:1911.03854*, 2019.

[21] 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.

[22] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[23] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709, 2015.

[24] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? pages 194–206, 2019.

[25] Tf-idf. Tf-idf — Wikipedia, the free encyclopedia, 2022. [Online; accessed 7-December-2022].

[26] Matheus Viana, Quoc-Bao Nguyen, John Smith, and Maria Gabrani. Multimodal classification of document embedded images. In Alicia Fornés and Bart Lamiroy, editors, *Graphics Recognition. Current Trends and Evolutions*, pages 45–53, Cham, 2018. Springer International Publishing.

[27] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

[28] Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196, 2019.

[29] Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. Gpt3mix: Leveraging large-scale language models for text augmentation. *CoRR*, abs/2104.08826, 2021.

[30] Jianfei Yu and Jing Jiang. Adapting bert for target-oriented multimodal sentiment classification. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5408–5414. International Joint Conferences on Artificial Intelligence Organization, 7 2019.