Детекция эмоций. Сравнение и анализ классических методов машинного обучения и методов обучения с трансформерами

A Preprint

Панин Никита Александрович
Факультет вычислительной математики и кибернетики
МГУ им. Ломоносова
s02200456@gse.cs.msu.ru

д.ф-м.н., профессор, Воронцов Константин Вячеславович
Факультет вычислительной математики и кибернетики
МГУ им. Ломоносова
vokov@forecsys.ru

Аннотация

В работе была рассмотрена задача детекции эмоций на датасете, в основу которого вошел WASSA датасет из твитов для детекции эмоций. На выходе алгоритма классификации эмоций в твитах была одна из 5 эмоций: нейтральная эмоция, грусть, страх, радость, гнев. Были применены различные методы "классического"машинного обучения, такие как, SVM, логистическая регрессия, метод k-ближайших соседей и наивный байесовский классификатор. Также классификация эмоций была проведена с помощью файн-тюнинга нескольких версий BERT. Основной целью работы являлось проведение сравнительного анализа для классических моделей машинного обучения(wKNN, Multinomial Bayes Classifier, Logistic Regression, SVM) и для моделей глубокого обучения (в качестве предобученной модели брались BERT, RoBERTa, BERTweet и их large-версии). В результате исследования было показано, что по метрике accuracy для моделей классического обучения с tf-idf векторизацией текстов лучше всего работает SVM с RBF ядром (accuracy 0.8387 на тесте), а наиболее качественные результаты получаются с помощью предложенной в исследовании модели с предобученным BERTweet (accuracy 0.88 на тесте).

Ключевые слова Детекция эмоций · NLP

1 Введение

В нашем эмоционально насыщенном мире разумно стремиться к пониманию тонких чувств и настроений, выраженных в текстах. С развитием машинного обучения и искусственного интеллекта, возникает возможность автоматизировать и усовершенствовать процессы анализа и интерпретации текстов, выявляя эмоции, ассоциированные с ними. Детекция эмоций в текстах стала актуальным направлением исследований в области обработки естественного языка (NLP) и анализа тональности и привлекает все большее внимание в последние два десятилетия [25, 9]. Термины "распознавание эмоций"(emotion detection) и "анализ тональности"(sentiment analysis) часто используются как взаимозаменяемые, хотя между этими двумя понятиями существуют очевидные различия [1]. Анализ тональности в основном измеряет субъективное отношение с точки зрения полярности настроения: нейтральное, положительное, негативное. Выявление эмоций предполагает идентификацию более детальных эмоциональных состояний, например, счастье, гнев, страх, удивление.

Эмоции имеют множество применений в разных сферах. В маркетинге анализ предпочтений потребителей помогает улучшить бизнес-стратегии [2]. В социальных сетях распознавание агрессивных эмоций помогает выявить потенциальных преступников или террористов [4]. Мониторинг эмоций в реальном времени на основе данных социальных сетей может помочь в профилактике самоубийств [13]. Определение эмоций во время кризисов или катастроф позволяет понять чувства людей по отношению к конкретной ситуации, что способствует управлению в кризис и принятию важных решений [32].

1.1 Постановка задачи

В данной работе была рассмотрена задача детекции эмоций на датасете, в основу которого вошел WASSA датасет из твитов для детекции эмоций [21]. На выходе алгоритма классификации эмоций в твитах была одна из 5 эмоций: нейтральная эмоция, грусть, страх, радость, гнев. Были применены различные методы "классического" машинного обучения, такие как, SVM, логистическая регрессия, метод k-ближайших соседей и наивный байесовский классификатор. Также классификация эмоций была проведена с помощью файн-тюнинга нескольких версий BERT. За основу бралась статья [17]. Основной целью работы являлось проведение сравнительного анализа полученных моделей. Также в работе предлагается сравнение новых моделей, таких как large-версии бертов, логистическая регрессия, wKNN, SVM с другими ядрами, с результатами из основной статьи.

1.2 Существующие решения

Ванг и др. [27] создали большой набор данных твитов, используя хэштеги эмоций. Они применили два различных классификатора, логистическую регрессию и Naïve Bayes, чтобы исследовать эффективность различных признаков, таких как n-граммы, лексикон эмоций и информация о части речи, для задачи идентификации эмоций. Наибольшая достигнутая точность составила 0,6557.

Мохаммад [20] создал корпус твитов, маркированных эмоциями, используя хэштеги. Он применил бинарные SVM, по одному для каждой из шести основных эмоций Экмана [12], и использовал наличие или отсутствие униграмм и биграмм в качестве бинарных признаков. Бинарные классификаторы смогли предсказать эмоции со сбалансированным F1-score 0,499.

Янссенс и др. [22] исследовали влияние использования слабых меток по сравнению с сильными метками на распознавание эмоций для корпуса, состоящего из 341 931 твита. Слабые метки были созданы путем использования хэштегов твитов, а сильные метки - с помощью краудсорсинга. Характеристики, извлеченные путем объединения n-грамм и TF-IDF (Term Frequency-Inverse Document Frequency), были применены к пяти алгоритмам классификации: Стохастический градиентный спуск, SVM, Naïve Bayes, Nearest Centroid и Ridge. Результаты показали снижение F1-score на 9,25% при использовании слабых меток.

К сожалению, классические методы машинного обучения не могут учесть последовательную природу текста, поэтому некоторые модели глубокого обучения, такие как рекуррентные нейронные сети (RNN), LSTM [15] и GRU [6], стали более перспективными в определении эмоций в тексте [19, 3, 29]. Хотя рекуррентные модели принимают во внимание последовательный характер текста и показывают передовые результаты для различных задач NLP, они обладают некоторыми слабостями: медленная скорость, необходимость обучения с нуля и ограниченная способность улавливать долгосрочные зависимости в тексте [15]. Также требуется большой объем размеченных данных для обучения. Подготовка большого объема размеченных данных является трудоемкой и дорогостоящей процедурой, и именно здесь вступает в игру перенос обучения(transfer learning). С его помощью можно добиться лучших результатов по сравнению с традиционными моделями глубокого обучения с гораздо меньшим количеством обучающей выборки. Предварительно обученные языковые модели, такие как BERT [10] (Bidirectional Encoder Representations from Transformers) и его варианты, OpenAI GPT (Generative Pre-trained Transformer) [24] и Transformer-XL [8], получили широкое распространение в различных задачах NLP и продемонстрировали впечатляющие результаты.

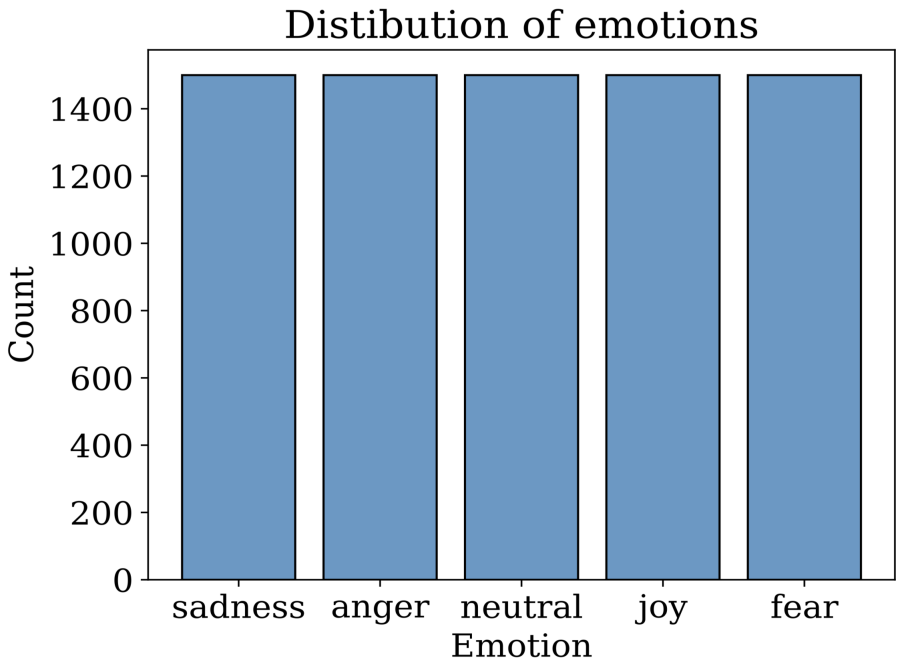
Некоторые работы используют предварительно обученные языковые модели для классификации эмоций или тональности в тексте. Например, исследование [23] использует BERT в качестве слоя эмбеддинга, после которого выводы передаются через слои CNN и BiLSTM для анализа настроений на бенгальском языке.

2 Описание данных

Данные[1](см. Рис. 1), используемые для обучения и оценки эмоций в твитах, были частично получены из набора данных WASSA, представленного участникам на семинаре по компьютерным методам анализа субъективности, настроения и социальных сетей (WASSA-2017) [21]. Было выбрано по 1500 твитов для каждой из четырех эмоций: страх, грусть, радость и гнев, при этом информация об интесивности эмоций из датасета была исключена. К этим четырем категориям был добавлен еще один класс из 1500 нейтральных твитов, так как в исследовании[18] было показано, что наличие нейтрального класса важно для классификации, поскольку это позволяет модели не относить неизвестные эмоции к одному из изучаемых классов. Нейтральные твиты были взяты с CrowdFlower. Данные были разделены таким образом, чтобы обеспечить хороший баланс между классами [26]. Сбалансированный набор данных содержит одинаковое количество примеров для каждого класса, что гарантирует, что модель не будет уделять больше внимания крупным классам при классификации.

Для обучения бралось 80% данных, для теста и валидации по 10%.

Рис. 1:



3 Методология

3.1 Метрики

Поскольку используемый датасет сбалансирован по классам, то в качестве основной метрики была взята доля правильных классификаций(accuracy), которая плохо работает в случае дисбаланса классов. Также сравнения велись по точности(Precision), полноте(Recall), F1 мере(F1) с макроусреднением.

Для каждого класса $y \in Y$:

- TP_y – верные положительные
- FP_y – ложные положительные
- FN_y – ложные отрицательные

[1]Данные были взяты с:
https://github.com/alexalbu98/Emotion-Detection-From-Tweets-Using-BERT-and-SVM-Ensemble-Model/blob/master/dataset.zip

Точность, полнота и F1 мера с макроусреднением:

$$Precision_{macro} = \frac{1}{|Y|}\sum_{y}\frac{TP_y}{TP_y + FP_y}$$

$$Recall_{macro} = \frac{1}{|Y|}\sum_{y}\frac{TP_y}{TP_y + FN_y}$$

$$F1_{macro} = \frac{2 * Precision_{macro} * Recall_{macro}}{Precision_{macro} + Recall_{macro}}$$

В дальнейшем изложении пометка "macro" будет опускаться.

3.2 Классическое машинное обучение

3.2.1 Предобработка текстов

"Сырые" твиты в большинстве своем это "грязные", сильно зашумленные данные. Это сопряжено прежде всего с природой твитов: люди часто пишут сокращениями, с ошибками и прочее. При предобработке текстов использовались стандартные методы в машинном обучении для подготовки текстовых данных на вход алгоритму:

- Эмотикон
  Поскольку через текст часто сложно передать эмоции, эмотиконы приобрели очень большую популярность и пользователи нередко используют их в своих твитах. В связи с этим эмотиконы были переведены в текстовый формат с помощью библиотеки Demoji Python, чтобы они помогали в идентификации эмоций в твитах [28].
- Хэштег
  Хэштеги - это фразы без пробелов с префиксом в виде символа решетки (#), которые часто используются пользователями для упоминания трендовой темы в Твиттере. Все хэштеги были заменены словами без символа решетки. Например, #hello заменяется на hello.
- Упоминание пользователей
  У каждого пользователя Твиттера есть имя пользователя, связанное с ним. Пользователи часто упоминают других пользователей в своих твитах через @username. В текстах все упоминания пользователей убирались.
- URL
  Пользователи часто делятся гиперссылками на другие веб-страницы в своих твитах. Какой-либо конкретный URL-адрес не важен для классификации текста и , если бы ссылки были оставлены, то это привело бы к очень разреженным признакам для текстов причем напрасно. Именно поэтому все URL-адреса в твитах были удалены.
- Фильтрация стоп-слов и стемминг
  Эмпирически доказано, что фильтрация стоп-слов в наборе данных повышает производительность и скорость вычислений [14], поэтому стоп-слова удалялись. Стемминг также использовался и это еще один метод повышения производительности модели путем сведения производных слов к грамматическому корню, называемому "stem".

3.2.2 Векторизация текстов

После предобработки тексты векторизовались с помощью наиболее популярного векторного представления – TF-IDF. Пусть d - документ из коллекции документов D, w_i - i-е слово из словаря W. Мощность (количество элементов) для множеств , как это принято в математике, будем обозначать $|\cdot|$. Тогда векторное представление каждого документа будет выглядеть так:

$$d_{vec} = \begin{pmatrix} tf-idf(w_1, d, D) \\ tf-idf(w_2, d, D) \\ \vdots \\ tf-idf(w_{|W|}, d, D) \end{pmatrix},$$

где

$$tf-idf(w_i, d, D) = \underbrace{tf(w_i, d)}_{term\ frequency} * \underbrace{idf(w_i, D)}_{inverse\ document\ frequency} =$$

$$= \underbrace{\frac{\sum_{w' \in d}[w_i = w']^2}{\sum_{w' \in d} 1}}_{tf(w_i,d)} * \underbrace{log\left(\frac{|D|}{\sum_{d \in D}[w_i \in d]}\right)}_{idf(w_i,D)}$$

Такие веса используются из предположения, что, чем больше документов, в которых встречается слово, тем меньше смысла оно несет в себе и следовательно его вес меньше.

3.2.3 Модели

1. wKNN
   Вместо классического KNN использовался взвешенный KNN (weighted KNN, wKNN) [11], в котором веса не равномерно распределены для ближайших соседей объекта, а обратно пропорционально от расстояний от объекта до ближайших соседей. Основными параметрами для этого метрического метода являются: расстояние между текстами и k ближайших соседей.
   В качестве метрики в метрических алгоритмах берут функции убывающие от "близости", поэтому будем отталкиваться именно от "близости" текстов. Согласно Хуанг [16], которая сравнивала различные меры близости для кластеризации текстов, лучшая оказалась косинусная мера:
   $$cosine\_similarity(x, y) = \frac{x^T y}{\|x\|\|y\|},$$
   где x, y - векторные представления текстов, а $\|\cdot\|$ - норма вектора. В качестве косинусного расстояния бралась функция:
   $$cosine\_distance(x, y) = 1 - cosine\_similarity(x, y)$$
   При подборе k можно было воспользоваться эмпирическим правилом и брать $k = \frac{\sqrt{|D|}}{2} = 17$(в нашем случае), однако, поскольку это лишь эмпирическое правило, было решено попробовать перебрать k и оказалось, что лучшим выбором было $k = 52$
2. Logistic Regression
   Логистическая регрессия - это алгоритм машинного обучения с учителем, используемый для задач бинарной или многоклассовой классификации. В контексте машинного обучения он используется для прогнозирования вероятности принадлежности объекта к определенному классу. Существует несколько подходов для использования logistic regression: либо использовать множество бинарных логистических классификаторов и на основе их выдавать ответ (one-vs-one или one-vs-rest) или изменить лосс функцию и на выходе вместо сигмоиды для предсказания вероятностей использовать софтмакс. В данной работе использовался второй подход. Формализуем задачу многоклассовой логистической регрессии.

   Линейный классификатор при произвольном числе классов $|Y|$
   $$a(x) = \arg\max_{y \in Y} \langle w_y, x \rangle, \ x, w_y \in \mathbb{R},$$
   где $\langle \cdot, \cdot \rangle$ - скалярное произведение, w_y - веса для метки $y \in Y$.
   Вероятность того, что объект x относится к классу y:
   $$P(y|x, w) = \frac{exp(\langle w_y, x \rangle)}{\sum_{z \in Y}(\langle w_z, x \rangle)} = \underset{y \in Y}{\text{SoftMax}}(\langle w_y, x \rangle)$$
   Обучение состоит в максимизации правдоподобия (log-loss) с регулизацией:
   $$L(w) = \sum_{i=1}^{l} log(P(y_i|x_i, w) - \frac{\tau}{2} \sum_{y \in Y} \|w_y\|^2 \to \max_w$$
   Для τ было подобрано значение равное 1.
3. Multinomial Naive Bayes
   Мультиномиальный байесовский классификатор (Multinomial Naive Bayes Classifier) является

[2]Используется нотация Айверсона

разновидностью наивного байесовского классификатора, который использует мультиномиальное распределение для моделирования данных. Этот алгоритм особенно хорошо подходит для решения задач классификации текста и анализа тональности, когда требуется разделить текстовые данные на несколько категорий. Несмотря на то, что классическая теория мультиномиального классификатора предполагает, что на вход подается векторизованные документы по частотам слов в каждом документе, существуют работы, в которых опытным путем было показано, что tf-idf дает лучше результаты [5]. По этой причине в работе был выбран tf-idf для мультиномиального байесовского классификатора.

4. Support Vector Classifier
   Метод опорных векторов состоит в поиске оптимальной разделяющей гиперплоскости, которая приводит к максимизации ширины разделяющей полосы между классами, следовательно, к более уверенной классификации. SVC работает для бинарной классификации, поэтому в случае мультиклассовой классификации приходится строить несколько алгоритмов SVC и выбирать класс с помощью голосования. В качестве подхода построения таких алгоритмов была выбрана схема one-vs-one, поскольку при использовании one-vs-rest SVC придется обучать на большой выборке данных, что вычислительно менее эффективно, так как на больших данных SVC ресурсоемкий и будет долго обучаться.

3.3 Трансформеры

3.3.1 Предобработка и токенизация текстов

Для всех рассматриваемых версий BERT существуют свои специфичные токенизаторы. Перед подачей в токенизатор предложений[3] в каждом были убраны ссылки и имена пользователей, поскольку они не несут никакой информации об эмоциях. Несмотря на разницу в токенизаторах, отметим общие принципы:

- Нормализация, т.е. приведение к нормальному виду, что обычно делается путем приведения слов к нижнему регистру, контролирования специальных символов, в том числе, удаление пробельных символов.
- Токенизация, т.е. отображение слов в токены. Для BERT - WordPiece [31], для BERTweet и RoBERTa - byte-level BPE(Byte-Pair-Encoding) [30].
- Добавление специальных токенов, такик как '[CLS]', с помощью которого агрегируется информацию о предложении, используемая для задач классификации, а также '[PAD]', '[SEP]', '[EOS]'.

3.3.2 Модели

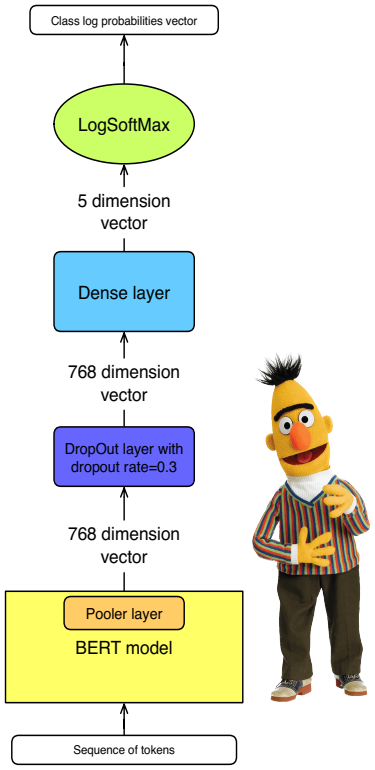
В работе рассмотрены три основные модели: BERT [10], RoBERTa [30], BERTweet [7], а также их large-аналоги, в которых больше слоев и больше обучаемых параметров. Для решения задачи классификации брался эмбеддинг, чья размерность 768, который соответствует позиции '[CLS]' токена в токенизированном предложении. Этот эмбеддинг проходил через dropout-слой с вероятностью обнуления элемента равной 0.3, чтобы снизить переобучение, далее через линейный слой и LogSoftMax, поскольку эта функция активации вычислительно более стабильна чем SoftMax (см. Рис. 2). Выбранной функцией потерь была NLLLoss(Negative Log Likelihood loss):

$$NLLLoss(y) = -log(y)$$

В качестве оптимизатора был выбран Adam. При обучении использовалось линейное уменьшение шага градиента и ограничение нормы градиента с максимальной нормой, равной 1, чтобы уменьшить вероятность появления исчезающего или взрывающегося градиента.

[3]Под "предложением" будем понимать кусок текста, а не синтаксическое предложение

Рис. 2: Предложенная модель



Список литературы

[1] A. G. Shahraki A. Yadollahi and O. R. Zaiane. Current state of text sentiment analysis from opinion to emotion mining. In ACM Comput. Surv., vol. 50, no. 2, page 1–33.

[2] E. Cambria. Affective computing and sentiment analysis. In IEEE Intell. Syst., vol. 31, no. 2, pages 102–107, Mar./Apr. 2016.

[3] Chinnakotla MK Srikanth R Galley M Agrawal P Chatterjee A, Gupta U. nderstanding emotions in text using deep learning and big data. In Comput Human Behav 93, page 309–317, 2019.

[4] M. Cheong and V. C. S. Lee. A microblogging-based approach to terrorism informatics: Exploration and chronicling civilian sentiment and response to terrorism events via twitter. In Information Systems Frontiers, vol. 13, no. 1, Mar. 2011.

[5] Chingmuankim and Prof.Rajni Jindal. A comparative study of naive bayes classifierswith improved techniqueon text classification.

[6] Gulcehre C Bahdanau D Bougares F Schwenk H Bengio Y Cho K, Van Merriënboer B. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In Conference on Empirical Methods in Natural Language Processingf. EMNLP, 2014.

[7] T. Vu D. Q. Nguyen and A. T. Nguyen. Bertweet: A pre-trained language model for english tweets. In In Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 9–14, Nov. 2020.

[8] Yang Y Carbonell J Le QV Salakhutdinov R Dai Z, Yang Z. Transformer-xl: Attentive language models beyond a fixed-length context. In Associat Comput Linguist, 2019.

[9] Jiawen Deng and Fuji Ren. A survey of textual emotion recognition and its challenges. 2021.

[10] Lee K Toutanova K Devlin J, Chang M. Bert: pre-training of deep bidirectional transformers for language understanding. In Association for Computational Linguistics, 2018.

[11] Sahibsingh A Dudani. The distance-weighted k-nearest-neighbor rule. In IEEE Transactions on Systems, Man, and Cybernetics, pages (4):325–327, 1976.

[12] P. Ekman. An argument for basic emotions, cognition and emotion. pages 169–200, 1992.

[13] X. Kang F. Ren and C. Quan. Examining accumulated emoional traits in suicide blogs with an emotion topic model. In IEEE J. Biomed. Health Inform., vol. 20, no. 5, Sep. 2016.

[14] Y. He H. Saif, M. Fernandez and H. Alani. On stopwords, filtering and data sparsity for sentiment analysis of twitter. In In Proc. of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, pages 810–817, 2014.

[15] Schmidhuber J Hochreiter S. Long short-term memory. Neural Comput 9(8), pages 1735–1780, 1997.

[16] Anna Huang. Similarity measures for text document clustering. In In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, pages volume 4, pages 9–56, 2008.

[17] Stelian SPINU Ionut-Alexandru ALBU. Emotion detection from tweets using a bert and svm ensemble model. 2022.

[18] M. Koppel and J. Schler. The importance of neutral examples for learning sentiment. In Computational Intelligence, vol. 22, no. 2, pages 100–109, 2006.

[19] Kraus M Feuerriegel S Prendinger H Kratzwald B, Ilić S. Deep learning for affective computing: Text-based emotion recognition in decision support. In Decision Support Syst 115, page 24–35, 2018.

[20] S. M. Mohammad. Emotional tweets. In Proc. of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 246–255, 2012.

[21] S. M. Mohammad and F. Bravo-Marquez. Wassa-2017 shared task on emotion intensity. In Proc. of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Copenhagen, Denmark, pages 34–49.

[22] E. Mannens S. Van Hoecke O. Janssens, S. Verstockt and R. Van de Walle. Influence of weak labels for emotion recognition of tweets. In In Prasath R., O'Reilly P., Kathirvalavakumar T. (Eds.), Mining Intelligence and Knowledge Exploration, Springer, pages 108–118, 2014.

[23] Kowsher M Murad SA Bairagi AK Masud M Baz M Prottasha NJ, Sami AA. Transfer learning for sentiment analysis using bert based supervised fine-tuning. 2022.

[24] Salimans T Sutskever I Radford A, Narasimhan K. Improving language understanding by generative pre-training. 2018.

[25] C. Strapparava and R. Mihalcea. Affect detection in texts 13. In The Oxford Handbook of Affective Computing, page 184–216, 2015.

[26] P. Kordik T. Borovicka, M. Jirina Jr. and M. Jirina. Selecting representative data sets. In In Karahoca A. (Ed.), Advances in Data Mining Knowledge Discovery and Applications, IntechOpen, pages 43–70, 2012.

[27] K. Thirunarayan W. Wang, L. Chen and A. P. Sheth. Harnessing twitter 'big data' for automatic emotion identification. In Proc. of the 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing, pages 587–592, 2012.

[28] W. Wolny. Emotion analysis of twitter data that use emoticons and emoji ideograms. In Information Systems Development: Complexity in Information Systems Development (ISD2016 Proceedings), Katowice, Poland, pages 476–483, 2016.

[29] Liu J Xu G, Li W. A social emotion classification approach using multi-model fusion. In Future Generat Comput Syst 102, pages 347–356, 2020.

[30] Naman Goyal Jingfei Du Mandar Joshi Danqi Chen Omer Levy Mike Lewis Luke Zettlemoyer† Veselin Stoyanov Yinhan Liu, Myle Ott. "roberta: A robustly optimized bert pretraining approach".

[31] Zhifeng Chen Quoc V. Le Mohammad Norouzi Yonghui Wu, Mike Schuster. Google's neural machine translation system: Bridging the gap between human and machine translation. 2016.

[32] A. Ekbal Z. Ahmad, R. Jindal and P. Bhattachharyya. Borrow from rich cousin: Transfer learning for emotion detection using cross lingual embedding. In Expert Syst. Appl., vol. 139, Jan. 2020.