2023 Data Mining Lab 2 — Emotion Recognition on Twitter — Report

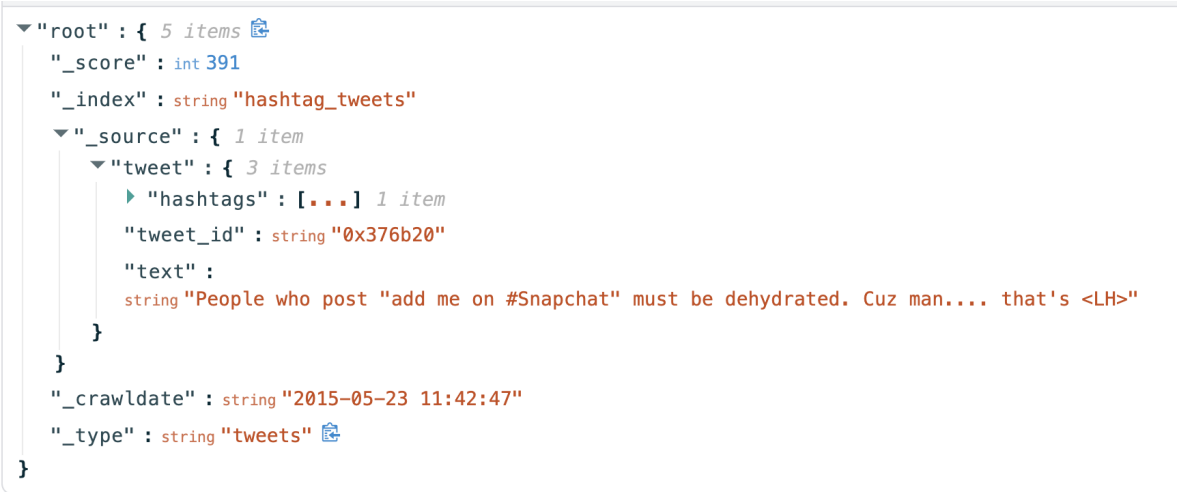
1. Introduction

Delving into the dynamic landscape of emotion recognition on Twitter, this article explores the application of machine learning techniques to predict and understand sentiments expressed in short-form text. Focused on deciphering the rich tapestry of emotions within the diverse Twitter community, our exploration delves into methodologies, challenges, and potential applications, offering insights into the significance of harnessing artificial intelligence for emotion analysis in the digital era.

2. Related work

a. Exploration Data Analysis:

First, i observe the original Json data Schema:



I can decide which columns might be useful and which are not.To use these columns , we have to flatten the json data to table data just like dataframe.

	_score	hashtags	tweet_id	text
0	391	Snapchat	0×376b20	People who post "add me on #Snapchat" must be ...
1	433	freepress TrumpLegacy CNN	0×2d5350	@brianklaas As we see, Trump is dangerous to #...
2	232	bibleverse	0×28b412	Confident of your obedience, I write to you, k...
3	376		0×1cd5b0	Now ISSA is stalking Tasha 😂😂😂 <LH>
4	989		0×2de201	"Trust is not the same as faith. A friend is s...
...	...	...	...	...
1867530	827	mixedfeeling butimTHATperson	0×316b80	When you buy the last 2 tickets remaining for ...
1867531	368		0×29d0cb	I swear all this hard work gone pay off one da...
1867532	498		0×2a6a4f	@Parcel2Go no card left when I wasn't in so I ...
1867533	840		0×24faed	Ah, corporate life, where you can date <LH> us...
1867534	360	Sundayvibes	0×34be8c	Blessed to be living #Sundayvibes <LH>

Second, i decide to use the “hashtags” and “text” column to predicted tweets emotion.

b. Feature Engineering:

i. Merge the hashtags and text column to create a new feat column named “new_feat”

```
tweets_df["new_feat"] = tweets_df["text"] + " " + tweets_df["hashtags"]
```

ii. Commencing feature engineering based on the new column 'new_feat.':

```
def remove_punctuation(text):
    return text.translate(str.maketrans('', '', string.punctuation))

def remove_stopWord(text):
    stop_words = set(stopwords.words('english'))
    words = word_tokenize(text)
    text = ' '.join([word for word in words if word.lower() not in stop_words])
    return text

def remove_html_tag(text):
    p = re.compile(r'<.*?>')
    return p.sub(r'', text)

def remove_url(text):
    return re.sub(r'http\S+', '', text, flags=re.MULTILINE)

def remove_extra_whiteSpace(text):
    text = ' '.join(text.split())
    return text

def text_stemming(text):
    ps = PorterStemmer()
    words = word_tokenize(text)
    text = ' '.join([ps.stem(word) for word in words])
    return text

def handel_contraction(text):
    for contraction, expansion in contractions_dict.items():
        text = text.replace(contraction, expansion)
    return text
```

i use seven method to do the text data cleaning,

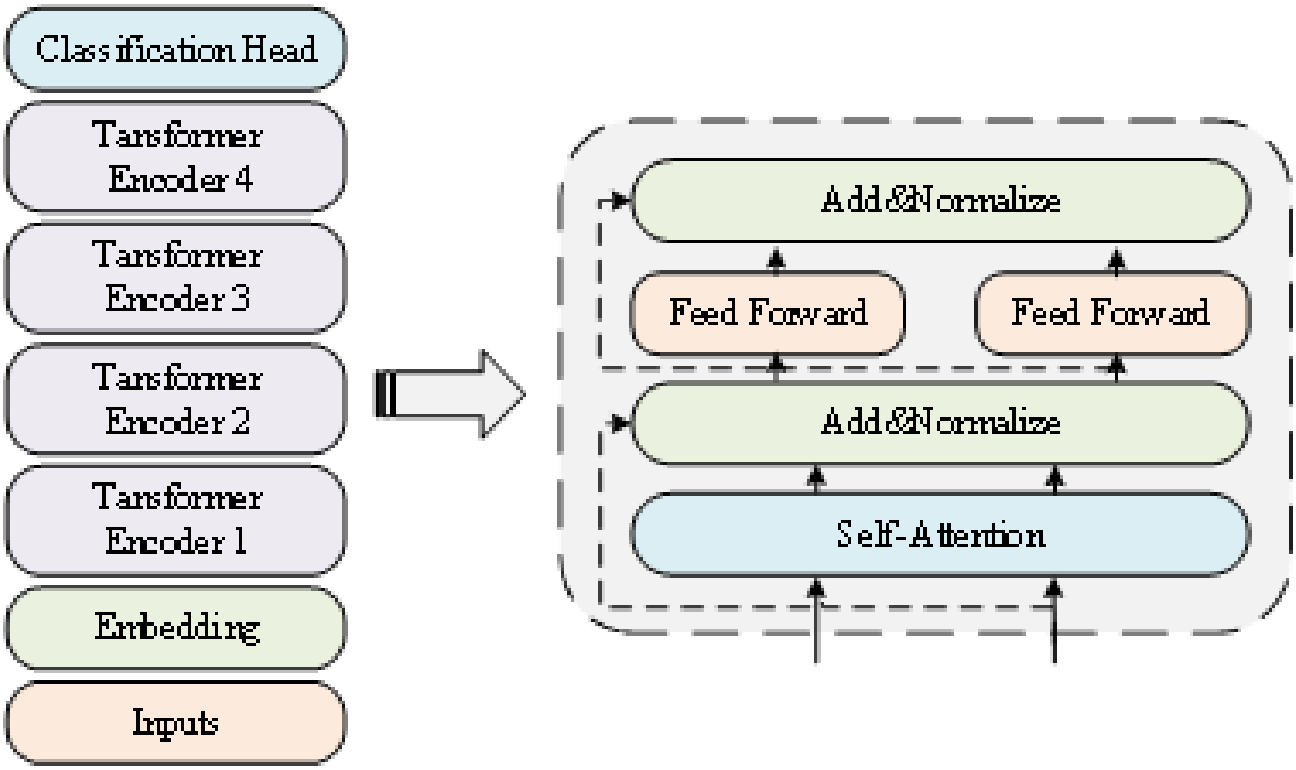
1. Convert all text to lowercase. This helps in treating words in a case-insensitive manner.
2. Remove unnecessary punctuation, as it usually doesn't carry much meaning in text analysis.
3. Stopwords are common words like "the," "is," and "and" that don't carry much meaning. Remove them to focus on the significant words.

4. If your text data contains HTML tags, remove them using a library like BeautifulSoup.
5. Remove extra whitespaces to standardize the text.
6. Reduce words to their base or root form to standardize them. This can be done using libraries like NLTK or spaCy.
7. Expand contractions for a consistent representation of words.

3. Methodology:
   a. Preprocessing:
   The Twitter data underwent extensive preprocessing to clean and tokenize the text. Special attention was given to handle user mentions, hashtags, and URLs appropriately. We leveraged the capabilities of the TinyBERT tokenizer to encode the tweets into suitable input representations for the model.

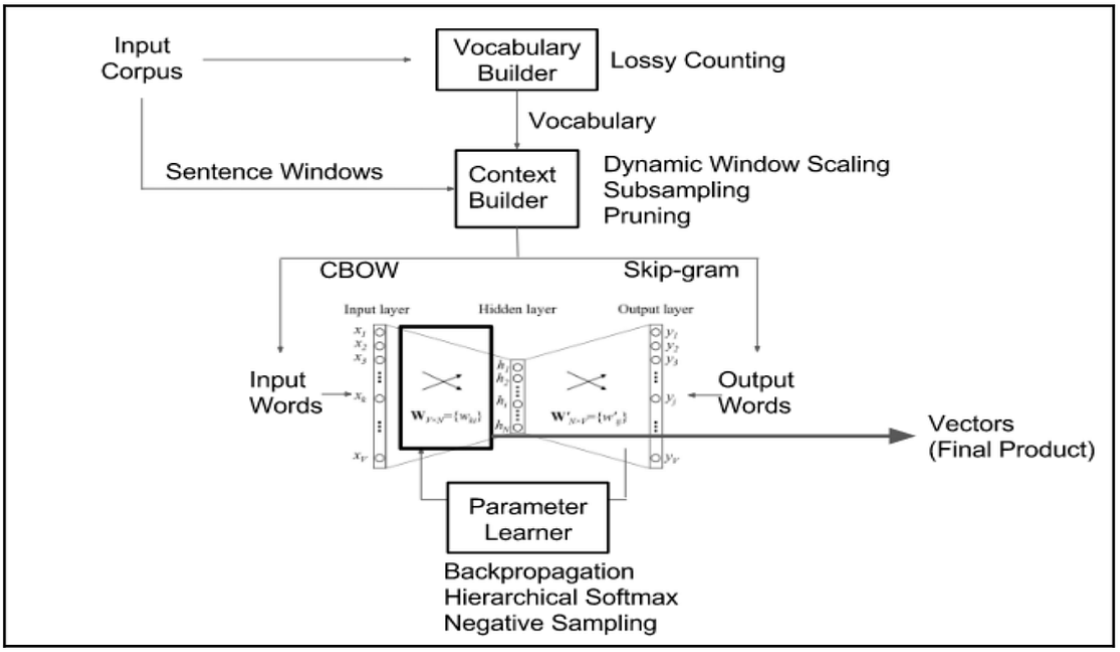
```
# Load TinyBERT model and tokenizer
model_name = "prajjwal1/bert-tiny"
tokenizer = BertTokenizer.from_pretrained(model_name)
model = BertForSequenceClassification.from_pretrained(model_name, num_labels=8)

# Tokenize and encode text data
train_encodings = tokenizer(X_train, truncation=True, padding=True, return_tensors='pt')
test_encodings = tokenizer(X_test, truncation=True, padding=True, return_tensors='pt')
```

   b. Model Selection: TinyBERT:



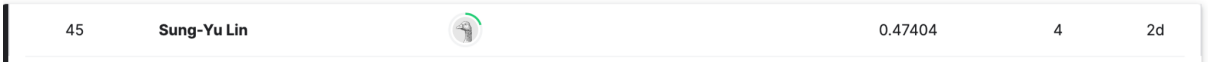
4. Experiments and Result
   a. use word2Vec and NN layer model to predict:

      I employed Word2Vec, a popular word embedding technique, to convert the preprocessed text into continuous vector representations. Both the Skip-Gram and Continuous Bag of Words (CBOW) architectures were explored, and the resulting word embeddings captured semantic relationships between words.
      The neural network model consisted of an embedding layer initialized with Word2Vec embeddings, followed by one or more fully connected (dense) layers. The model architecture was designed to capture complex patterns in the embedded text data, enhancing its ability to discern emotions.



      (result score: 0.1792)

   b. use tiny-bert:
      To assess the effectiveness of our emotion recognition model, we employed standard evaluation metrics such as accuracy, precision, recall, and F1 score. Additionally, we conducted a detailed analysis of the model's performance across different emotion categories.
      The obtained results were thoroughly analyzed to gain insights into the model's strengths and areas for improvement. We explored instances where the model excelled and identified challenges it faced in recognizing specific emotions.



      (final result score: 0.47404)

5. Discussion and future work
   a. Challenges and Limitations:
      Despite the positive outcomes, our study encountered challenges inherent to Twitter data, such as the use of informal language, slang, and the brevity of tweets. Additionally, the

dynamic nature of language evolution on social media poses ongoing challenges for emotion recognition models.

b. Ethical Considerations:
   The ethical considerations addressed in our study include potential biases in the training data and the responsible deployment of emotion recognition technology. We acknowledge the importance of ongoing scrutiny in mitigating biases and ensuring fair and unbiased predictions.
c. Interpretability:
   Interpreting the decisions of emotion recognition models remains a crucial area of exploration. While we implemented visualization techniques for the Word2Vec-NN model, further research is needed to enhance interpretability and transparency, facilitating user trust and understanding.
d. Future Work:
   i. Multimodal Emotion Recognition:
      xpanding our model to incorporate multiple modalities, such as images and emojis in addition to text, could further improve emotion recognition accuracy. Integrating these diverse data sources may provide a more comprehensive understanding of user emotions.
   ii. Fine-Tuning for Domain-Specific Emotions:
      Customizing the model for specific domains, such as mental health discussions or product reviews, represents a valuable avenue for future work. Fine-tuning the model on domain-specific datasets can enhance its sensitivity to context-specific emotional expressions.
   iii. Robustness to Evolving Language:
      Adapting the model to handle the ever-evolving nature of language on Twitter is crucial. Continuous monitoring and updates to the model's vocabulary and semantics will ensure its relevance and effectiveness in capturing emerging linguistic expressions.
   iv. Cross-Cultural Considerations:
      Extending the model's generalizability to diverse cultural contexts is an important direction for future research. Exploring cross-cultural nuances in emotion expression on Twitter will contribute to a more inclusive and globally applicable emotion recognition model.