[redacted]

Emotion-Focused Analysis of Stock Tweets: Challenges and Insights

Text mining project – Master’s degree in Data Science (AY: 2024/2025)

Outline

- Introduction
- EDA
- Dataset cleansing
- Text preprocessing
- Feature extraction
- Text classification task
- Topic modeling task
- Conclusions and future perspectives

Outline

- Introduction
- EDA
- Dataset cleansing
- Text preprocessing
- Feature extraction
- Text classification task
- Topic modeling task
- Conclusions and future perspectives

Introduction

Key Tasks

Emotion Classification: Multi-class classification of stock-related tweet emotions.

Topic Modeling: Discover underlying topics linked to emotional expressions.

Relevance

Correlation between public sentiment and stock market trends.

Practical implications for **financial decision-making**.

Challenges

Reduced **dataset size.**

Brevity and linguistic **variability** of tweets.

Fine-grained emotion categorization

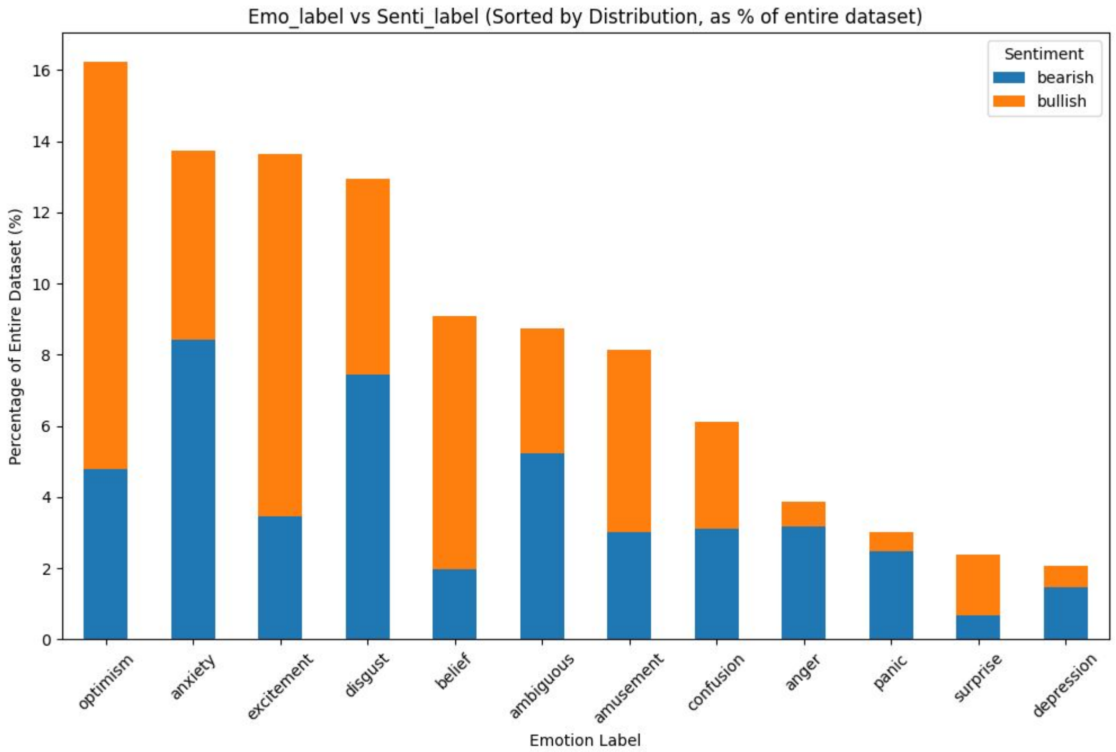
Outline

- Introduction
- EDA
- Dataset cleansing
- Text peprocessing
- Feature extraction
- Text classification task
- Topic modeling task
- Conclusions and future perspectives

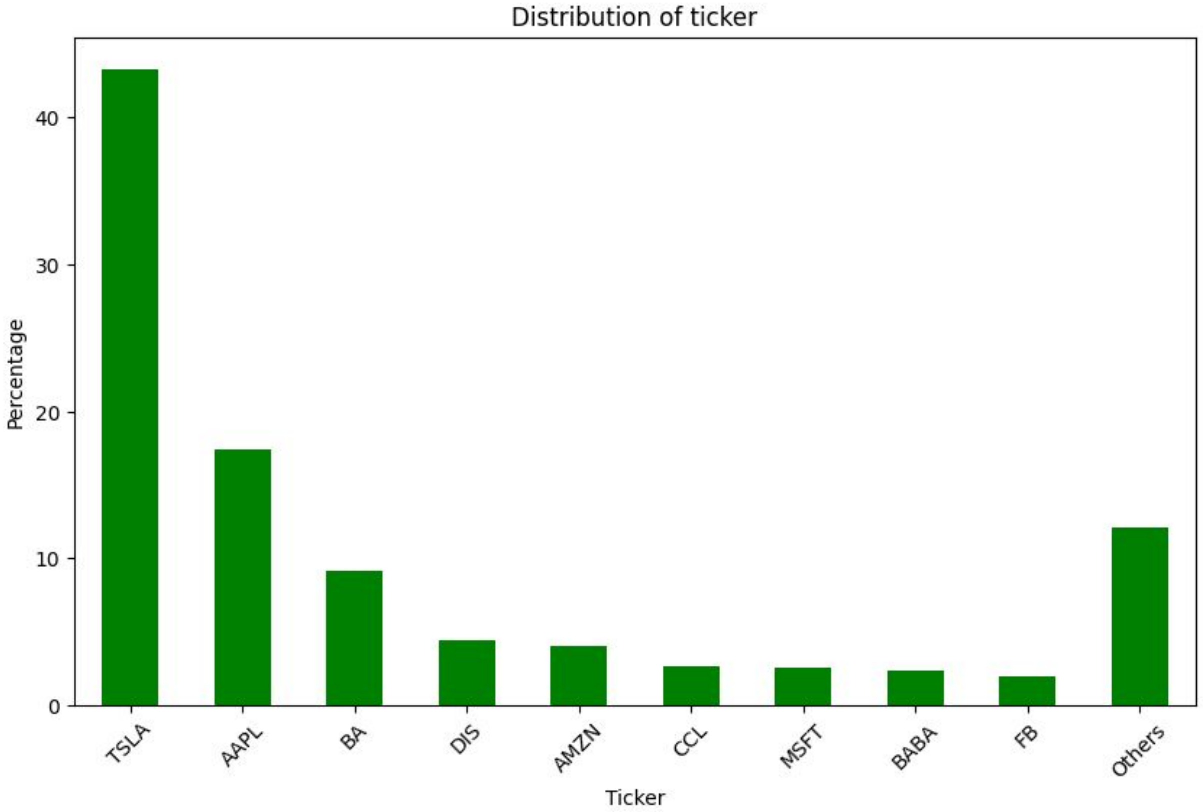
Dataset exploration
[Overview]

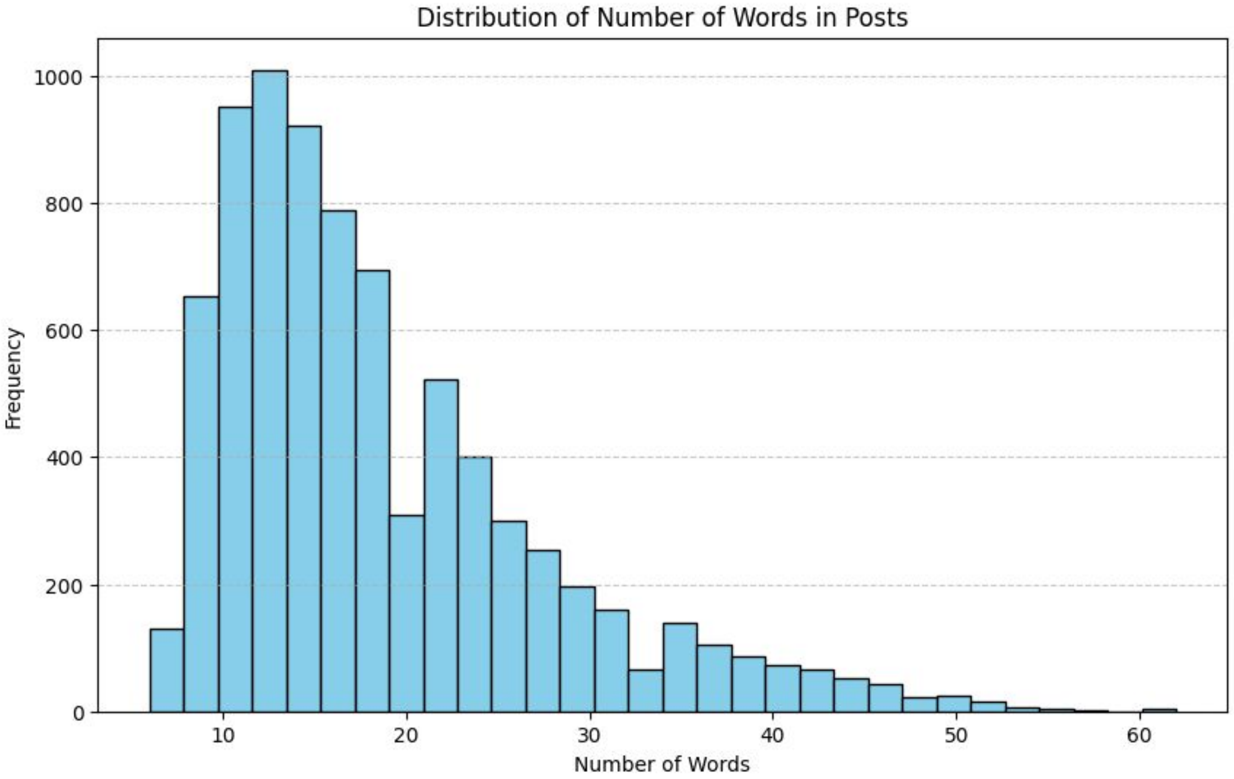
Columns breakdown:

- id [Int]
- date [yyyy-mm-dd]
- ticker [String]
- emo_label [String]
- senti_label [String]
- original [String]
- Processed [String]



Dataset exploration
[Distributions]





Outline

- Introduction
- EDA
- Dataset cleansing
- Text preprocessing
- Feature extraction
- Text classification task
- Topic modeling task
- Conclusions and future perspectives

Dataset cleansing

1

Unique Ticker Extraction

From training, validation, and test sets.

2

Complete Sorted Ticker List

Alphabetically sorted list from the training set.

3

Ticker-to-Company Mapping

Dictionary associating tickers with company names.

4

Redundant Column Removal

Dropping ‘id’, ‘date’, and ‘ticker’, retaining ‘processed’ for comparison.

Outline

- Introduction
- EDA
- Dataset cleansing
- Text preprocessing
- Feature extraction
- Text classification task
- Topic modeling task
- Conclusions and future perspectives

Text pre-processing [Overview]

Corpus Definition

'original' column = document collection

Modular Pipeline

3 additive modules based on text representation

Implementation

Python's *re* library, custom functions

Tokenization

Methods vary by text representation

Text pre-processing [Module 1]

Ticker Replacement

Map tickers to company names using predefined dictionary.

Placeholder Mapping

Replace company names with company_name to mitigate bias.

Newline Removal

Eliminate newline expressions (\n).

Text pre-processing [Module 1]

Quote standardization

- Convert diverse quote symbol to standard " format

Whitespace normalization

- Remove multiple whitespaces

Text pre-processing [Module 2]

Repeated Punctuation Tokenization	Replace multiple punctuation with tokens (multiple_exclamation, multiple_question, multiple_ellipsis).
Neutral Punctuation Removal	Remove [	,"], conditionally handle [.;:] based on context (e.g., decimals, emoticons).
Emoji Encoding	Translate emojis into single-term tokens using underscores (e.g., smiling_face).

Text pre-processing [Module 3]

Lemmatization

- Correctness + Completeness
- Small document collection
- *Spacy* library for integrated POS tagging.

Text pre-processing [Modules]

Module 1

Contexualized embeddings

Module 2

Word2Vec embeddings

Module 3

BoW and TF-IDF

Text pre-processing
[Lowercasing and tokenization]

Contextualized Embeddings

- Case information preserved
- Tokenization handled by the model.

Word2Vec

- *Nltk* library used
- Lowercasing and tokenization performed sequentially

BoW and TF-IDF

- *Spacy* library used
- Automatic ower casing and prior to lemmatization

Text pre-processing [Stopwords removal]

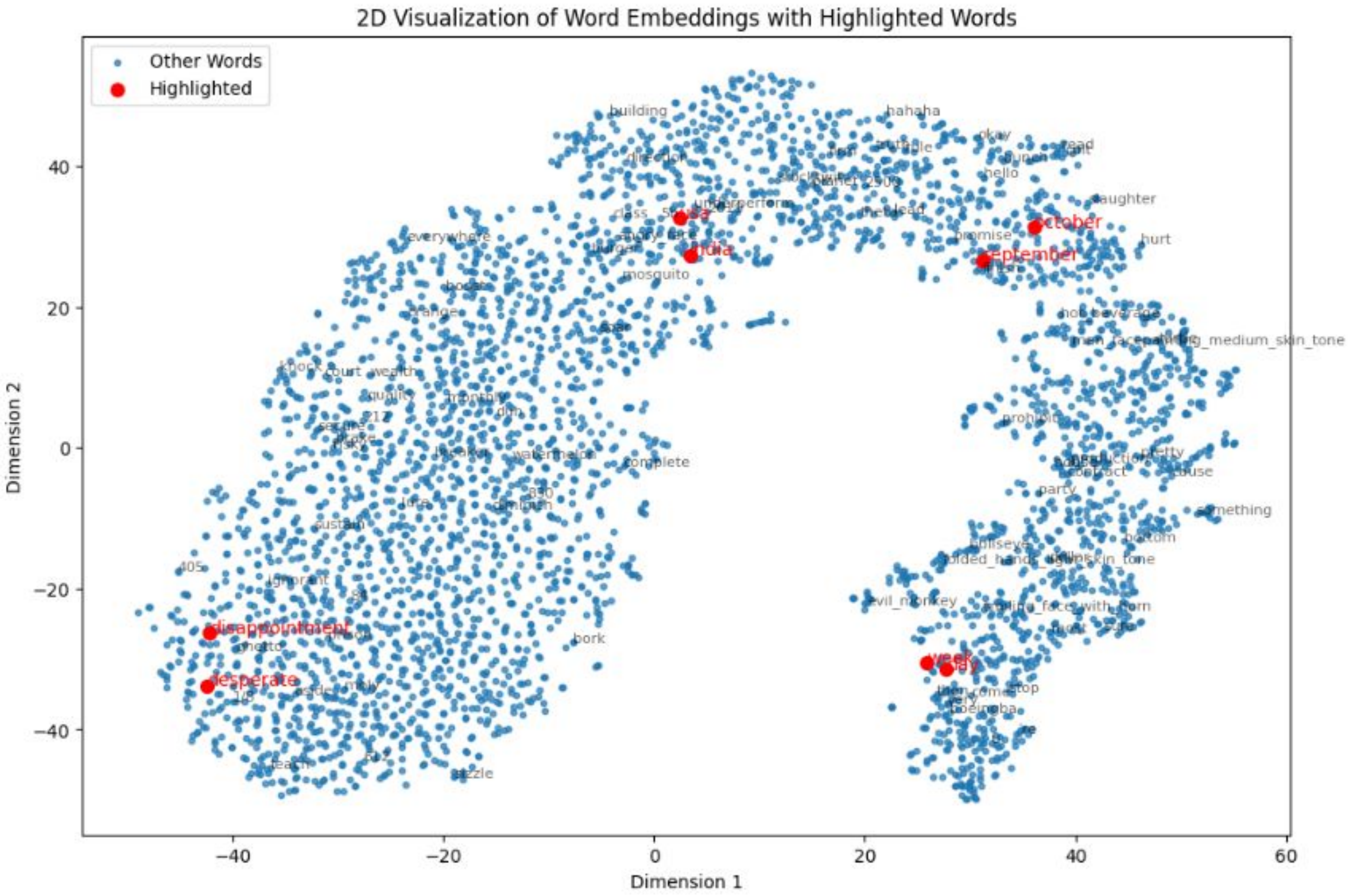
BoW

- NLTK *stopwords* method
- Beneficial due to sensitivity to high-frequency words

TF-IDF

- *stop_words* argument within the vectorizer
- Robust to stopwords, but still preferable to reduce dimensionality

Text pre-processing
[Embedding exploration: word2vec]



Outline

- Introduction
- EDA
- Dataset cleansing
- Preprocessing
- Feature extraction
- Text classification task
- Topic modeling task
- Conclusions and future perspectives

Feature extraction [Textual features]

BoW	TF-IDF	TF-IDF	Embeddings	Contextualized embeddings	
	Unigram based	Bigram based	Word2Vec	BERTweet	Distil-RoBERTa fine-tuned

Feature extraction [Hand-crafted features]

Text Length

Total number of tokens in a document

Uppercase Ratio

Ratio of uppercase terms to text length

Processing

Tokenization with NLTK

Emojis excluded in computation

Feature extraction [Bing Liu's Lexicon]

Agreement Score (AS):
Measures the balance of positive and negative terms.

$$\mathrm{AS} = \begin{cases} 0, & \text{if } T_{\mathrm{p}} + T_{\mathrm{n}} = 0 \\ 1 - \sqrt{1 - \left| \frac{T_{\mathrm{p}} - T_{\mathrm{n}}}{T_{\mathrm{p}} + T_{\mathrm{n}}} \right|}, & \text{otherwise} \end{cases}$$

Polar Word Occurrence (PWO):
Indicates sentiment prevalence.

$$\mathrm{PWO} = \begin{cases} 1, & \text{if } T_{\mathrm{p}} > T_{\mathrm{n}} \\ -1, & \text{if } T_{\mathrm{p}} < T_{\mathrm{n}} \\ 0, & \text{if } T_{\mathrm{p}} = T_{\mathrm{n}} \end{cases}$$

Feature extraction [NRC Lexicon]

Emotions: anger, anticipation, disgust, fear, joy, negative, positive, sadness, surprise, trust.

Normalized Emotion Count:

$$\text{Normalized Emotion Count } (e) = \frac{N(e)}{n}$$

Feature extraction [VADER]

$$\texttt{vader\_pos} = \frac{\sum_{i \in \mathcal{P}} \mathbf{I}_i}{\sum_{i \in \mathcal{P}, \mathcal{N}, \mathbf{N}} \mathbf{I}_i}$$

$$\texttt{vader\_neu} = \frac{\sum_{i \in \mathbf{N}} \mathbf{I}_i}{\sum_{i \in \mathcal{P}, \mathcal{N}, \mathbf{N}} \mathbf{I}_i}$$

$$\texttt{vader\_neg} = \frac{\sum_{i \in \mathcal{N}} \mathbf{I}_i}{\sum_{i \in \mathcal{P}, \mathcal{N}, \mathbf{N}} \mathbf{I}_i}$$

$$\texttt{vader\_compound} = \frac{\sum_{i \in \mathcal{P}} \mathbf{I}_i - \sum_{i \in \mathcal{N}} \mathbf{I}_i}{\sum_{i \in \mathcal{P}, \mathcal{N}, \mathbf{N}} (\mathbf{I}_i)^2}$$

Feature extraction [Resulting dataset]

ALL FEATURES AGGREGATED INTO A SINGLE DATASET

NRC LEXICON EMOTIONS HAVE DEDICATED COLUMN.

SUITABLE INPUT BOTH FOR ML MODELS OR NN

Outline

- Introduction
- EDA
- Dataset cleansing
- Preprocessing
- Feature extraction
- Text classification task
- Topic modeling task
- Conclusions and future perspectives

Text classification

Parameter	Values
n_estimators	{100, 200, 300}
max_depth	{10, 20, None}
min_samples_split	{2, 5, 10}
min_samples_leaf	{1, 2, 4}

Parameter	Values
learning_rate	{0.01, 0.05, 0.1}
max_depth	{4, 6, 8}

Parameter	Values
C	{0.01, 0.1, 1, 10, 100}

Random forest, XGBoost and SVM are evaluated in terms of mF1-score

Brute force HP optimization

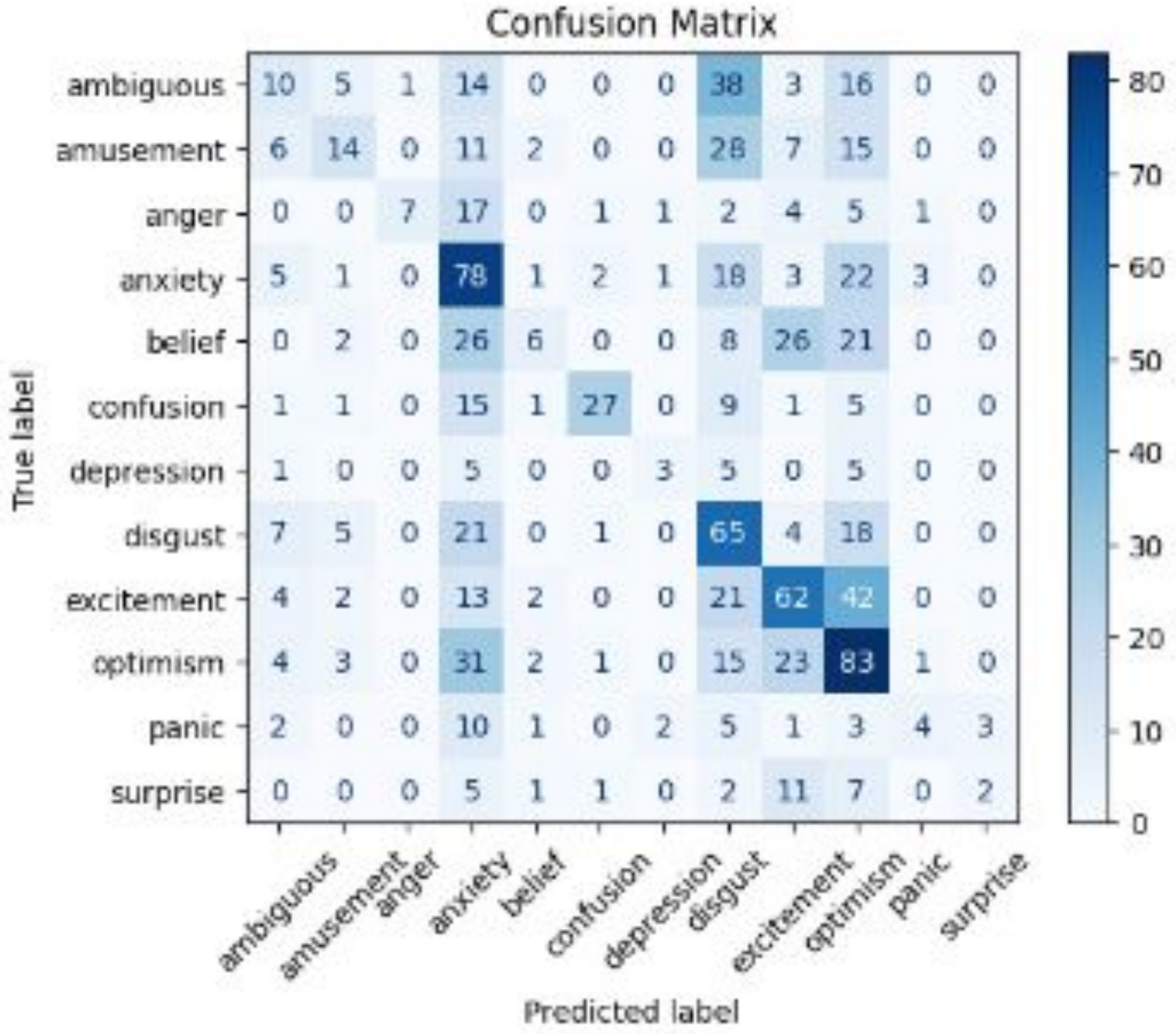
Different representations

For each classifier, two configurations:

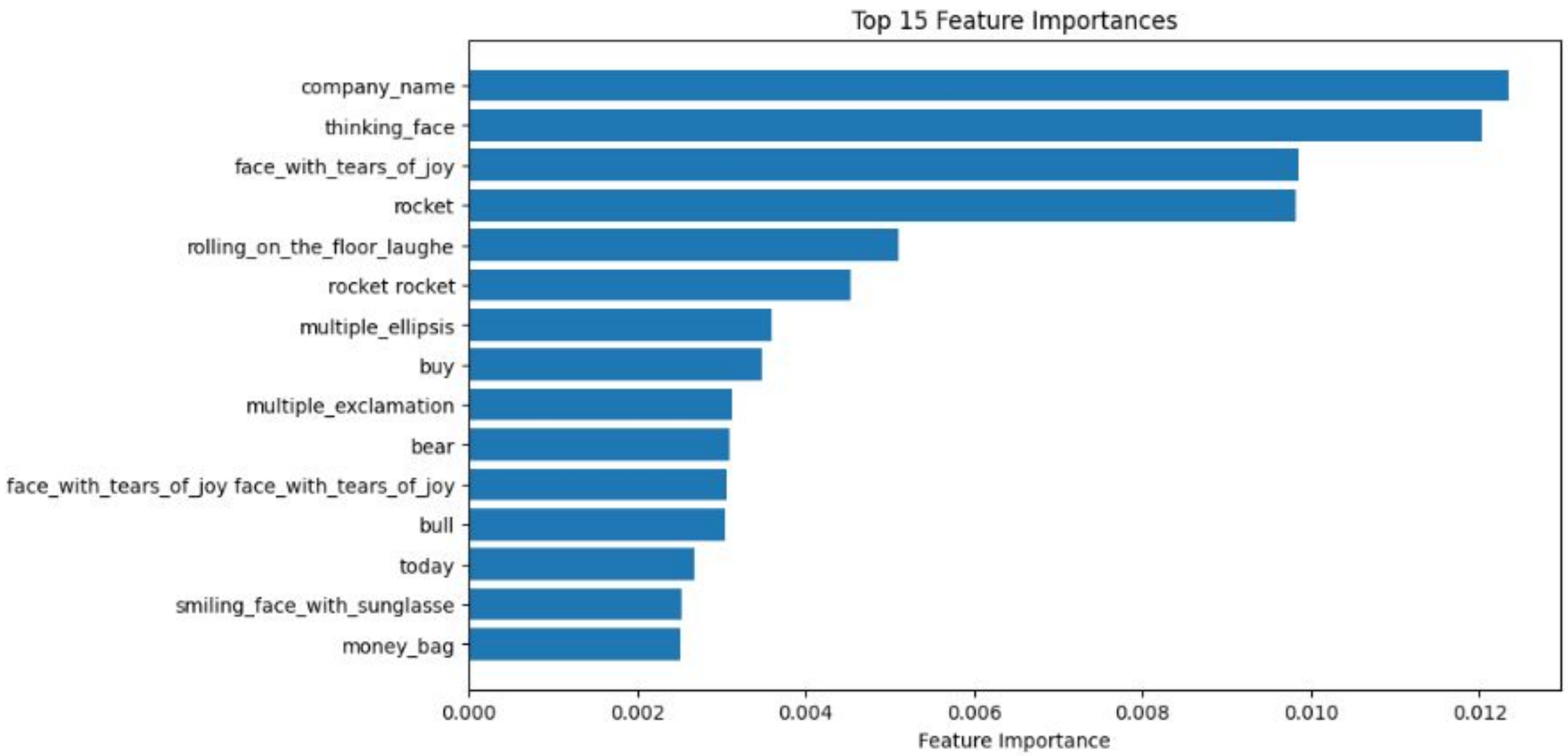
- With engineered features
- W/o engineered features

Text classification

Model	Acc.	m-F1
TF-IDF B	0.34	0.28
TF-IDF E	0.34	0.28
Bigram TF-IDF B	0.36	0.30
Bigram TF-IDF E	0.34	0.28
Word2Vec B	0.23	0.13
Word2Vec E	0.23	0.13
RoBERTa B	0.26	0.15
RoBERTa E	0.26	0.15
BERTweet B	0.26	0.11
BERTweet E	0.26	0.12



Text classification

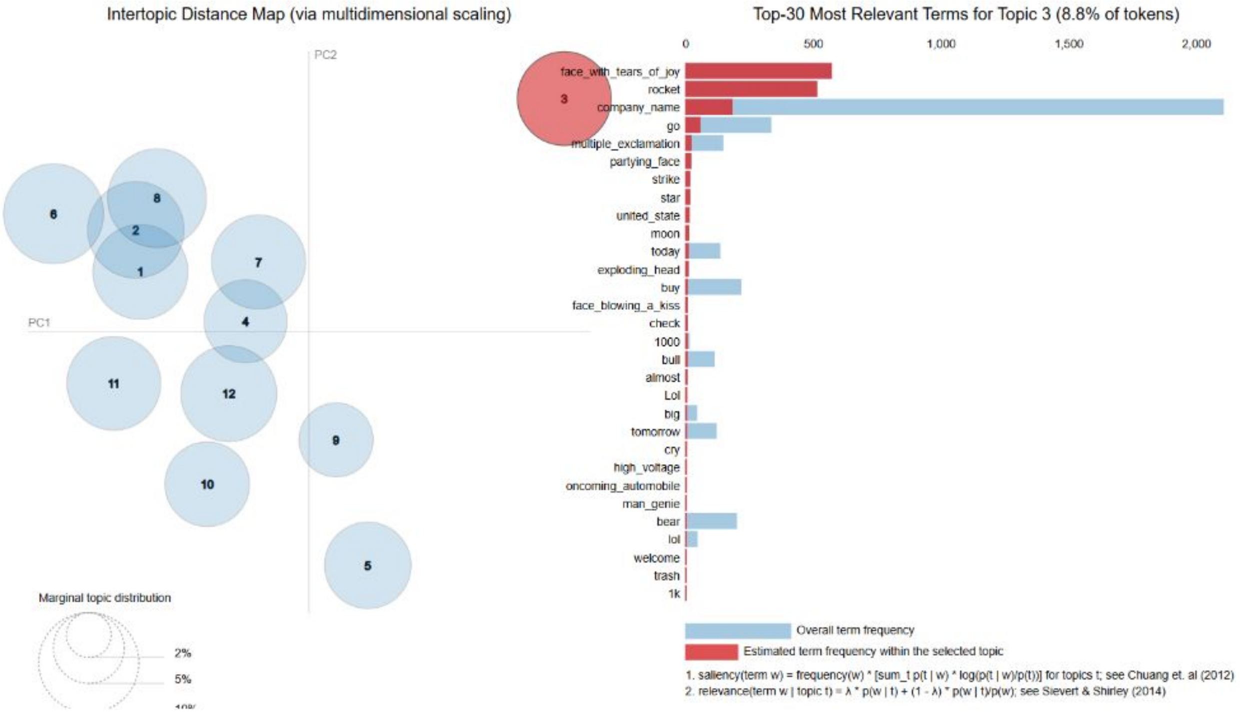


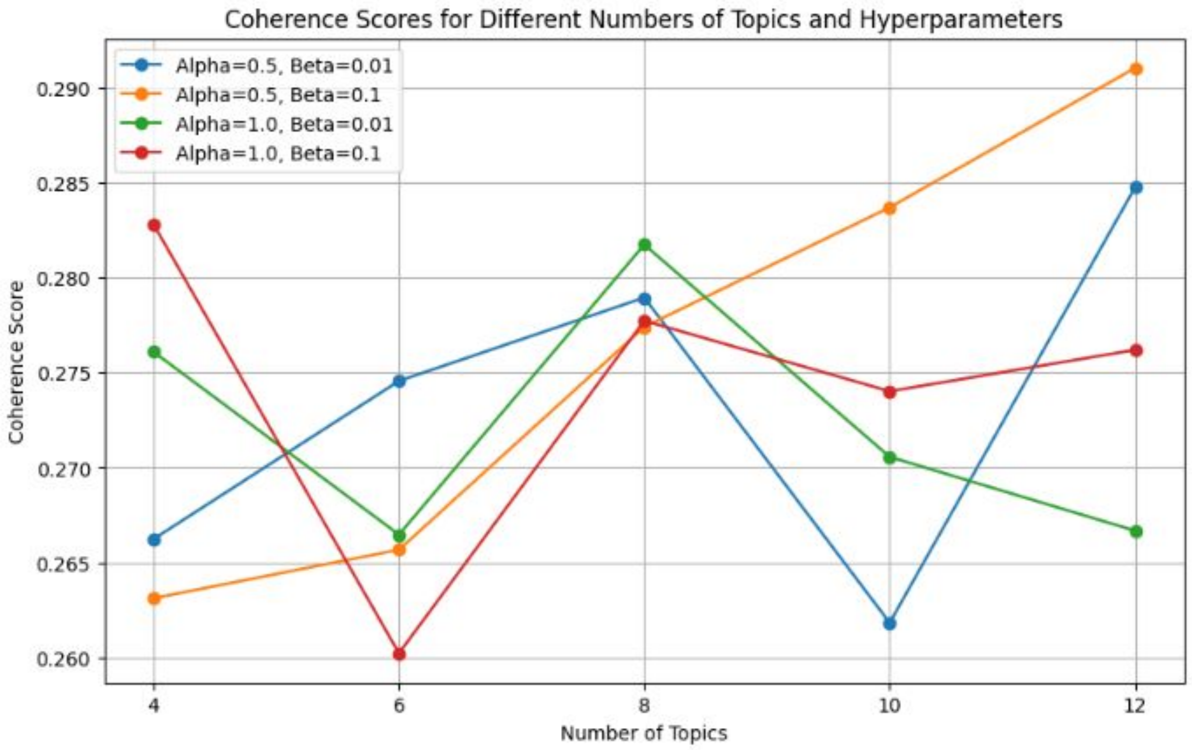
Outline

- Introduction
- EDA
- Dataset cleansing
- Preprocessing
- Feature extraction
- Text classification task
- Topic modeling task
- Conclusions and future perspectives

Topic modeling [LDA]

- Implemented on BOW representation
- Hyperparameters optimization [**k**, **α** and **β**] via brute force

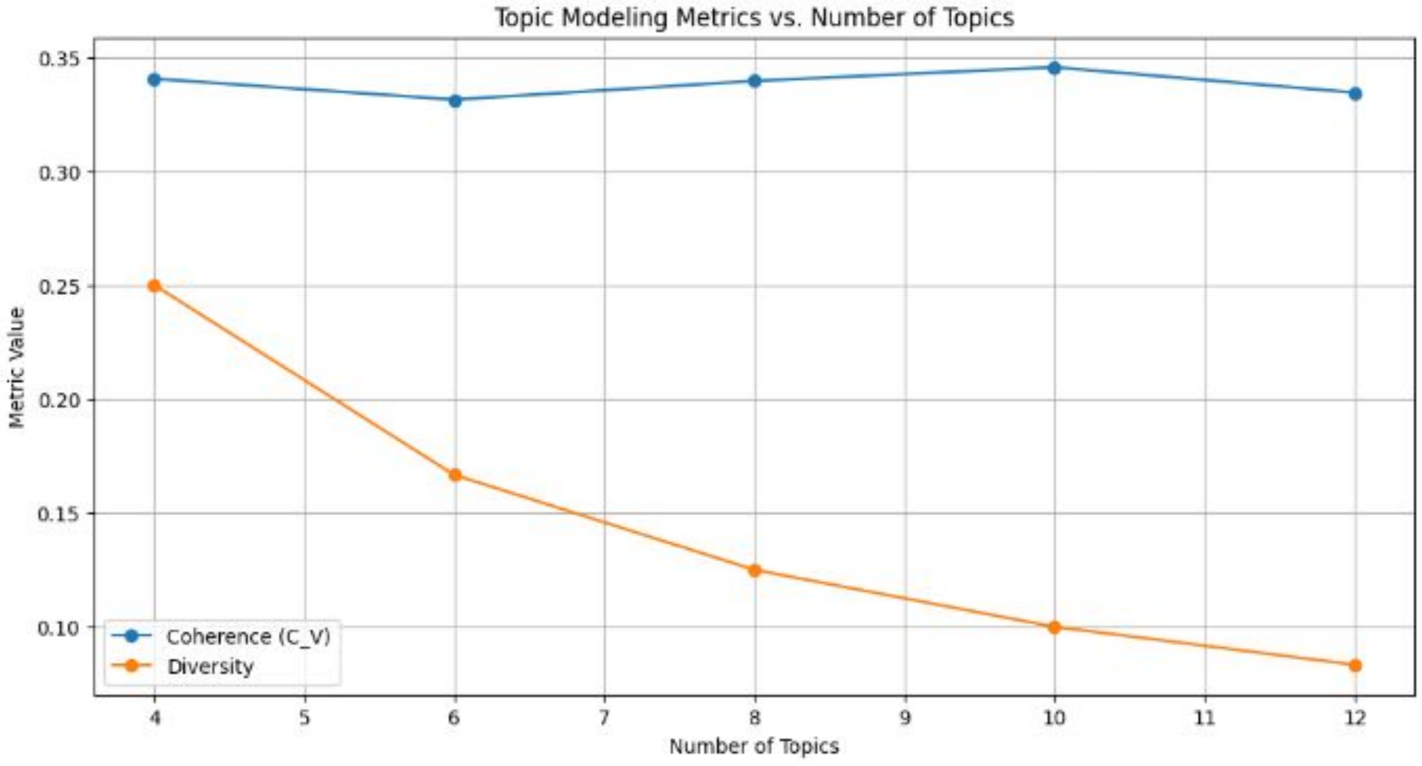




Topic modeling [BERTopic]

- Implemented with word2vec embedding
- Hyperparameters optimization [**k**] via brute force

Topic	Observed similarity
0	excitement, surprise
1	amusement
2	ambiguous
3	surprise, excitement, belief
4	amusement
5	ambiguous
6	unknown
7	belief
8	surprise



Outline

- Introduction
- EDA
- Dataset cleansing
- Preprocessing
- Feature extraction
- Text classification task
- Topic modeling task
- Conclusions and future perspectives

Conclusions

1. XGB and RF on bigram TF-IDF (0.32 mF1 and 0.31 mF1)
2. Engineered features don't show improvement
3. Overall poor performance confirmed by state-of-the-art models
4. Importance of emojis inclusion

Conclusions

Topic modeling task

- Overall poor performance, close to 0.3 of C_V
- Noise, temporal variability and shortness of documents
- Independent of hyperparameters values

Future perspectives

- Trying different models and evaluation metrics
- Enhance topic modeling with clustering results
- Automation of comparison between topics and labels
- Using topic modeling results for classification

Thanks for your attention.