UNVEILING ONLINE DECEPTION- A DEEP LEARNING APPROACH TO DETECT FAKE REVIEWS :THESIS DISSERTATION

By

Sridhar Pai Tonse

Student ID: **1096325**

Thesis Supervisor : Dr. Anil Vuppala

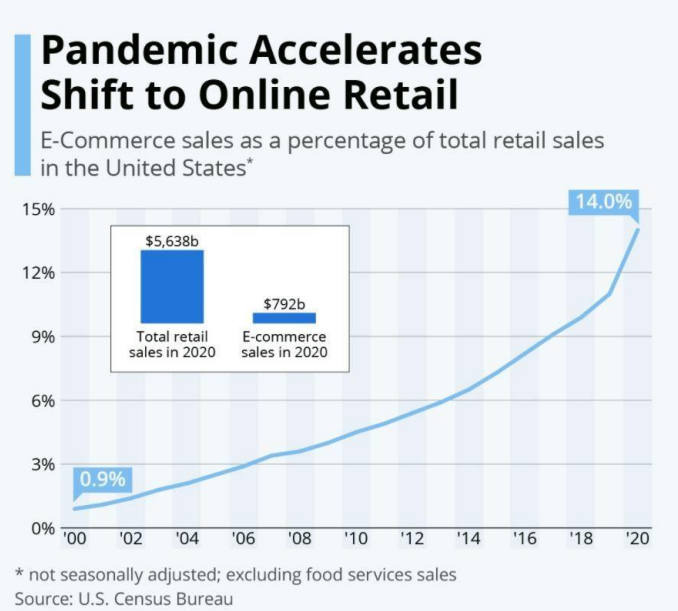
Masters in Data Science

Liverpool John Moore University, UK

Agenda

- What is Fake Reviews and how big is it ?
- How do we do Fake Reviews Detection ?
- What has been done so far ?
- Aims and Objectives of Thesis
- Problem Statement
- Research Methodology
- Implementation
- Results and Analysis
- Conclusion and Recommendations
- Q&A

Online Fake Reviews



A fake review is a fraudulent or deceptive review that is posted online about a product / service.

Fake online reviews cost $152 billion a year – World Economic Forum, 2021

Estimates suggest up to 30% of online reviews might be fake [survey by University of Illinois]. This means a significant portion of information influencing your purchases could be misleading.

Most e-shoppers don't trust online reviews

What is the impact of Fake Reviews ?

Consumer Trust : Fake reviews can erode consumer trust in online platforms and reviews.

Misleading Information: Fake reviews can mislead consumers by artificially inflating or deflating the perceived quality of a product or service.

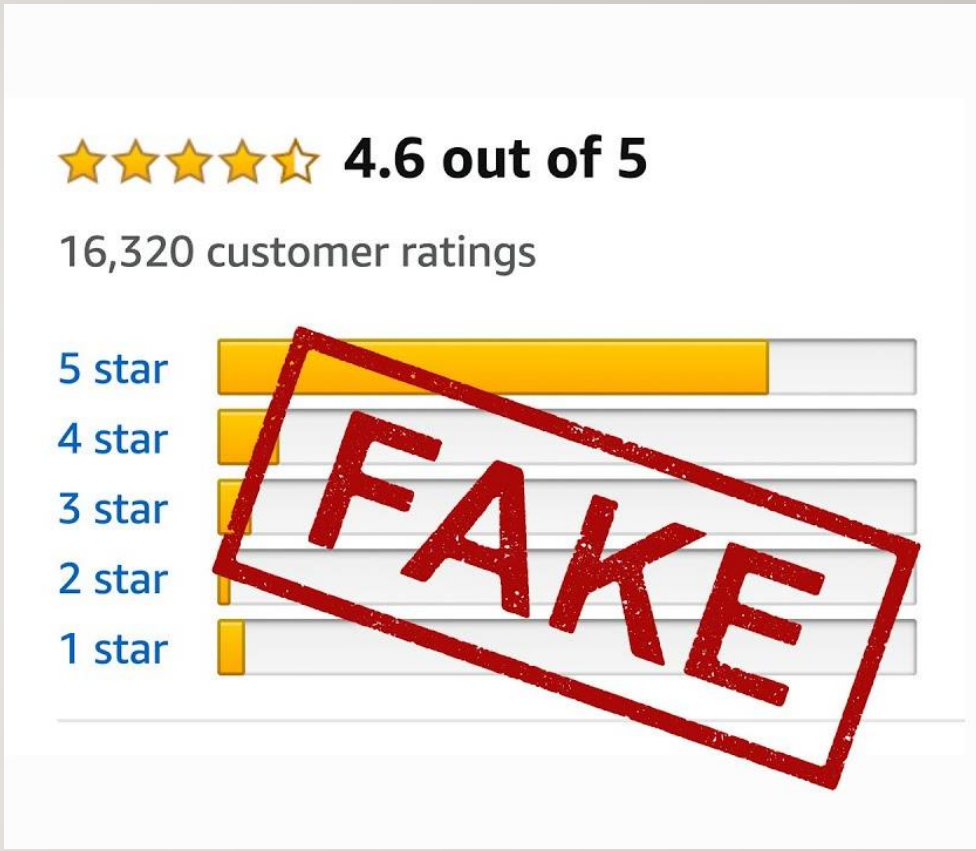
Unfair Competition: Businesses that engage in posting fake reviews may gain an unfair advantage over competitors who rely on genuine reviews.

Reputation Damage: Businesses caught using fake reviews risk damaging their reputation

Review platforms like TripAdvisor employ automated systems and human moderators to detect and remove suspicious activity. In 2020, TripAdvisor prevented over 67% of fake reviews from being posted.

Introduction on Fake Reviews Detection

- Fake Reviews is a major challenge in the era of digital commerce.
- We propose a novel framework that harnesses the power of Natural Language Processing (NLP) and Deep Learning Networks to identify fake reviews.
- We study and evaluate traditional and deep learning embedding techniques such as **TFIDF, Glove, BERT, RoBERTa, DistilBERT, XLNet** and the results demonstrate the effectiveness of our proposed model in accurately identifying fake reviews.
- This thesis research will contribute to the academic community by advancing the understanding of fake review detection and providing valuable resources (e.g., algorithms, datasets, insights) for future research.



Research Gap:

I. Existing studies in fake review detection predominantly center around supervised learning methods, which heavily rely on **Labeled data** for training and testing.

II. There remains a significant gap in the exploration of **Embedding techniques** coupled with machine learning (ML) and deep learning (DL) approaches for fake review detection.

III. Despite the growing interest in advanced Embedding Techniques such as **TF-IDF, Glove, BERT, ALBERTA, DISTILBERT, and XLNET**, their application in this domain remains relatively unexplored.

IV. This research aims to address this gap by investigating the effectiveness of these **Embedding techniques** in conjunction with ML and DL methods for fake review detection

Problem Statement

1. The proliferation of fake reviews online presents a complex challenge for consumers, businesses, and digital platforms. Identifying of fake reviews is complex challenge.

2. This research aims to come up with novel approach to identify fake reviews. As part of this we will study and compare various Embedding Techniques for fake review detection such as **TF-IDF, Glove, BERT, ALBERTA, DISTILBERT, and XLNET** across Machine Learning and Deep Neural Networks.

Aims and Objectives of Thesis

Aim:

- Aim of this Thesis is to come up with A Novel approach using Advanced ML / DL Techniques that will clearly distinguish the fake reviews and genuine reviews resulting effective Fake Review Detection.

Objectives:

- To analyse various embedding techniques along with Advanced ML and DL models for effective fake review detection.
- To propose a suitable model architecture for the fake review detection.
- To assess the effectiveness of different embedding techniques such as TFIDF, Glove, BERT, RoBERTa, DistilBERT, XLNET along with ML and DL Models.
- To evaluate the overall performance of the proposed framework using appropriate metrics and validation techniques to assess its efficiency and accuracy in fake review detection.

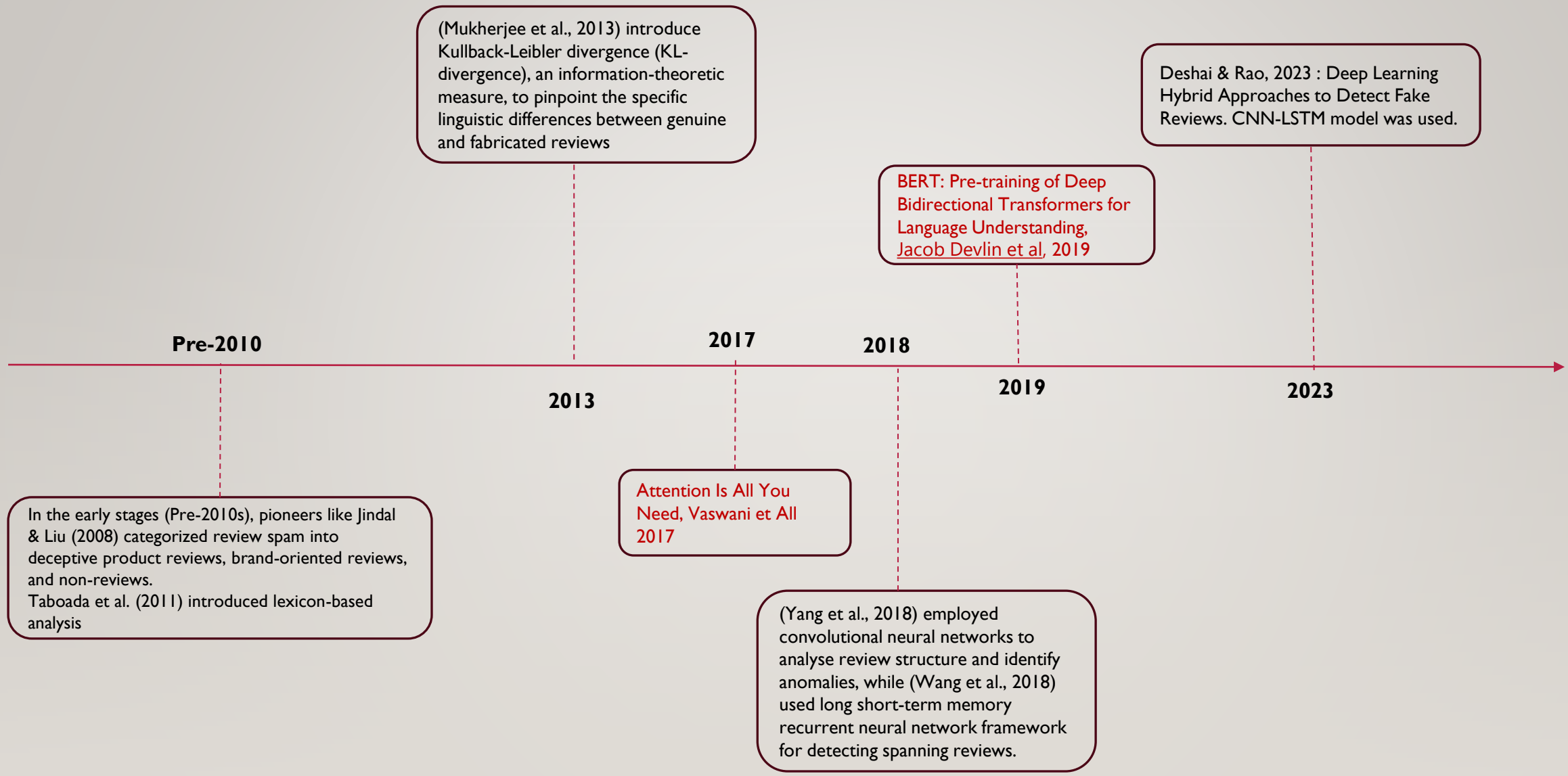
Literature Review

Explore the history and progress of detecting fake reviews from 2005 to 2023, studying over 40+ research papers.

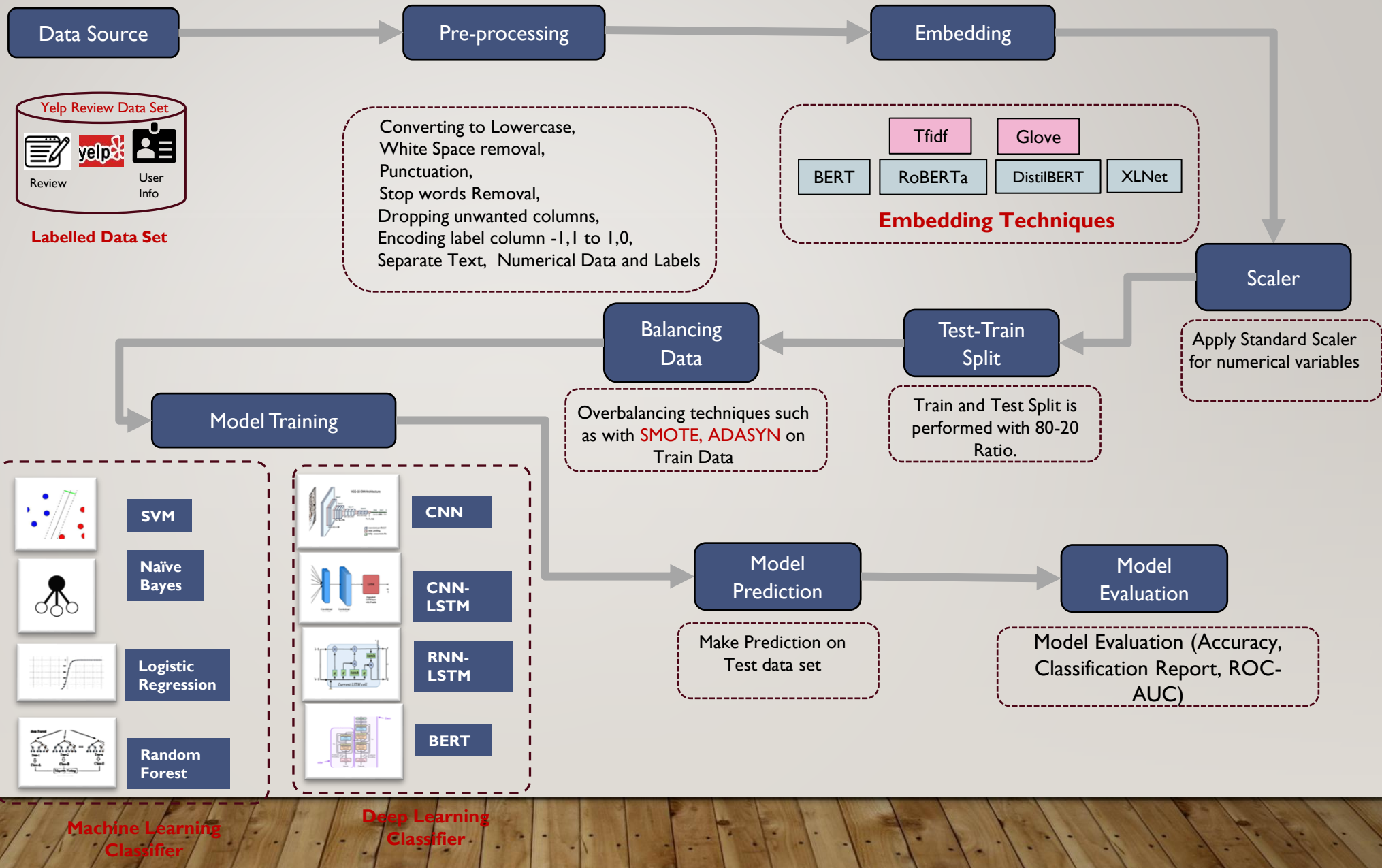
I. Early approaches (pre-2010) :	• Linguistic and behavioral differences between genuine and fake reviews. • Exploration of lexical diversity and writing style as foundational elements for future research in fake review detection.
II. Machine Learning approaches	• Utilization of labeled datasets and classification algorithms to train models. • Focus on feature engineering and model optimization techniques for improved performance.
III. Deep Learning approaches	• Incorporation of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks for enhanced performance. • Semi supervised methods ie combination of labeled and unlabeled data. Unsupervised methods ie no labelled data. • Exploration of the vulnerability of deep learning models to adversarial attacks.

Overall, the papers provide a comprehensive journey through various detection methodologies, highlighting advancements and challenges in the evolving landscape of fake review detection

Literature Review



Research Methodology Framework



Embedding techniques

What is Embedding Techniques ?

Traditional feature engineering may involve creating numerical features based on domain knowledge or extracting statistical properties from raw data. However, in NLP tasks, raw text data cannot be directly fed into machine learning models. Embedding techniques like Word2Vec, GloVe, BERT, and others transform words or text sequences into dense numerical vectors, capturing semantic and contextual information.

What are different Embedding Techniques ?

1. TF-IDF: (Term Frequency Inverse document frequency)

Focuses on identifying important words in a document. It considers how often a word appears in the document (frequency) and how rare it is overall (inverse document frequency). This helps prioritize keywords that are specific and essential to the document's content.

2. GloVe: (Global Vectors for Word Representation)

Captures semantic relationships between words. It analyzes a massive amount of text data to see how often words co-occur, essentially learning that words appearing together frequently share similar meanings. This allows tasks like finding synonyms or analogies based on word vector similarities.

Glove global corpus - glove.6B.200d.txt

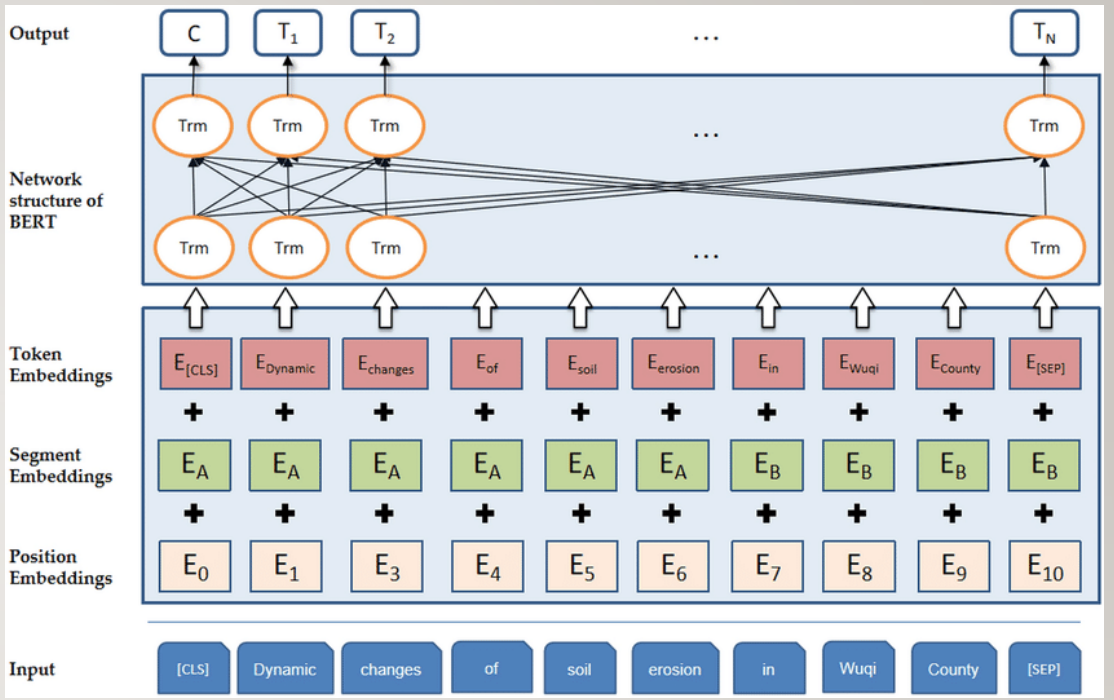
Embedding techniques

3. BERT (Bidirectional Encoder Representations from Transformers) **:**

- A powerful technique for understanding words based on their context in a sentence. It's pre-trained on a huge dataset of text and code, allowing it to learn contextual word representations.
- BERT can be used as an embedding technique to generate contextualized word embeddings for text data. Unlike traditional word embeddings like Word2Vec or GloVe, which produce fixed representations for each word regardless of context, BERT captures contextual information by considering the entire sentence bidirectionally..

4. RoBERTa: Builds on BERT's success, aiming for better efficiency and performance. It utilizes a more sophisticated masking strategy during training and removes unnecessary steps, making the training process faster and potentially improving performance on certain NLP tasks while retaining BERT's core strengths.

5. DistilBERT: Creates a compact and efficient version of BERT through a technique called knowledge distillation. It learns from a larger pre-trained model (like BERT) but with a significantly smaller size, allowing for faster processing and lower resource requirements, while maintaining good performance on NLP tasks



Source: https://www.researchgate.net/publication/359301499_Deep_learning-based_methods_for_natural_hazard_named_entity_recognition/figures?lo=1

6. XLNet: Addresses limitations in BERT's pre-training process by considering all possible permutations of ordering the input words. This can potentially capture more nuanced relationships between words compared to BERT, but it comes with a more complex architecture and even higher computational demands.

Comparison of different Embedding techniques

Feature	TF-IDF	GloVe	BERT	Roberta	DistilBERT	XLNet
Technique	Statistical	Neural Network	Neural Network	Neural Network	Neural Network (compressed)	Neural Network
Focus	Word importance in document	Word co-occurrence	Contextual word representation	Contextual word representation (improved efficiency)	Efficient contextual representation	Advanced contextual representation
Strengths	Simple, interpretable, good for keywords	Captures semantics	State-of-the-art NLP performance	Efficient training over BERT	Faster inference than BERT	Potentially better relationships between words
Weaknesses	No word semantics	Less interpretable, computationally expensive	Complex, high resource requirements	Similar limitations as BERT	Lower accuracy than BERT	More complex architecture

Research Implementation

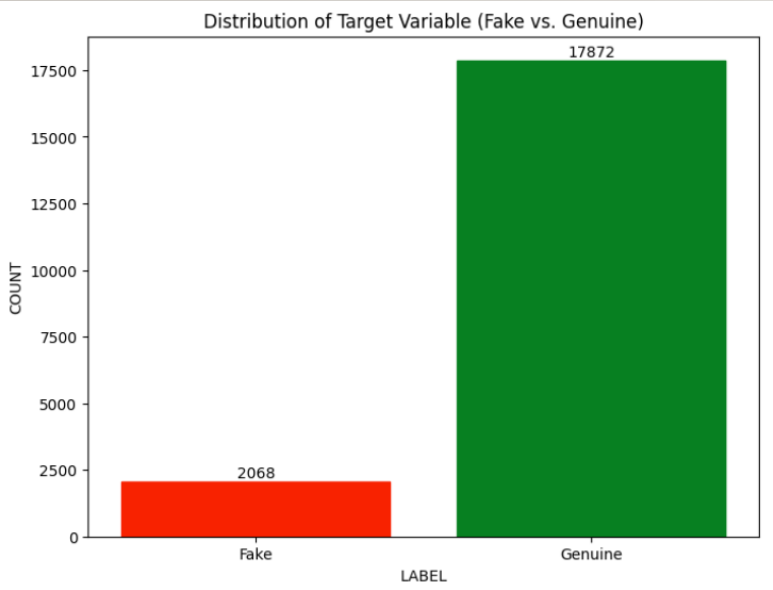
Dataset Description
EDA (Exploratory Data Analysis)
Balancing Techniques
Hardware and Software requirements

Data definition of Yelp Labeled Data Set

SL NO	COLUMN	NON-NULL COUNTS	DATA TYPE
0	ID	19940	INT64
1	USER_ID	19940	INT64
2	PRODUCT_ID	19940	INT64
3	RATING	19940	INT64
4	DATE	19940	OBJECT
5	LABEL	19940	INT64
6	REVIEW_TEXT	19940	OBJECT
7	AVERAGE_RATING	19940	FLOAT64
8	RATING_DEVIATION	19940	FLOAT64
9	TOTAL_PRODUCT_REVIEWS	19940	INT64
10	REVIEW_LENGTH	19940	INT64
11	RATING_CATEGORY	19940	INT64
12	SINGLE_RATING_CATEGORY	19940	INT64
13	REVIEW_COUNT_DATE	19940	INT64
14	SAME_DATE_MULTIPLE_REVIEWS	19940	INT64
15	MAX_USER_REVIEWS_DAY	19940	INT64
16	TIMESTAMP_DIFFERENCE	19940	OBJECT
17	AVERAGE_USER_REVIEW_LENGTH	19940	FLOAT64
18	TOTAL_USER_REVIEWS	19940	INT64
19	PERCENTAGE_POSITIVE_REVIEWS	19940	FLOAT64
20	RATIO_POSITIVE_NEGATIVE	19940	FLOAT64

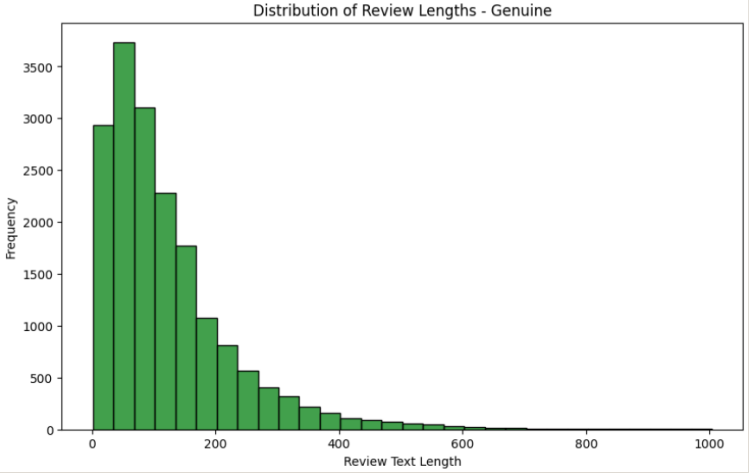
Data Set Used : Yelp Labeled data set available publicly is being used.
Dataset Dimensions:
Number of rows: 19940
Number of columns: 21

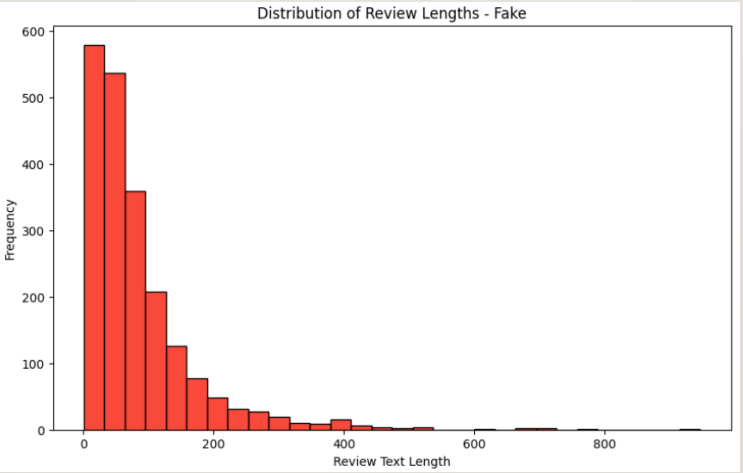
Distribution of target variable ie LABEL (Fake vs. Genuine)



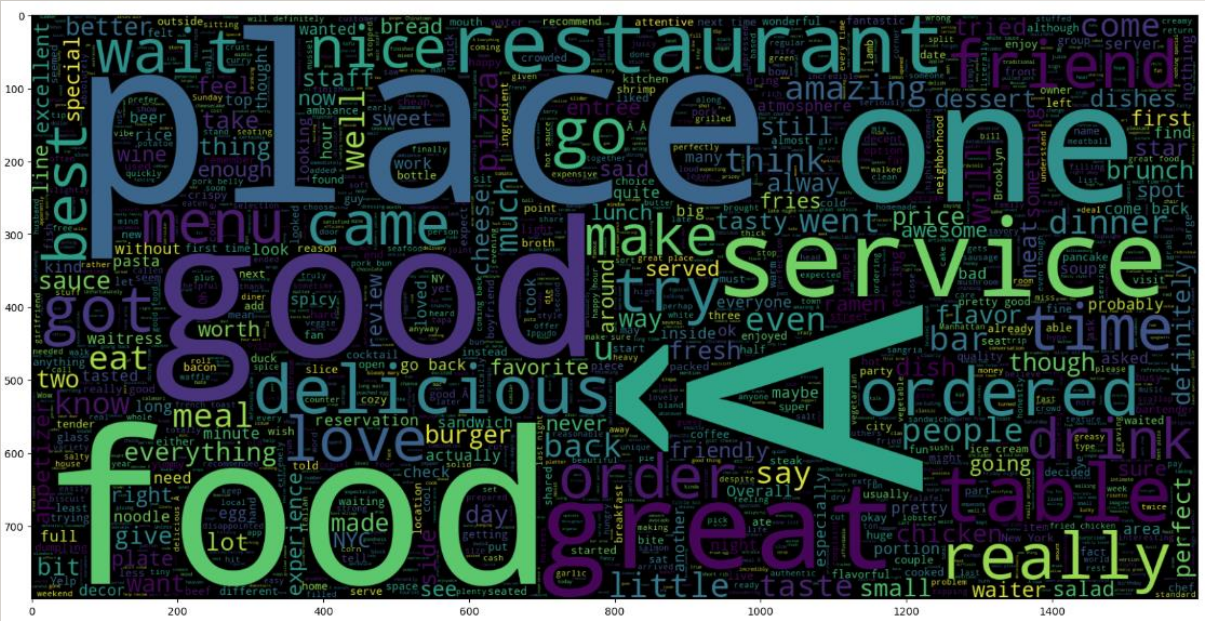
Label	Description	# of Entries	% of Entries
1	Genuine Review	17872	89.6%
-1	Fake Review	2068	10.4%

Distribution of Fake and Genuine Reviews

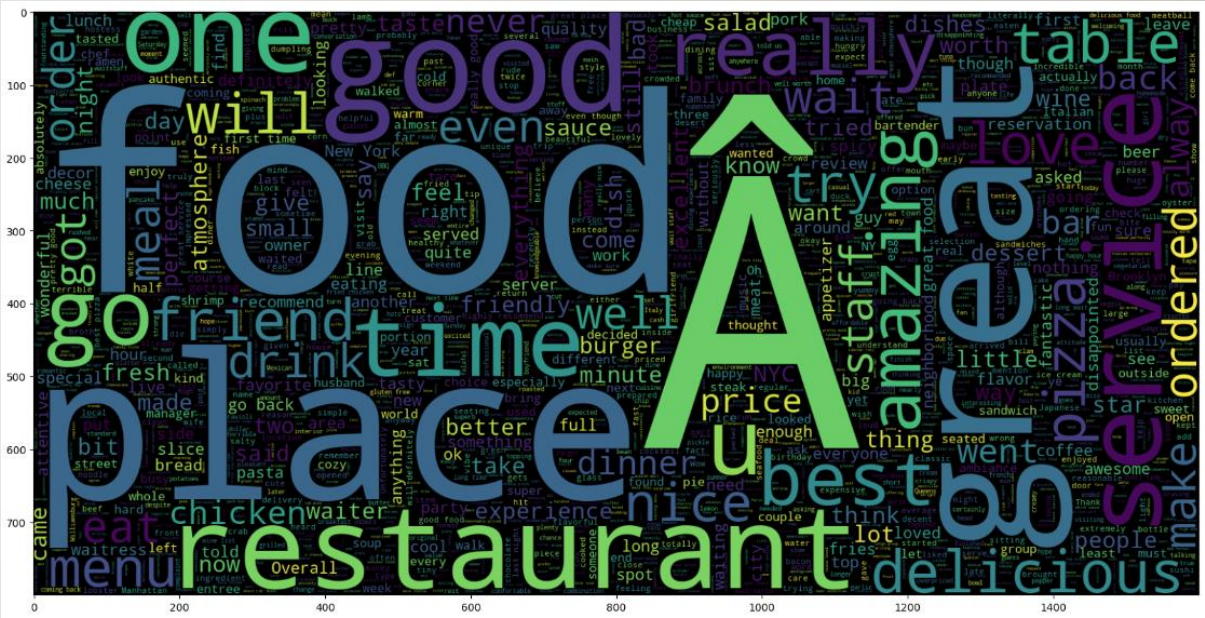




Word Cloud Fake Reviews



Word Cloud Genuine Reviws



Balancing Techniques

- To overcome imbalance we have two options

  1. Under sampling majority class
  2. Oversampling minority class

Label	Description	# of Entries	% of Entries
1	Genuine Review	17872	89.6%
-1	Fake Review	2068	10.4%

- SMOTE (Synthetic Minority Over-sampling Technique) is balancing technique that generates synthetic data. SMOTE generates synthetic samples for the minority class by interpolating between existing minority class samples. It works by identifying minority class instances that are close to one another and then creating synthetic instances along the line segments joining them.

- ADASYN is another balancing techniques which takes care of imbalance by generating synthetic data using K-nearest neighbour of each minority class. Advantage of ADASYN is that it will not just copy minority data multiple times, but generates more data for complex parts of data which is tricky for ML Algorithm to train on.

Implementation : Hardware and software requirements

Hardware Requirements

A computer with high processing power and a large storage capacity (at least 32GB of RAM, quad-core processor, and dedicated graphics card with at least 4GB of VRAM)500GB of storage space

Software Requirement

- Python programming language version 3.8 or higher for data analysis and machine learning tasks.
- TensorFlow version 2.4 or higher, an open-source library for machine learning and deep learning tasks.
- Scikit-learn version 1.0.3 or higher, a machine learning library for Python.
- Keras version 2.4 or higher, an open-source neural network library for Python.
- Matplotlib version 3.7.1 or higher, a plotting library for Python.
- Seaborn version 0.11.2 or higher, a data visualization library for Python.
- Pandas version 2.0.2 or higher, a data analysis library for Python.
- Numpy version 1.24.2 or higher, a numerical computing library for Python.
- Jupyter Notebook version 6.1 or higher, an interactive computing environment for Python.
- A version control system like Git version 2.29 or higher to track and manage changes to the code and data throughout the research process.

Software Requirement

- Imblearn – Imblearn.Pipeline - Used for building pipeline to merge text and numerical data, scaling, sampling technique.
- Imblearn.Smote used for balancing of the data.
- SVC, MultinomialNB, LogisticRegression, RandomForestClassifier from Scikit-learn library for Python.
- Transformers
  - Bert pre-trained model - bert-base-uncased tokenization
  - Roberta pre-trained model - roberta-base tokenization
  - DistilBERT pre-trained model - distilbert-base-uncased tokenization
  - XLNET pre-trained model - xlnet-base-cased tokenization

Evaluation Metrics

1. Sensitivity (Recall) - the ability of a model to correctly identify fake reviews out of all actual fake reviews present in the dataset.
2. Precision - accuracy of the model in correctly identifying fake reviews out of all reviews predicted as fake.
3. Accuracy - the overall performance of a fake review detection model
4. F1-score or area under the ROC curve (AUC-ROC)

Sensitivity (Recall) : The proportion of true positives that were correctly identified.

$$\text{Formula: } \frac{TP}{(TP+FN)}$$

Precision: The proportion of true positives among the predicted positives.

$$\text{Formula : } \frac{TP}{(TP+FP)}$$

Accuracy: The Ratio of proportion of correctly classified instances.

$$\text{Formula: } \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

Confusion matrix

	Predicted Genuine	Predicted Fake
Actual Genuine	TN	FP
Actual Fake	FN	TP

True Positives (TP): These are the reviews correctly classified as fake by the model.
False Positives (FP): These are the reviews incorrectly classified as fake by the model (i.e., the model predicted them to be fake, but they are actually genuine).
True Negatives (TN): These are the reviews correctly classified as genuine by the model.
False Negatives (FN): These are the reviews incorrectly classified as genuine by the model (i.e., the model predicted them to be genuine, but they are actually fake).

Results and Analysis

Embedding Techniques and their comparison with different Machine Learning Models.

TFIDF Embeddings										
Machine Learning Models					Genuine Review			Fake Review		
	Sensitivity	Specificity	Accuracy%	ROC Area	precision %	recall %	F1 Score %	precision %	recall %	F1 Score %
Logistics Regression Classifier	84.03%	35.52%	79.15%	0.67	92.00%	84.00%	88.00%	20.00%	36.00%	26.00%
Random Forest	83.11%	38.92%	78.67%	0.67	92.00%	83.00%	88.00%	20.00%	39.00%	27.00%
SVM Classifier (Linear)	76.69%	41.11%	73.11%	0.65	92.00%	77.00%	84.00%	16.00%	41.00%	24.00%
Naïve Bayes Classifier	85.46%	35.64%	80.44%	0.69	92.00%	85.00%	89.00%	22.00%	36.00%	27.00%
XGBoost	91.61%	20.15%	84.49%	0.64	91.00%	92.00%	91.00%	21.00%	20.00%	21.00%

Glove Embeddings										
Machine Learning Models					Genuine Review			Fake Review		
	Sensitivity	Specificity	Accuracy%	ROC Area	precision %	recall %	F1 Score %	precision %	recall %	F1 Score %
Logistics Regression Classifier	62.80%	50.89%	61.60%	0.58	92.00%	63.00%	75.00%	13.00%	51.00%	21.00%
Random Forest	83.00%	38.92%	78.67%	0.67	92.00%	83.00%	88.00%	20.00%	39.00%	27.00%
SVM Classifier (Linear)	66.30%	47.30%	64.39%	0.58	92.00%	66.00%	77.00%	14.00%	47.00%	21.00%
Naïve Bayes Classifier	87.48%	27.34%	81.44%	0.63	92.00%	87.00%	89.00%	20.00%	27.00%	23.00%
XGBoost	88.58%	11.57%	80.84%	0.49	90.00%	89.00%	89.00%	10.00%	12.00%	11.00%

Results and Analysis

Embedding Techniques and their comparison with different ML and DL Models.

Embedding Techniques	Deep Learning Models				Genuine Review			Fake Review		
		Sensitivity	Specificity	Accuracy%	Precision %	Recall %	F1 Score %	Precision %	Recall %	F1 Score %
BERT Embeddings	SVM Classifier	99.00%	92.00%	95.55%	92.00%	99.00%	96.00%	99.00%	92.00%	95.00%
BERT Embeddings	MLP Classifier	91.00%	99.00%	95.01%	99.00%	91.00%	95.00%	92.00%	99.00%	95.00%
RoBERTa Embedding	MLP Classifier	92.00%	99.00%	95.71%	99.00%	92.00%	96.00%	93.00%	99.00%	96.00%
DistilBERT Embedding	MLP Classifier	92.00%	99.00%	95.30%	99.00%	92.00%	95.00%	92.00%	99.00%	95.00%
XLNet Embedding	MLP Classifier	92.00%	99.00%	95.37%	99.00%	92.00%	95.00%	92.00%	99.00%	96.00%

Comparison of Deep Learning Models with standard embedding techniques

Deep Learning Models				Genuine Review			Fake Review		
	Sensitivity	Specificity	Accuracy %	precision %	recall %	F1 Score %	precision %	recall %	F1 Score %
CNN Classifier	88.00%	92.00%	84.88%	83.00%	88.00%	85.00%	99.00%	92.00%	95.00%
CNN + LSTM Classifier	83.00%	78.00%	80.18%	79.00%	83.00%	81.00%	82.00%	78.00%	80.00%
RNN LSTM Classifier	90.00%	83.00%	86.49%	84.00%	90.00%	87.00%	89.00%	83.00%	86.00%

Conclusion and Recommendations

- Various Embedding techniques **TFIDF, Glove, BERT, RoBERTa, DistilBERT and XLNET** along with Machine Learning and Deep Learning models were tested and evaluated for Fake Review Detection. It is found that Deep Learning techniques with **BERT and BERT based embeddings** techniques are giving much better results**.**
- These techniques are outperforming traditional embedding techniques such as TFIDF and Glove.
- Processing Time for model training using Deep Learning techniques-based model is significantly large than processing time for model training with Machine Learning models. For eg. Training the machine learning model takes around 15-20 minutes. Whereas training deep learning based model takes around 3-4 hours. This is due to additional complexity involved in neural networks and transformer based pre-trained model.
- Due to resource and processor limitation, selected experimentation were done on different embedding techniques such as **TFIDF, Glove, BERT, RoBERTa, DistilBERT and XLNET**.
- Future work can focus on fine tuning these Embedding techniques in Deep Neural Networks improving the fake review detection methods using **Unsupervised** and **Semi supervised** models.

REFERENCES

1. Opinion Spam and Analysis, Jindal and Liu 2008 : http://www.cs.uic.edu/~liub/FBS/opinion-spam-WSDM-08.pdf
2. Yelp official blog : https://blog.yelp.com/news/how-yelp-protects-consumers-from-fake-reviews/
3. Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews, (Mukherjee et al., 2013) https://www2.cs.uh.edu/~arjun/papers/UIC-CS-TR-yelp-spam.pdf
4. Attention Is All You Need, Vaswani et All 2017 : https://arxiv.org/pdf/1706.03762.pdf
5. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Jacob Devlin et al, Google AI : https://arxiv.org/pdf/1810.04805.pdf
6. Detecting spamming reviews using long short-term memory recurrent neural network framework, (Wang et al., 2018) : https://dl.acm.org/doi/abs/10.1145/3234781.3234794
7. Deep Learning Hybrid Approaches to Detect Fake Reviews, Deshai & Rao, 2023 : https://ischolar.sscldl.in/index.php/ijsir/article/view/220875
8. Attention-based Bidirectional LSTM for Deceptive Opinion Spam Classification, A Salunkhe 2021 : https://arxiv.org/abs/2112.14789
9. Evaluating BERT-based Pre-training Language Models for Detecting Misinformation R Anggrainingsih et al : https://arxiv.org/ftp/arxiv/papers/2203/2203.07731.pdf
10. Roberta: A robustly optimized bert pretraining approach, Yinhan Liu et al, 2019, Facebook AI : https://arxiv.org/pdf/1907.11692.pdf%5C
11. Leveraging Transfer learning techniques-BERT, RoBERTa, ALBERT and DistilBERT for Fake Review Detection Gupta, P., (2021) : https://norma.ncirl.ie/5164/1/priyankagupta.pdf
12. Augmenting fake content detection in online platforms: A domain adaptive transfer learning via adversarial training approach, Ng et al., 2023 : https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=8781&context=sis_research

Q&A

Thanks