Traditional Machine Learning Models Outperforms Deep Learning on Limited Training Corpous, Taking Restaurant Review as Example

Wenpu Zhang wez038@ucsd.edu
Mengjie Chen mec014@ucsd.edu
Ran Ji Raji@ucsd.edu
Lynn Li lil057@ucsd.edu

Abstract

This research explored performance differences between deep learning and traditional models in multi-class classification tasks, analyzing a dataset of 13,144 restaurants with diverse attributes. We compared deep learning models (BERT and LSTM) with traditional methods (TF-IDF and Word2Vec), focusing on free text, numerical, and categorical features. The dataset's small size and bias presented challenges in model performance and generalization. Through exploratory data analysis (EDA) and feature engineering, we prepared the dataset for multi-class prediction. Contrary to expectations, traditional models performed better than deep learning models under the constraints of limited and biased data, highlighting the importance of dataset characteristics in model selection. This study offers insights for the restaurant industry and empirical guidance on choosing machine learning models with limited data.

1 Introduction

Within the realms of machine learning and natural language processing (NLP), deep learning models have garnered widespread attention for their remarkable capabilities in handling complex data. Particularly in multi-class classification tasks, these models have demonstrated an adeptness in effectively capturing intricate relationships within data, thereby enhancing predictive accuracy. However, the question of whether deep learning models retain their advantage when confronted with datasets of limited sample size and inherent bias remains insufficiently addressed. Traditional models, such as TF-IDF and Word2Vec, have proven their efficacy in handling early text analysis tasks, albeit with potential limitations in processing high-dimensional and complex data when compared to deep learning models. This study aims to investigate the performance disparities between deep learning models (particularly BERT and LSTM) and traditional models (TF-IDF and Word2Vec) in tackling a multi-class classification prediction task within the restaurant industry, characterized by pronounced data bias.

Despite the significant theoretical and practical advancements in deep learning techniques, their reliance on large-scale, high-quality datasets poses challenges in practical applications. Notably, when datasets are limited in scale or exhibit significant bias, the effectiveness and applicability of traditional approaches have resurged as subjects of research interest. Through the analysis of a representative dataset from the restaurant industry, this study seeks to validate whether traditional models can compete with, or even surpass, deep learning models under such specific conditions.

This research endeavors to bridge a gap in the existing literature by comparing the performance of deep learning models and traditional models in multi-class classification prediction tasks under specific data conditions. Our findings aim not only to elucidate the relative advantages of different models in specific data environments but also to provide empirical grounds for selecting appropriate machine learning models in scenarios of limited resources or suboptimal dataset conditions.

2 Related Work

Multi-class classification is a text-mining model that identifies and select a single most appropriate label from numerous categories. Instead of multi-label classification (MLC) research provides methods for associating a single instance with multiple labels, such as Binary Relevance (Godbole and Sarawagi, 2004) and Label Power-set (Boutell et al., 2004), multi-class classification tasks emphasize prediction from a ”pick one” perspective, imposing higher demands on a model’s discriminative capabilities.

Text classification requires appropriate text representation methods, and numerous research has been comparing different methods under various scenarios. Traditional methods like TF-IDF have been widely applied to text processing for document classification and information retrieval tasks (Salton and Buckley, 1987). TF-IDF’s effectiveness lies in its ability to capture keyword importance while diminishing common word interference. However, this approach fails to capture contextual information, which may be necessary for certain tasks (Sparck Jones, 1972). Recently, Cheng et al., improved the TFIDF algorithm by introducing information entropy and relative entropy (Cheng et al., 2020) , enhancing keyword extraction accuracy.

Word2Vec is another widely-used text representation model that introduced the concept of word embeddings, far surpassing traditional vector space models in capturing complex semantic relationships between words (Mikolov et al., 2013). Subsequent research has shown that combining Word2Vec with deep neural networks, particularly LSTM networks, can effectively process sequential data and capture long-range dependencies (Sutskever et al., 2014), which is crucial for natural language understanding.

Recently, the BERT model has significantly boosted multi-class lable performance by pretraining on large corpora to capture deep bidirectional context representations (Devlin et al., 2019). BERT and other Transformer-based models (Vaswani et al., 2023) have become new benchmarks in NLP, excelling in complex tasks such as sentiment analysis, question answering, and text summarization.

3 Dataset Analysis

In our dataset, user reviews for 13,144 restaurants were collected, forming a task designed specifically for multi-class classification. The core of the dataset lies in two fields: the ’text’ of the user reviews and the ’label’ of the restaurant cuisine types. These cuisine categories range widely, from ”american (traditional)” to ”italian” and ”asian fusion”.

During the data preprocessing stage, we reduced vocabulary redundancy through stemming and removed all punctuation. We further cleaned the text data by removing English stopwords and any single-character words. This step was implemented using the NLTK library, ensuring that the machine learning models could focus on the core meaning of the vocabulary without interference from irrelevant characters.

Exploratory data analysis (EDA) provided insights into the distribution of cuisine labels. Although the dataset was designed to support multiclass classification, we found significant differences in the prevalence of cuisine labels in the restaurant market as shown in Figure 1: Restaurant Type 1, suggesting a class imbalance issue where certain cuisine labels are more common than others in the dataset. This imbalance could lead to the model overfitting to common classes while neglecting less frequent ones.

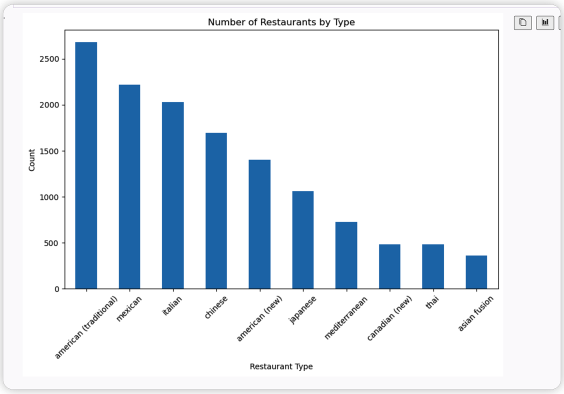

Figure 1: Restaurant Type

Furthermore, word cloud analysis of the review texts revealed the rich dimensions expressed by users. Through the prominent words in the word cloud in Figure 2: Review2, we could see that customers frequently mentioned the quality of ”food”, the quality and ”attitude” of service, and the overall ”experience”. High-frequency words in the word cloud such as ”recommend”, ”favorite”, ”good”, and ”revisit” not only indicated key indicators of customer satisfaction but also provided clues for understanding potential factors influencing cuisine classification.

To evaluate model performance, we split the dataset into training and test sets. Using the `train_test_split` function with `test_size=0.2` and `random_state=42`, we ensured consistency and reproducibility in the data splitting. This step provided us with a sta-

place
food
good
order
one
ordered
great
restaurant
service
pizza
back
drink
table
will
love
time
try
went
well
delicious
location
people
friend
menu
came
price
nice
come
got
best
make
eat
staff
taste
think
fresh
meal
really
want
lot
take
know
even
always
excellent
sushi
salad
chicken
bar
go
friendly
made
sauce
still
little
say

Figure 2: Review

ble benchmark for evaluating the performance of different models on the multi-class classification task.

Based on the in-depth analysis and feature engineering of the text data, we particularly emphasized leveraging the rich semantic information within the text to enhance the models' predictive power. These efforts ensured that our models could effectively identify complex patterns in the vocabulary and accurately predict the single cuisine classification for each review, providing robust support for understanding the diversity of the restaurant market.

4 Predictive Task

This study aims to leverage machine learning models for accurately predicting the cuisine classification corresponding to restaurant reviews, such as ”American (Traditional),” ”Mexican,” or ”Chinese.” Although the possibility of multi-label classification was initially explored, data analysis revealed that most restaurant reviews indeed pointed towards a distinct cuisine. Consequently, our research focus shifted to a multi-class classification problem, more precisely matching each review with its single yet specific cuisine.

Our carefully curated dataset comprises two primary fields: the user's restaurant review ('text') and the corresponding cuisine ('label') for each review. Observational data indicated that reviews typically do not span multiple cuisines, allowing the prediction task to concentrate on identifying the most suitable cuisine from multiple candidates for each review.

In evaluating model performance, accuracy as equation 1 emerged as the primary evaluation metric. Considering the study's focus on multi-class classification, accuracy could intuitively reflect the model's ability to correctly identify the cuisine corresponding to each review.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}} \tag{1}$$

While the F1 score as equation 2 was initially planned to address potential class imbalances, we ultimately did not employ weighted or resampling strategies. We believe that accurately identifying the single cuisine corresponding to a review, coupled with precise model selection and tuning, can effectively address the challenges posed by limited and biased data.

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{2}$$

Overall, this study is dedicated to evaluating and comparing the performance of different machine learning models in the multi-class cuisine classification task. By thoroughly analyzing how models process text data and accurately predict cuisine classifications, we aim to provide insights for restaurant review analysis and guidance for selecting suitable machine learning models in practical applications. Despite facing data limitations and biases, our research outcomes demonstrate effective utilization of machine learning techniques in the face of common challenges.

5 Methodology

Before running the models, we would like to give a brief introduction on the characteristics of each model we choose.

5.1 LSTM

Long Short-Term Memory (LSTM) networks are a specialized type of recurrent neural network (RNN) that incorporate multiple gating mechanisms to regulate the flow of information. These gates – the input, forget, and output gates – enable LSTM models to maintain information over long sequences, making them particularly effective for analyzing textual data.

In our research, we utilized LSTM models to process and analyze textual data from restaurant reviews. The models learn sequential features from the text, capturing semantic flows and sentiment expressions within the reviews. This capability is due to LSTM's structure, specifically designed to recognize and utilize sequential dependencies and patterns in textual data.

An LSTM unit includes the following components at each time step t:

- Forget gate: $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
- Input gate: $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$
- Cell state update: $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$
- Output gate: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
- Final cell state: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$
- Output: $h_t = o_t * \tanh(C_t)$

Where:

- σ denotes the sigmoid function.
- W and b are weights and biases associated with each gate.
- x_t is the input vector at time step t.
- h_t is the output vector at time step t.
- C_t is the cell state vector at time step t.

This configuration allows LSTM models to effectively retain important information and filter out the noise, which is essential for the nuanced task of sentiment analysis in restaurant reviews.

5.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) has a uniqueness of its bidirectional training architecture, enabling the model to learn information from both left-to-right and right-to-left contexts simultaneously.

Built upon the Transformer model, BERT leverages self-attention mechanisms to capture relationships between words. This allows BERT to excel at handling long-range dependencies. The core of the Transformer is its ability to process all words in a sequence simultaneously, contrasting with traditional recurrent neural networks (RNNs) or long short-term memory networks (LSTMs), which process sequences sequentially.

Within the BERT model, pretraining comprises two tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). MLM randomly masks a portion of words from the input sentence, requiring the model to predict these masked words, while NSP tasks the model with determining whether two sentences are consecutive in the original text.

In this research, we will leverage the BERT model to process the textual data of restaurant reviews. BERT's pretraining capabilities allow us to utilize the rich language representations learned from large-scale corpora and then adapt the model to our specific task through fine-tuning. During fine-tuning, the BERT model will be trained on our dataset, adjusting its internal word embeddings to suit the characteristics.

By employing BERT for feature extraction, we anticipate capturing semantic nuances and complex contextual relationships within the text more accurately, thereby enhancing the performance of multi-class classification. BERT's capabilities are particularly well-suited for handling diverse and rich textual data such as restaurant reviews, which encompass evaluations of service, food quality, ambiance, and other multifaceted aspects.

5.3 TF-IDF

TF-IDF is a widely adopted weighting scheme in information retrieval and text mining for evaluating the importance of a term in a document collection or corpus. It is based on two fundamental concepts: term frequency (TF) and inverse document frequency (IDF). Term frequency (TF) measures the frequency of a term's occurrence within a single document, while inverse document frequency (IDF) quantifies the rarity of this term across the entire corpus.

Term Frequency (TF) can be calculated in various ways:

- **Raw TF**: Simply counting the occurrences of a term in a document.
- **Logarithmically Scaled TF**: Using a logarithmically scaled term frequency, computed as $\log(f(t,d) + 1)$, which dampens the linear growth of frequency, preventing frequent terms from dominating the weight.
- **Maximum Frequency Normalization**: Normalizing by $0.5 + 0.5 \times [f(t,d)/\text{MaxFreq}(d)]$, where $\text{MaxFreq}(d)$ is the frequency of the most frequent term in the document.
- **BM25 TF**: This method normalizes TF by accounting for document length, using the formula $(k+1) \times f(t,d)/(f(t,d) + k \times (1 -$

$b + b \times |d|/\text{avgdl}))$, where $|d|$ is the document length, and avgdl is the average document length in the corpus.

Normalization of TF is considered a crucial step in enhancing information retrieval effectiveness.

Inverse Document Frequency (IDF) expresses how much information a term provides, indicating its discriminating power. Common IDF calculations include:

- **Log IDF**: Computed as $1 + \log(|D|/df(t))$, where $|D|$ is the total number of documents in the corpus, and $df(t)$ is the number of documents containing term t.
- **Smoothed IDF**: To avoid division by zero, a smoothed IDF is used: $\log((|D| + 1)/(df(t) + 0.5))$.

The final TF-IDF weight is the product of TF and IDF. This scheme aims to diminish the impact of common terms while enhancing the weight of rare terms, as rare terms are often more indicative of a document's content. Specific to our dataset and multi-class classification task, the TF-IDF model will be employed to transform text data, providing input features for traditional machine learning algorithms.

5.4 Word2Vec

Word2Vec is a popular word embedding technique that uses deep learning to map vocabularies into a vector space, placing words with similar contexts in close proximity. The model's intuition is that a word's meaning comes from the context it appears in; thus, Word2Vec captures semantic and syntactic word relationships by predicting nearby words in a sequence.

The **skip-gram model** predicts surrounding context words for a given target word and optimizes the following objective function:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \qquad (3)$$

where w_t is the target word, c is the context window, and T is the total number of words. This model excels on smaller datasets and handles rare words effectively.

The **CBOW model** predicts the target word from vector averages of context words and maximizes the following objective function:

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_t|w_{t-c},\ldots,w_{t+c}) \qquad (4)$$

which is more efficient on large corpora but less sensitive to rare words.

Training involves optimizing word vectors to maximize the log-likelihood, often using hierarchical softmax or negative sampling for efficiency.

In our research, Word2Vec extracts features from textual reviews, capturing semantic word relationships that assist in classifying restaurant attributes.

6 Experiment Results

We compared two types of text representation and classification models: traditional models (including TF-IDF and Word2Vec), versus deep learning models (including NN and LSTM).

6.1 Text Preprocessing

Before building models, we preprocessed the text in three steps: tokenization, stemming, and stopword removal. The stopword dictionary is acquired from nltk.corpus.stopwords package. After cleaning the text, each document became a list of stemmed words with stopwords removed. For example,

- Original text: [”the”, ”quick”, ”brown”, ”foxes”, ”jumps”, ”over”, ”the”, ”lazy”, ”dog”]
- Text after clean: [”quick”, ”brown”, ”fox”, ”jump”, ”lazy”, ”dog”]

6.2 Deep Learning Methods

We started with deep learning models, where text representations and classification are integrated within one neural network. The activation function is selected as sigmoid because it supports multi-class prediction.

6.2.1 LSTM

We explored the use of Long Short-Term Memory (LSTM) models for multi-class cuisine classification of restaurant reviews. First, through preprocessing steps for the review text, including stemming, stopword removal, and using regular expressions to exclude non-textual characters, we optimized the model's input data. Then, we applied Word2Vec to vectorize the preprocessed text and used NLTK for tokenization, thereby preparing the dataset for model training.

The LSTM model architecture consists of several layers.

- Embedding Layer: This layer converts the tokenized input sequences into dense vectors of fixed size, which are then fed into the LSTM layer. The output dimension of the embedding layer is determined by the embedding_dim parameter.
- LSTM Layer: The LSTM layer processes the input sequences and learns to capture long-term dependencies in the data. We used an LSTM layer with 128 units and applied dropout regularization to prevent overfitting.
- Dense Layers: After the LSTM layer, we added a dense layer with 64 units and ReLU activation function to introduce non-linearity into the model. We also applied dropout regularization to this layer to further prevent overfitting.
- Output Layer: The final layer of the model is a dense layer with a sigmoid activation function, which outputs the probability of each class.

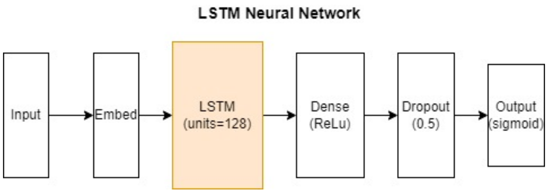

Figure 3: LSTM Model

The model was compiled using the Adam optimizer and binary cross-entropy loss function, and employed an early stopping strategy to prevent overfitting by monitoring the validation loss and stopping training after three consecutive epochs without improvement. After a maximum of 20 training epochs, the model achieved an accuracy of 0.7200 on the test set, showcasing the effectiveness of LSTMs for text classification tasks.

This experimental result emphasized the LSTM model's capability in handling textual data with complex semantic and syntactic structures. By observing the changes in accuracy and loss values across different training epochs, we further validated the importance of the early stopping strategy and the impact of model parameter settings on performance improvement.

6.2.2 BERT with NN prediction

We explored the use of the DistilBERT model for the multi-class cuisine classification task on restaurant reviews. Through carefully designed text preprocessing steps, including stemming and stopword removal, we optimized the text data for better application of deep learning techniques. Then, leveraging the pre-trained DistilBERT model, we transformed the text into word vectors, which were subsequently used for training the deep learning-based classification model.

Regarding model construction, we employed a Sequential model comprising a series of fully connected and Dropout layers. The model includes:

- A 768-dimensional BERT word vectors as input.
- A fully connected layer with 512 units.
- A Dropout layer to reduce overfitting.
- A 256-unit fully connected layer and another Dropout layer.
- Output layer with units units equal to the number of classes, using a softmax activation function for multi-class classification.

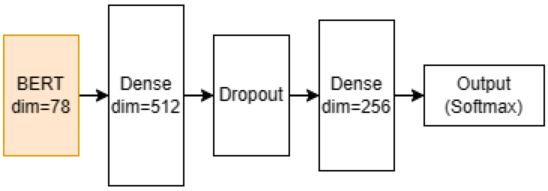

Figure 4: BERT and Neural Network

To optimize the model, we chose the Adamax optimizer and set a custom learning rate of 0.001. The model was trained by minimizing the sparse categorical cross-entropy loss, using accuracy as the performance metric.

After 20 training epochs, our model's performance on the validation set was as follows: macro-averaged F1 score of 0.6153, micro-averaged F1 score of 0.7402, weighted-averaged F1 score of 0.7136, and an accuracy of 0.7402. These results demonstrated the effectiveness of

DistilBERT in handling complex text classification tasks, particularly in the multi-class cuisine classification of restaurant reviews.

6.3 Traditional Methods

Using traditional text representation models, we first transform each documents to vectors, and then build prediction models on these vectors.

6.3.1 TF-IDF

In this study, to accurately predict the cuisine corresponding to restaurant reviews, we employed a combination of the TF-IDF algorithm and logistic regression model for multi-class classification prediction.

First, for text preprocessing, we optimized the input data by removing stopwords, converting punctuation, performing stemming, and utilizing customized NLTK tokenization.

Then, we vectorized the text using TfidfVectorizer, setting min_df to 2 to exclude rare vocabularies, max_df to 0.98 to ignore overly common words, and considering up to 3-word combinations (ngram_range set to (1, 3)) to fully capture the text's semantic information. During the logistic regression model training, we ensured model convergence by setting max_iter to 1,000,000,000, adjusted the regularization strength with C=10, set multi_class to "auto" for automatic strategy selection, used random_state for reproducibility, and optimized training efficiency via warm_start.

This series of parameter configurations aimed to enhance the model's performance in handling the restaurant review data with pronounced bias. Ultimately, our model achieved an accuracy of 0.7649 on the validation set, demonstrating the effectiveness of our approach in the multi-class cuisine classification task and providing valuable experience for future research and practice.

6.3.2 Word2Vec

We adopted the Word2Vec model combined with logistic regression to predict the cuisine classification corresponding to restaurant reviews, showcasing an exploration of deep semantic understanding of textual data. First, we preprocessed the dataset, including stemming and removing stopwords using NLTK, and excluded non-textual characters using regular expressions, ensuring accuracy and efficiency in our analysis.

Next, utilizing the Word2Vec model from the gensim library, we transformed the preprocessed text into word vectors. The Word2Vec model configuration included: sentences as input, vector size set to the length of the longest sentence in the preprocessed text, window size of 5, minimum count of 1, and 4 worker threads. These parameter choices aimed to maximize the model's ability to learn semantic relationships between words from the preprocessed text.

After converting the text to word vectors, we used the average of document vectors to represent the vector for an entire document. This concise yet effective approach allowed us to reduce the high-dimensional features generated by Word2Vec to a fixed-length vector, suitable for subsequent logistic regression model training.

During the model training stage, we employed a logistic regression model for classification prediction, setting the maximum number of iterations to 1000 to ensure model convergence. By using the document vectors as features and cuisine labels as target variables, we split the dataset into training and test sets, with a test size of 20%, and used 42 as the random state to ensure reproducibility.

After training and testing, the model achieved an accuracy of 0.7706 on the test set, demonstrating the effectiveness of combining Word2Vec and logistic regression for the cuisine classification task. This result not only showcased the advantages of word vector-based text representation in capturing semantic information but also validated the capability of the logistic regression model in handling multi-class text classification problems.

7 Result

We evaluated the performance of four different models on the multi-class cuisine classification task, including two traditional text classification methods: TF-IDF and Word2Vec, as well as two deep learning models: LSTM and BERT. The results showed that on the test set, Word2Vec performed the best with an accuracy of 77.06%, followed by the TF-IDF model at 76.49%. Among the deep learning models, BERT achieved an accuracy of 74.02%, while LSTM performed slightly lower at 72.00%. These findings suggest that when dealing with a small dataset exhibiting pronounced bias, traditional models may slightly outperform deep learning models in terms of accuracy. This observation underscores the importance of considering dataset characteristics and model complexity when selecting an appropriate model for text classification tasks.

Model	Type	Test Set Accuracy
TF-IDF	Traditional	76.49%
Word2Vec	Traditional	77.06%
LSTM	Deep Learning	72.00%
BERT	Deep Learning	74.02%

Table 1: Model Performance Comparison

8 Conclusion & Discussion

In our study, we compared the performance of two traditional text classification methods (TF-IDF and Word2Vec) and two deep learning methods (LSTM and BERT) in a multi-class cuisine classification task. The experimental results revealed a key finding: although deep learning models are generally praised for their ability to capture complex feature relationships, traditional methods such as Word2Vec and TF-IDF may provide higher accuracy on smaller datasets. Notably, Word2Vec achieved a slight edge in our experiments.

This outcome serves as a reminder that model selection in practical applications should not solely rely on the novelty or complexity of models but should carefully consider the characteristics of the dataset and the specific task requirements. Furthermore, while deep learning models excel on large-scale datasets, their demand for substantial data and computational resources may limit their applicability in resource-constrained settings.

Our discussion also points to potential avenues for further research, including an in-depth exploration of the reasons behind the strong performance of traditional models on small datasets, as well as the development of new models or strategies to improve the performance of deep learning models in such scenarios. Additionally, future work could investigate hybrid models that combine traditional methods and deep learning approaches, leveraging the strengths of both to provide more accurate predictions for specific datasets and tasks.

References

[Godbole and Sarawagi2004] Shantanu Godbole and Sunita Sarawagi. 2004. Discriminative Methods for Multi-Labeled Classification. In *Advances in Knowledge Discovery and Data Mining*, volume 3056, pages ???, Springer. DOI: 10.1007/978-3-540-24775-3_5.

[Boutell et al.2004] Matthew Boutell, Jiebo Luo, Xipeng Shen, and Christopher Brown. 2004. Learning multi-label scene classification. *Pattern Recognition*, 37:1757–1771.

[Salton and Buckley1987] Gerard Salton and Chris Buckley. 1987. Term Weighting Approaches in Automatic Text Retrieval. Technical report, Cornell University.

[Sparck Jones1972] K. Sparck Jones. 1972. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation*, 28(1):11–21.

[Cheng et al.2020] L. Cheng, Y. Yang, K. Zhao, and Z. Gao. 2020. Research and Improvement of TF-IDF Algorithm Based on Information Theory. In *Advances in Intelligent Systems and Computing*, volume 905, pages 1-67. Springer.

[Mikolov et al.2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

[Sutskever et al.2014] Ilya Sutskever, Oriol Vinyals, and Quoc Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*.

[Devlin et al.2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

[Vaswani et al.2023] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. arXiv preprint arXiv:1706.03762.