Interpretable Multi-Class Text Classification Using DistilBERT and TF-IDF Baselines

Durgesh Premchand Bhirud [2], Debmalya Chatterjee[2], Thanusri Aenugula[2], John McWhirter[1]
[1]*Department of Electrical and Computer Engineering, Texas A&M University*, College Station, TX, USA
[2]*Institute of Data Science, Texas A&M University*, College Station, TX, USA
durgeshbhirud@tamu.edu, debmalya0132@tamu.edu, thanusriaenugula@tamu.edu, jmcwhirter1608@tamu.edu

***Abstract*—This project uses the DBpedia 14-class ontology dataset to investigate text classification and model interpretability. Two different modeling techniques were used: a transformer-based DistilBERT model optimized for sequence classification, and also a conventional TF-IDF representation combined with a linear Support Vector Machine (SVM). Building and assessing classifiers that could correctly predict Wikipedia-derived article categories was the main task. Interpretability techniques were then used to analyze model behavior.**

Our findings demonstrate a distinct performance difference between contemporary transformer architectures and traditional machine-learning techniques. The DistilBERT model consistently performed better than the TF-IDF + SVM baseline, achieving test accuracy values of more than 99%. In addition to these metrics, we visualized token-level contributions using LIME, which revealed variations in the decision patterns of both models and offered insights into how they make predictions basically to understand why the model performed in a certain way.

Overall, this study shows that when accompanied by explanation frameworks, transformer-based models offer notable improvements for large-scale text classification while preserving interpretability. The main conclusion is that contemporary pre-trained language models can be combined with explanation tools to support transparent and reliable NLP pipelines in addition to providing superior accuracy.

***Index Terms*—text classification, natural language processing, Support Vector Machine, TF-IDF, DistilBERT, LIME.**

I. Introduction

A fundamental task in natural language processing (NLP) is text classification, which is the automatic assignment of predetermined labels to textual inputs. It serves as the foundation for many different applications, including information retrieval, content filtering, and document organization. For the purpose of evaluating these models, large and varied benchmark datasets are crucial. An excellent testbed for multi-class text classification is the DBpedia Ontology Classification dataset, and this offers an organized subset of Wikipedia abstracts mapped into clearly defined semantic categories. Because models must learn fine-grained semantic difference across numerous categories rather than just two or three labels, multi-class classification in particular is very difficult.

A. Dataset Overview

There are 14 mutually exclusive classes in the dataset used for this project. In each sample a title and an abstract taken from Wikipedia are included. The DBpedia corpus has 560,000 training samples, which is further split to have 56,000 validation samples and 70,000 test samples, with the samples split equally amongst the classes.

B. Project Objective

The objective of this project is to create and contrast two different modeling strategies for classification: A transformer-based model, DistilBERT, optimized for sequence classification, and a traditional baseline utilizing TF-IDF vectorization in conjunction with a linear Support Vector Machine (SVM). In experiments, confusion matrices were generated, misclassifications analyzed, accuracy and F1-scores assessed, and LIME was used to interpret model predictions. A detailed understanding of how different architectures learn from textual data is provided by the combination of interpretability and performance metrics.

C. Literature Review

Due to their excellent baseline performance and efficiency, traditional text classification techniques like TF-IDF in conjunction with SVMs have been extensively utilized. Joachims [1] demonstrated early success of SVMs for high-dimensional text categorization, while Manning et al. [2] highlighted the importance of term-weighting schemes such as TF-IDF. More recently, transformer-based models such as BERT and its distilled variants have redefined the state of the art in multiple NLP tasks. Devlin et al. [3] introduced BERT, showing substantial improvements in the classification benchmarks. Distillation approaches such as DistilBERT preserve these advantages while reducing computational cost. Model explainability has also become crucial in modern NLP; Ribeiro et al. [4] introduced LIME as a model-agnostic tool that explains predictions by approximating local decision boundaries, allowing human-interpretable insights into complex models.

II. METHODOLOGY

This section details the implementation specifics for the project. The process is divided into several key steps for the evaluation and analysis of the final model. The First part describes data set preprocessing, including cleaning, transforming, and organizing raw data to make it suitable for model input. Next comes Data Analysis, EDA, where the dataset features to be understood regarding its distribution and

structure were examined. Later, we discussed how to construct a Baseline Model which sets a benchmark for classical Text classification by combining Support Vector Machine(SVM) with TF-IDF vectorization. After that, we described the pre-trained language model DistilBERT transformer, optimized using state-of-the-art deep learning approaches for classification by performance. Finally, we integrated Explainability using LIME to provide insights into the model's decision-making process and improve interpretability.

A. Data Preprocessing

The dataset was cleaned and made ready for model training during the preprocessing stage. The main goal was to make sure the dataset was correctly organized, error-free, and optimized for both transformer-based and traditional machine learning models.

Initially, we created a single text feature by combining each article's title and abstract. Because both the title and the abstract offer vital context for the classification task, this combination was significant. To avoid any null values appearing in the final text, any missing values in the title or text were substituted with empty strings.

To make sure that only rows with valid, non-empty text were kept in the dataset, we eliminated rows with blank or empty text entries after combining the title and abstract. In order to prevent possible problems during tokenization and model training, this step was essential.

Additionally, by deducting 1 from each label value, the dataset's labels which originally ranged from 1 to 14 were modified to adhere to a 0-based indexing scheme. Consistency with common Python indexing conventions was guaranteed by this modification.

B. Exploratory Data Analysis

First, the class distribution of the DBpedia training set was examined. The label histogram shows that all 14 classes were approximately balanced, with each category contributing a similar number of samples. This balance is appropriate for impartial assessment and multi-class training and it can be seen in Fig 1.

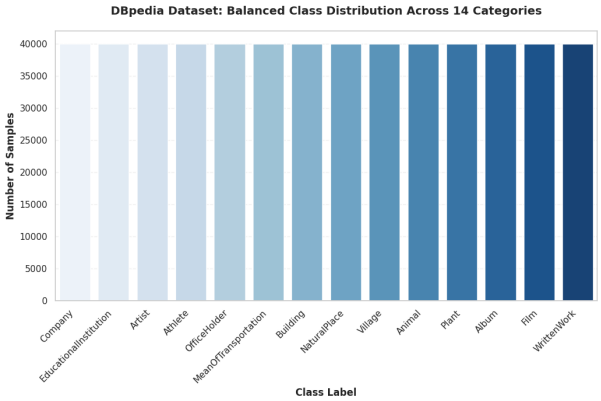

Fig. 1. Number of samples across classes

After that, text-length statistics were investigated by calculating the number of words per example, summarizing character and sentence lengths, and visualizing their distributions. As shown in the Fig 2, the vast majority of samples fall in a relatively narrow range roughly between 25 and 75 words indicating that most abstracts are short to medium-length descriptions.

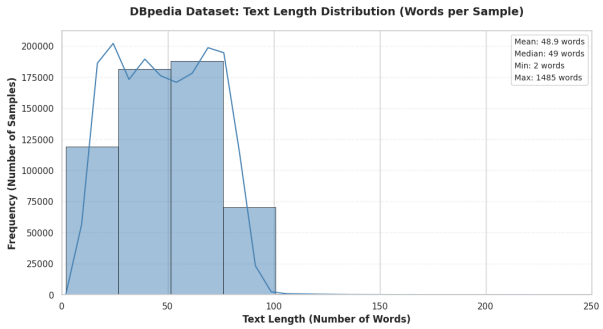

Fig. 2. Text Length Distribution

To get a better sense of how the DBpedia categories are structured, we visualized 2,000 samples by converting their TF-IDF embeddings into two-dimensional space using t-SNE. The plot shows that many classes naturally form their own clusters, suggesting that the dataset is fairly separable based on simple lexical features. Still, some overlap is expected categories like Plant and Animal often share terms such as "species" and others like Artist, OfficeHolder, and WrittenWork have similar language patterns. Overall, the visualization suggests that while TF-IDF captures meaningful distinctions between classes, some categories are still too linguistically close, which is part of why a context-aware model like DistilBERT performs better. (See Fig. 3.)

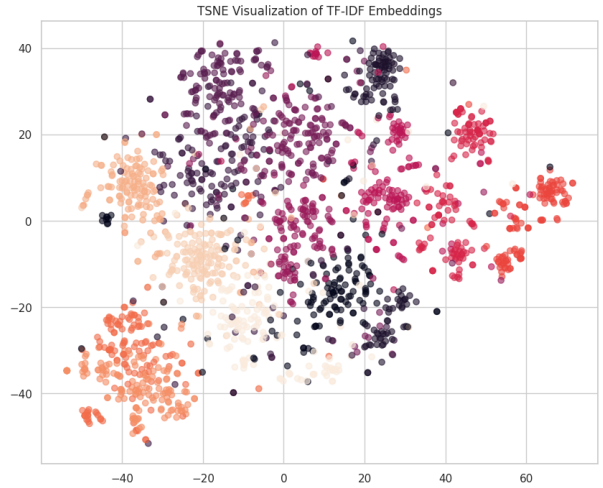

Fig. 3. TSNE Visualization of TF-IDF Embeddings showing overlap

C. Model Architecture

Figure 4 provides an overview of how our models learn from the DBpedia dataset. The input text formed by combining

each article's title and abstract enters the Mapper as Data x, which represents either the TF-IDF + SVM baseline or the DistilBERT classifier. The SVM model uses TF-IDF features to predict classes, while DistilBERT processes the text contextually to output class probabilities. The ground-truth labels (Data d) are compared with the model's predictions to compute error, using hinge loss for SVM and cross-entropy loss for DistilBERT. This error is fed back through each model's learning algorithm AdamW optimization for DistilBERT and the SVM's built-in optimization routine to update parameters. Finally, we apply LIME as an interpretability layer, highlighting the key words that most influenced each model's decisions.

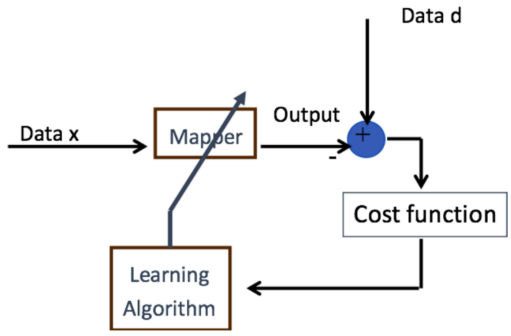

Fig. 4. Model Architecture and Training Workflow

III. Baseline Model: TF-IDF + SVM

A. Architecture of TF-IDF + SVM

The TF-IDF + SVM model's architecture is based on a typical text-processing pipeline that includes a linear classifier and a feature extractor. Each input document is converted by the TF-IDF module into a high-dimensional sparse vector with entries that match weighted n-grams. The vectorizer creates a representation that is ideal for linear decision boundaries by controlling vocabulary size using the maxfeatures parameter and capturing local lexical patterns using adjustable n-gram ranges. On top of this representation, the Linear SVM maximizes the classification margin by using the vast feature space to divide classes. This architecture efficiently captures discriminative vocabulary patterns without requiring contextual embeddings because DBpedia categories frequently contain distinctive lexical cues, such as characteristic words for movies, species, places, or organizations.

B. Implementation of TF-IDF + SVM

A TF-IDF vectorizer was built using stopword-aware tokenization, unigram–bigram feature extraction, and a controlled vocabulary to handle dimensionality. The resulting TF-IDF matrix was then used as input to train a Linear SVM classifier on the labeled samples from the DBpedia training subset. In order to balance predictive performance with computational efficiency, hyperparameter tuning concentrated on the SVM regularization strength and the vocabulary size of the TF-IDF encoder. Accuracy, precision, recall, and F1-score were used to evaluate the model on both the validation and test splits. To find class-specific misclassification patterns, a confusion matrix was also looked at. Before the switch to transformer-based models, this pipeline offered a solid classical baseline. The workflow is conceptually organized as follows: text is transformed into TF-IDF vectors, labels are linked, the Linear SVM is trained, and predictions are generated for instances that have not yet been seen.

IV. DistilBERT

A. Architecture of DistilBERT

Bidirectional context and semantic dependencies in the text are captured by a transformer encoder with multi-head self-attention layers in the DistilBERT architecture. DistilBERT uses WordPiece tokenization to embed subword tokens and encode them into dense contextual vectors, in contrast to TF-IDF, which views documents as unordered bags of words. Above the encoder stack, the classification head of the model uses a feed-forward layer to map the [CLS] token representation to the 14 DBpedia classes. DistilBERT offers a good trade-off between expressiveness and efficiency, with a hidden size of 768 and less depth than BERT. In order to balance contextual completeness with training speed and make sure the model gets enough information from the combined title and abstract, the maximum sequence length was set to 128 tokens.

B. Implementation of DistilBERT

After tokenizing each input document into subword units, the maximum sequence length of 128 was either padded or truncated. In accordance with accepted best practices for transformer fine-tuning, the model was optimized using the AdamW optimizer with a learning rate of 2e-5, a batch size of 8, a weight decay of 0.01, and three training epochs. During training, the transformer encoder and the classification head were both updated using cross-entropy loss. A multi-class confusion matrix that showed prediction trends for each of the 14 classes was used to support the evaluation, which included accuracy, macro and micro precision, recall, and F1-score. The finished model was saved after training and then reloaded for interpretability and inference analysis. LIME was used to improve transparency by identifying significant tokens and revealing how DistilBERT internally weighted textual features during classification.

V. Model Explainability with LIME

To better understand how each model makes its predictions, we applied LIME as a post-hoc interpretability technique. LIME works by generating local perturbations of an input and training a simple surrogate model to approximate the classifier's behavior around that instance. For TF–IDF + SVM, LIME highlights the specific n-grams and keywords that most strongly influence the decision boundary. For DistilBERT, it identifies the subword tokens that contribute most to the contextual embedding used for classification. This allows us to compare how each model relies on lexical cues versus contextual semantics, offering insight into why DistilBERT reduces misclassifications on overlapping or ambiguous categories.

VI. Result

A. Baseline Model: TF-IDF + Linear SVM

The TF–IDF + Linear SVM baseline was evaluated on the DBpedia ontology classification dataset, which contains 14 balanced and semantically diverse classes. Table I and Table II summarize the validation and test metrics, while Fig. 5 presents the corresponding confusion matrix. Overall, the model demonstrated remarkably strong and stable performance across all categories, achieving a macro-averaged F1-score of 0.99 on both validation and test sets.

Per-class results show consistently high precision and recall, with F1-scores ranging from 0.97 to 1.00. Categories with distinctive lexical patterns such as *Athlete*, *Village*, and *Album*-were classified nearly perfectly, indicating that TF–IDF captures discriminative vocabulary extremely well. The confusion matrix in Fig. 5 further highlights this behavior: off-diagonal errors are sparse and isolated, showing that misclassifications are rare and not concentrated between any specific class pair.

The close match between validation and test performance 56000 vs. 70000 samples demonstrates excellent generalization and suggests minimal overfitting. Both weighted and macro-averaged metrics remained at 0.99, confirming that no class disproportionately influenced the model. Taken together, these results show that the TF–IDF + SVM pipeline forms a strong and reliable classical baseline for large-scale text ontology classification.

TABLE I
Validation Report for TF–IDF + Linear SVM

Class	Precision	Recall	F1-score
0	0.97	0.97	0.97
1	0.99	0.99	0.99
2	0.98	0.97	0.98
3	1.00	1.00	1.00
4	0.99	0.98	0.98
5	0.99	0.99	0.99
6	0.98	0.98	0.98
7	0.99	1.00	0.99
8	1.00	1.00	1.00
9	0.99	0.99	0.99
10	0.99	0.99	0.99
11	0.99	0.99	0.99
12	0.99	0.99	0.99
13	0.98	0.98	0.98
Accuracy		0.99	
Macro Avg	0.99	0.99	0.99
Weighted Avg	0.99	0.99	0.99

B. DistilBERT Fine-Tuned Model Results

The DistilBERT classifier was evaluated on the DBpedia test set of 70,000 samples spanning 14 balanced categories. Table III summarizes the detailed classification metrics, while Fig. 6 presents the resulting confusion matrix. Overall, the model achieved an accuracy of 99.27%, with macro-averaged precision, recall, and F1-scores all equal to 0.9927. Per-class F1-scores ranged from 0.9753 to 0.9987, showing that

TABLE II
Test Report for TF–IDF + Linear SVM

Class	Precision	Recall	F1-score
0	0.96	0.97	0.97
1	0.99	0.99	0.99
2	0.97	0.97	0.97
3	1.00	0.99	1.00
4	0.98	0.98	0.98
5	0.99	0.99	0.99
6	0.98	0.98	0.98
7	0.99	1.00	0.99
8	1.00	1.00	1.00
9	0.99	0.99	0.99
10	0.99	0.99	0.99
11	0.99	0.99	0.99
12	0.99	0.99	0.99
13	0.98	0.98	0.98
Accuracy		0.99	
Macro Avg	0.99	0.99	0.99
Weighted Avg	0.99	0.99	0.99

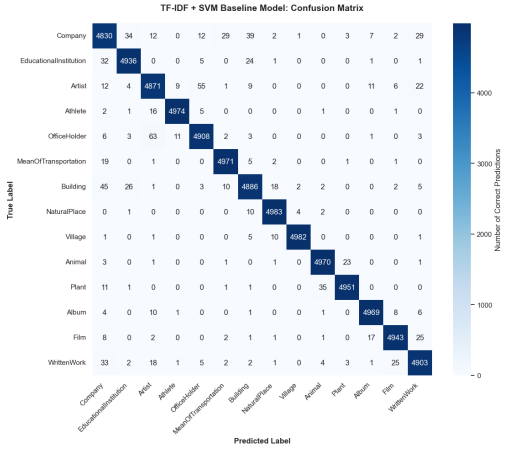

Fig. 5. Confusion matrix for the TF–IDF + Linear SVM classifier on the DBpedia test set.

the model performs strongly even on the most challenging categories.

Several patterns stand out. DistilBERT performed exceptionally well on semantically well-defined categories such as *Athlete*, *Village*, *Animal*, and *Plant* (F1 $>$ 0.997). Categories such as *Company* (Class 0) and *Building* (Class 6) showed slightly lower recall, indicating that these labels occasionally overlap with related domains. The confusion matrix in Fig. 6 remains strongly diagonal, with sparse misclassifications and no recurring confusion pairs, showing that the transformer model handles fine-grained distinctions robustly. Compared to the TF–IDF + SVM baseline, DistilBERT reduces residual mistakes and improves class separation, especially for classes with subtle semantic similarities.

These observations highlight DistilBERT's ability to capture contextual semantics and its advantage over traditional lexical models. Its uniformly high per-class performance and stable generalization demonstrate the effectiveness of transformer-based text encoders for large-scale ontology classification.

TABLE III
DISTILBERT CLASSIFICATION REPORT ON DBPEDIA TEST SET

Class	Precision	Recall	F1-score
0	0.9781	0.9726	0.9753
1	0.9878	0.9910	0.9894
2	0.9843	0.9912	0.9877
3	0.9964	0.9982	0.9973
4	0.9914	0.9866	0.9890
5	0.9946	0.9970	0.9958
6	0.9872	0.9842	0.9857
7	0.9978	0.9970	0.9974
8	0.9980	0.9994	0.9987
9	0.9984	0.9990	0.9987
10	0.9990	0.9964	0.9977
11	0.9974	0.9962	0.9968
12	0.9974	0.9946	0.9960
13	0.9904	0.9948	0.9926
Accuracy		0.9927	
Macro Avg	0.9927	0.9927	0.9927
Weighted Avg	0.9927	0.9927	0.9927

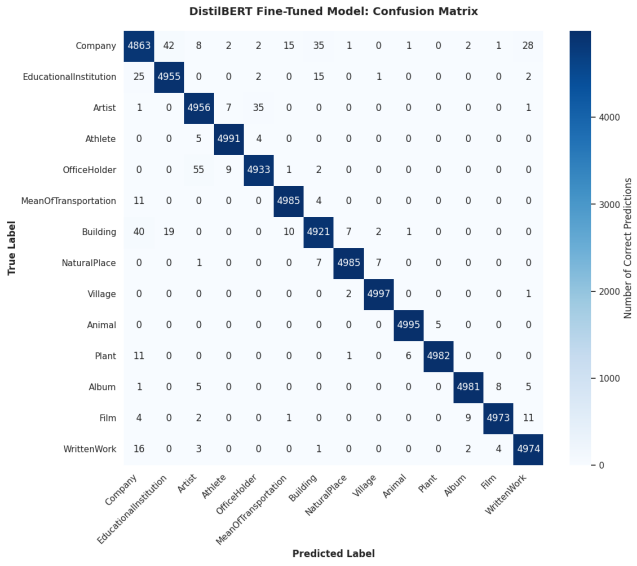

Fig. 6. Confusion Matrix of the DistilBERT Fine-Tuned Model on DBpedia Test Set

TABLE IV
COMPARISON BETWEEN TF–IDF + SVM AND DISTILBERT MODELS

Model	Accuracy	F1-Weighted
TF–IDF + SVM	0.9868	0.9868
DistilBERT	0.9927	0.9927

C. Model Explainability with LIME

To understand how the models make predictions, LIME was applied to both the TF-IDF + SVM classifier and the DistilBERT model. For SVM, LIME showed that decisions were dominated by high-weighted lexical tokens and n-grams, confirming that the classical model relies heavily on surface level keyword matching. In contrast, LIME explanations for DistilBERT highlighted contextually meaningful subword tokens and broader semantic clues rather than isolated keywords. The The transformer's explanations were more stable across examples, showing that its predictions depend on patterns distributed throughout the text, rather than single trigger words. These qualitative differences are in agreement with accuracy improvements reported in Table IV, which further shows that DistilBERT not only outperforms but also makes decisions informed by richer contextual information.

VII. CONCLUSION

The TF–IDF + SVM baseline already performs exceptionally well on the DBpedia classification task, achieving 98.68% accuracy and a 0.9868 weighted F1-score, which highlights how well the dataset's categories can be separated using traditional bag-of-words features. However, as seen from in TABLE II and TABLE III, the fine-tuned DistilBERT model consistently outperforms the SVM across every single class. The improvements, though sometimes small in absolute value, are systematic: per-class F1-score gains range from +0.3% to +1.8%, with the largest benefits appearing in semantically dense categories such as Artist, WrittenWork, and Company. DistilBERT also achieves a higher overall 99.27% accuracy and 0.9927 weighted F1-score, demonstrating its advantage in capturing contextual and semantic relationships that classical models miss. These results clearly indicate that while TF–IDF + SVM is a strong and reliable baseline, transformer-based models provide superior robustness and finer semantic discrimination, particularly for complex real-world text classification tasks.Furthermore, LIME analysis showed that while the TF–IDF + SVM relies primarily on isolated lexical cues, DistilBERT grounds its predictions in broader contextual semantics, reinforcing that its accuracy gains stem from deeper linguistic understanding rather than larger feature sets.

REFERENCES

APPENDIX

[1] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," ECML, 1998.

[2] C. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge University Press, 2008.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," NAACL, 2019.

[4] M. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?": Explaining the predictions of any classifier," KDD, 2016.