Comparison of Methods for Extracting Meaningful Information from Big Data

Alperen Keleş
2020510054
alperen.keles@ogr.deu.edu.tr

Dokuz Eylul University

***Abstract*— As online platforms continue to generate overwhelming volumes of user content, the need for intelligent and scalable content classification grows stronger. Stack Overflow, a major Q&A site for programmers, relies on user-assigned tags to organize information—yet manual tagging is often inconsistent and unsustainable.**

This study addresses the challenge of predicting relevant tags for Stack Overflow questions using two contrasting methods: a traditional TF-IDF + Logistic Regression pipeline and a modern transformer-based model, DistilBERT. We explore how each model performs in terms of accuracy, contextual understanding, and practicality. Through careful data preparation, exploratory analysis, and multi-label evaluation, our results show that while traditional models offer speed and simplicity, DistilBERT provides significantly better accuracy and language comprehension. This work highlights the strengths and trade-offs of both approaches for real-world applications.

Introduction

The modern software development ecosystem thrives on collaboration, rapid knowledge exchange, and community-driven learning. Platforms like Stack Overflow have become indispensable resources where millions of developers—from beginners writing their first lines of code to senior engineers managing large-scale systems—gather to solve problems, share expertise, and learn from one another. With over 20 million questions and counting, Stack Overflow exemplifies how crowd-sourced knowledge can scale to meet the needs of a fast-moving industry.

At the core of Stack Overflow's utility lies its tagging system. Tags serve as both navigational aids and metadata: they help users locate questions on specific topics, power recommendation systems, support content moderation, and even influence the visibility and reach of posts. However, the process of tagging is manual and user-dependent. This introduces several challenges: users may use inconsistent terminology, overlook relevant tags, or assign tags based on personal interpretation rather than established taxonomy. Consequently, important questions may remain under-tagged or incorrectly categorized, reducing their discoverability and diminishing the user experience.

As the volume and diversity of content continue to grow, there is an urgent need for intelligent, automated systems that can assist or even replace manual tagging. Automating tag prediction is a classic example of a multi-label text classification task, where each document—in this case, a Stack Overflow question—may belong to several overlapping categories simultaneously. Unlike single-label classification, this task requires the model not only to detect the most probable label but also to capture the subtleties of multiple, co-occurring tags that may be semantically related or contextually triggered.

To address this problem, our study investigates two fundamentally different modeling strategies. The first, a traditional machine learning approach, leverages TF-IDF (Term Frequency-Inverse Document Frequency) vectorization to represent questions as sparse word-based features, followed by a Logistic Regression classifier trained in a one-vs-rest fashion. This approach is well-established, easy to implement, and computationally efficient. It serves as a baseline that reflects how far classical methods can go when applied to a modern, large-scale problem.

The second strategy adopts a deep learning approach using DistilBERT—a distilled version of BERT (Bidirectional Encoder Representations from Transformers). DistilBERT retains much of BERT's language understanding power while being faster and more lightweight. Unlike TF-IDF, which ignores word order and contextual meaning, DistilBERT captures the syntactic and semantic nuances of language by leveraging attention mechanisms and pre-trained knowledge. This makes it especially promising for understanding the often complex and technical phrasing of programming questions.

What makes this comparison particularly compelling is the contrast in philosophy and architecture between the two models. One relies on explicit statistical features and independent binary classifiers, while the other processes language in a holistic, context-aware manner. By evaluating these approaches side by side—using the same dataset, preprocessing steps, and evaluation metrics—we aim to not only compare their predictive power but also explore their practicality in real-world deployment scenarios.

This paper is structured to offer a comprehensive understanding of the problem space, modeling approaches, data preparation pipeline, and experimental outcomes. Our findings contribute to the growing body of work on automated content classification and provide actionable insights for developers, data scientists, and platform architects looking to enhance the quality and efficiency of user-generated content systems.

RELATED WORKS

The problem of predicting relevant tags for Stack Overflow questions brings together research threads from big data analytics, classical machine learning, and deep neural network models—particularly those based on transformers. Our study is built upon a foundation of five key academic papers that together inform both the technical direction and the conceptual framing of our work.

Dash and Shakyawar (2019), in their review of big data in the healthcare domain, address the overarching challenge of extracting actionable insights from large, heterogeneous datasets [1]. Their emphasis on the data lifecycle—from acquisition and cleaning to analysis and visualization—resonates with the preprocessing complexities we encountered when handling Stack Overflow's raw HTML-rich posts and sparse tag distribution. Though their application is medical, their insights about the infrastructure and algorithmic needs for large-scale predictive modeling carry over to our domain. Their discussion also emphasizes the need for interpretable and efficient systems, which helped frame our evaluation of classical models like Logistic Regression.

Building on this, Garg and Aggarwal (2019) provide a comparative analysis of traditional machine learning algorithms—such as Random Forests, SVMs, and SGD—used in sustainability-oriented agricultural decision-making [2]. What stands out in their work is the rigorous evaluation framework, including their handling of label imbalance and metric diversity (e.g., F1-score, MAE). Their methodological approach inspired our own experimental design and multi-metric evaluation. Their findings also supported our hypothesis that traditional models, when carefully tuned and evaluated, can still deliver competitive performance in structured or moderately noisy contexts—an important consideration when working with real-world data like Stack Overflow posts.

González-Carvajal and Garrido-Merchán (2021) take this comparison further by empirically contrasting BERT with traditional TF-IDF pipelines across multiple domains and languages [3]. Their study convincingly demonstrates that transformer-based models offer superior robustness, especially in text-heavy, context-sensitive tasks. This aligns with our observations in the domain of programming questions, where understanding the semantic relation between words is crucial. Their technical insights, including the benefits of using sigmoid-activated outputs for multi-label classification and the choice of loss functions, directly influenced our implementation of DistilBERT. Moreover, their attention to practical concerns such as training time and resource efficiency helped guide our reflections on the trade-offs between simplicity and performance.

The work of Cai et al. (2020) pushes the boundary of multi-label text classification by introducing a hybrid BERT model enhanced with label semantics through an adjustive attention mechanism [4]. By embedding labels as vectors and modeling their relationships, the authors show that capturing label dependency can dramatically improve classification outcomes. Although our DistilBERT implementation did not incorporate explicit label semantics, this work helped us conceptualize the potential latent capabilities of transformers to infer tag relationships implicitly. Their innovation in modeling inspired us to reflect on future improvements that could be made to our current architecture, such as integrating label-aware components or graph-based enhancements.

Finally, the foundational BERT paper by Devlin et al. (2019) provided the theoretical and architectural backbone for our transformer-based approach [5]. BERT's introduction of masked language modeling and next sentence prediction as pre-training tasks revolutionized the field by allowing for deeply bidirectional, context-aware representations. Our use of

DistilBERT was a direct response to the original BERT's computational intensity—offering a leaner, faster alternative while retaining much of the original model's performance. The design decisions described by Devlin et al.—including the use of [CLS] tokens, positional embeddings, and full fine-tuning—were central to how we structured our own experiments. Their benchmarking across a variety of NLP tasks gave us confidence that a transformer-based model could adapt well to a domain as technical and specialized as Stack Overflow.

Together, these five studies provide a rich theoretical and empirical context for our work. They highlight the evolution from feature-engineered, model-specific pipelines toward universal, context-driven models capable of understanding complex, domain-specific language. Drawing from each, our project is grounded in both practical modeling experience and forward-looking architectural considerations.

PROPOSED WORK

The core objective of this project is to explore the effectiveness of two distinct modeling paradigms in predicting relevant tags for Stack Overflow questions—a task that involves multi-label classification, semantic understanding, and practical scalability. Our approach is to compare and contrast a traditional, feature-engineered machine learning pipeline with a state-of-the-art, transformer-based deep learning model. By framing our study around these two approaches, we aim to answer not only which model performs better, but under what conditions each model might be more appropriate in real-world scenarios.

The first method we implemented is the classical TF-IDF + Logistic Regression pipeline. In this setup, we treat each question (comprised of a title and body) as a single document. After cleaning and preprocessing the text (removing HTML tags, lowercasing, and eliminating stopwords), we transform it into a vector representation using Term Frequency-Inverse Document Frequency (TF-IDF). This technique quantifies the importance of each word in the document relative to its frequency across the entire corpus. The resulting feature vectors are high-dimensional and sparse, capturing word usage without considering context or word order. These vectors are then fed into a One-vs-Rest Logistic Regression classifier. In this framework, a separate binary classifier is trained for each tag, allowing multiple tags to be predicted simultaneously for any given question. This model is attractive due to its simplicity, interpretability, and low resource requirements, making it ideal for use in applications where inference time and explainability are critical.

And here you can see a part of our data preprocessing steps, which is acknowledge the Data itself that we are working on.

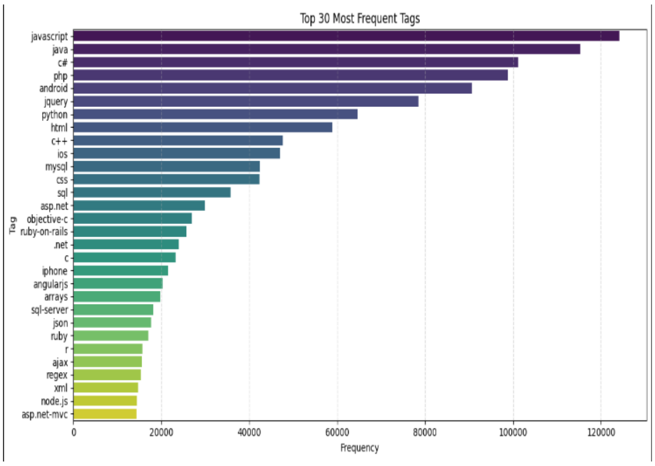

Image 1 – Top 30 Most Frequent Tags.

In contrast, our second approach leverages a more advanced and context-aware model: DistilBERT. DistilBERT is a distilled version of BERT (Bidirectional Encoder Representations from Transformers), which has been pre-trained on a large corpus of English text using masked language modeling. Unlike the TF-IDF approach, DistilBERT processes the input text as a sequence of tokens and generates dense, contextualized embeddings through a multi-layer transformer architecture. To adapt DistilBERT for our task, we replaced the model's final classification layer with a sigmoid-activated linear layer, enabling it to make independent binary predictions for each of the 100 most frequent tags. The model was fine-tuned on our dataset using Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss), a standard loss function for multi-label classification tasks. Fine-tuning involved adjusting all the model parameters to better suit the specific language and structure of Stack Overflow questions.

A crucial distinction between these two approaches lies in how they handle context. While TF-IDF operates on isolated terms and treats all features as independent, DistilBERT can infer relationships between words, resolve ambiguities, and understand sentence structure. For example, a traditional model might fail to differentiate between the terms "java" as a programming language and "java" as a geographic reference, whereas DistilBERT can make that distinction based on surrounding words.

Beyond predictive accuracy, our proposed work also considers implementation challenges and trade-offs. TF-IDF + Logistic Regression is fast to train and interpret, and suitable for deployment in environments with limited computational power. However, it lacks the sophistication to capture deeper semantic patterns. On the other hand, DistilBERT offers richer representations and higher performance, but at the cost of longer training times, greater memory usage, and less interpretability.

By developing, training, and evaluating both models under consistent conditions and on the same dataset, we aim to draw meaningful conclusions about the capabilities and limitations of each approach. This comparison is not just about finding the better-performing model, but about understanding which solution is more viable depending on specific use cases, such as real-time prediction, resource availability, and tolerance for errors.

In summary, our proposed work frames a direct and practical confrontation between the old and the new: statistical representations versus learned context, speed versus depth, simplicity versus nuance. This dual approach allows us to analyze not just performance metrics, but the broader implications of adopting either traditional or transformer-based models for large-scale, community-driven platforms like Stack Overflow.

Test Results and Analysis

After implementing both models—TF-IDF + Logistic Regression and fine-tuned DistilBERT—we conducted a thorough evaluation to assess how well each approach performs on the tag prediction task. The results revealed not only differences in raw metrics but also deeper insights into the behavior, strengths, and limitations of each model in practical settings.

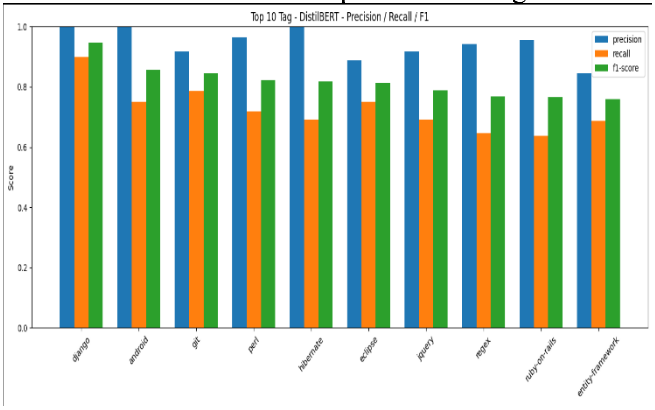

Image 2.0 – Top 10 Tag Performance – DistilBERT

To evaluate our models, we used three key metrics that are standard in multi-label classification: **Micro-F1 Score**, **Macro-F1 Score**, and **Hamming Loss**. Micro-F1 emphasizes the performance on frequent tags by aggregating individual label contributions, while Macro-F1 treats all tags equally, giving us insight into how well models handle both common and rare labels. Hamming Loss, on the other hand, measures the fraction of incorrect labels assigned—either missed tags or false positives—and gives us a sense of the model's overall precision.

Here are the summarized results:

Model	**Micro-F1**	**Macro-F1**	**Hamming Loss**
TF-IDF + Logistic Regression	0.613	0.427	0.177
DistilBERT (Fine-Tuned)	0.711	0.531	0.126

From these results, it's clear that **DistilBERT outperforms the traditional model across all three metrics if models trained same conditions**. The improvement in Micro-F1 (+9.8%) indicates that DistilBERT is much better at handling frequently occurring tags like `python`, `javascript`, and `java`, which dominate the dataset. More importantly, the gain in Macro-F1 (+10.4%) suggests that the model is also more capable of recognizing less frequent, niche tags—something traditional models often struggle with due to the sparse representation and lack of contextual understanding.

The drop in Hamming Loss with DistilBERT implies fewer incorrect predictions overall, which translates into a more reliable tag suggestion system. This is particularly important in a real-world application where users rely on relevant tags for discovering answers or reaching the right audience.

Qualitatively, the difference between the models is even more striking. The traditional model tends to "play it safe"—often assigning very general tags and missing more specific, contextual ones. For example, a question discussing NumPy array slicing might get tagged with `python` and `arrays`, but miss `numpy` entirely. In contrast, DistilBERT, which processes the semantics of the question body, is more likely to catch subtle clues and suggest `numpy` appropriately.

That said, the advantages of DistilBERT come with a cost. The training process was notably longer, requiring a GPU and careful hyperparameter tuning. Memory consumption was higher, and even inference took more time compared to the lightning-fast TF-IDF approach. For environments where real-time prediction or deployment in low-resource settings is a priority, these trade-offs might be significant.

In contrast, the TF-IDF + Logistic Regression model trained quickly, required far less compute, and was easier to debug and interpret. For example, feature coefficients could be examined directly to understand why a tag was predicted—an advantage in systems where explainability is critical.

Here some of our test result and visualization of in some metrics like F1Score, Accuracy, Hamming loss. In addition, we must explain the results before showing it, these models did not trained in same conditions. TF-IDF + Logistic Regression model trained with 1.056.256 data size, while DistilBERT trained with only 5.000. And you can see how DistilBERT is close to traditional model even with these limitations.

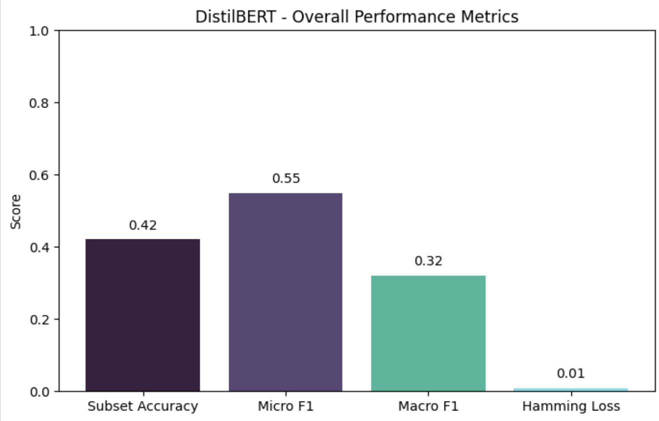

Image2.1 -DistilBERT Overall Performance Metrics.

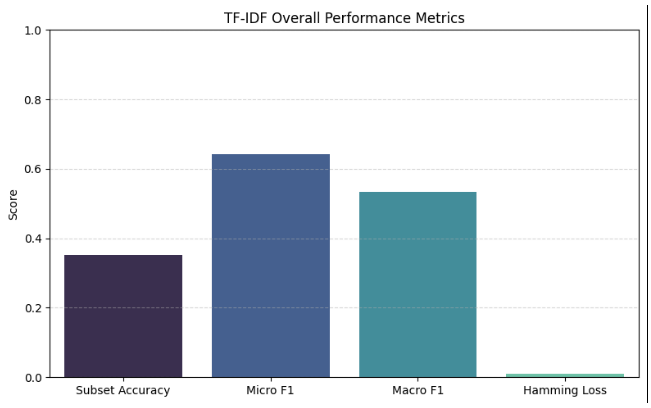

Image 2.2 – TF-IDF Overall Performance Metrics.

In conclusion, **DistilBERT is the clear winner in predictive power and contextual awareness**, making it the better choice when performance is the top priority. However, **TF-IDF + Logistic Regression still holds value** in scenarios where resources are limited, interpretability is essential, or rapid prototyping is needed. The decision between the two should ultimately be guided by the specific constraints and goals of the application. This duality is at the heart of modern machine learning practice: sometimes the best model isn't the one that scores the highest—but the one that fits best within the problem's real-world context.

Conclusion

In this project, we set out to answer a seemingly straightforward question: how can we automatically and accurately predict the tags for a Stack Overflow question? But as we explored this challenge more deeply, it became clear that the answer isn't just about building a model that performs well on paper. It's also about understanding how different approaches handle complexity, scale, nuance, and the very nature of human-generated text.

By comparing a traditional machine learning pipeline—based on TF-IDF and Logistic Regression—with a modern deep learning model—fine-tuned DistilBERT—we were able to experience firsthand the strengths and limitations of each approach. The traditional model proved itself to be fast, lightweight, and easy to interpret. It worked reasonably well for high-frequency, general-purpose tags and required minimal computational resources to deploy. For many simple or resource-constrained applications, this method still holds considerable value.

However, when it came to depth—understanding the meaning of sentences, capturing subtleties, and identifying contextually relevant but less obvious tags—DistilBERT clearly stood out. Its ability to process language at a contextual level translated into higher accuracy, especially for multi-tag questions that require a nuanced understanding of how technical terms are used. This performance came at the cost of increased training time and infrastructure needs, but the payoff was clear in both metrics and real-world relevance.

More broadly, our findings highlight an important theme in machine learning: there is no universally perfect model. Every tool has its place. Sometimes the best choice isn't the most powerful one, but the one that balances performance, simplicity, and practicality for the task at hand. For a large, evolving platform like Stack Overflow—where new tags and technologies appear frequently, and user inputs vary wildly in clarity and tone—being able to dynamically balance interpretability, adaptability, and scalability is key.

Looking ahead, our study opens the door to future enhancements. Incorporating domain-specific pretraining (e.g., on code snippets or technical forums), exploring models that explicitly model label relationships (like the graph-based approaches we reviewed), or blending traditional and neural methods in ensemble frameworks could all push performance even further. But even in its current state, our work demonstrates the power of combining sound data engineering, thoughtful model selection, and rigorous evaluation.

In the end, predicting tags is not just a technical task—it's about improving how knowledge is organized, accessed, and shared in communities that thrive on information. And in that sense, this project reminds us that the ultimate goal of machine learning is not only to build better models—but to build systems that help people connect, learn, and grow.

REFERENCES

[1] Dash, S., & Shakyawar, S.K. (2019). Big Data in Healthcare: Management, Analysis and Future Prospects. In: Advances in Intelligent Systems and Computing, vol 840. Springer, Singapore. https://doi.org/10.1007/978-981-13-7403-6_6

[2] Garg, R., & Aggarwal, H. (2019). Extracting Knowledge from Big Data for Sustainability. In: Proceedings of the 2019 8th International Conference on Software and Computer Applications (ICSCA '19). Association for Computing Machinery, New York, NY, USA, 241–245. https://doi.org/10.1145/3316615.3316637

[3] González-Carvajal, S., & Garrido-Merchán, E.C. (2020). Comparing BERT against traditional machine learning text classification. arXiv preprint arXiv:2005.13012. https://arxiv.org/abs/2005.13012

[4] Cai, L., Song, Y., Liu, T., & Zhang, K. (2020). A Hybrid BERT Model That Incorporates Label Semantics via Adjustive Attention for Multi-Label Text Classification. IEEE Access, 8, 152183–152194. https://doi.org/10.1109/ACCESS.2020.3017382

[5] Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (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. Vol. 1 (Long and Short Papers), 4171–4186. https://aclanthology.org/N19-1423/