Discovering Hidden Subcategories from Supervised Category Learning

Eli Strugo, Matan Aivas, Doron Sharon, Akiva Blackman

The Hebrew University of Jerusalem

Abstract

This work explores our efforts to classify Reddit posts into specific categories and subcategories using Natural Language Processing (NLP). Our team implemented and fine-tuned a DistilBERT model using an adapter, employing a dataset obtained from Kaggle. Post-training, we performed dimensionality reduction and clustering on the generated embeddings to discern the extent to which our model can distinguish subcategories it was not directly trained on.

1. Introduction

Reddit, as a vast online forum, is an abundant source of diverse and multi-themed textual data. In this project, our team has leveraged this data to attempt classification based on categories and subcategories using NLP techniques. The primary aim of this project is to test the efficacy of a fine-tuned DistilBERT model with an adapter and to examine whether the model's embeddings, after dimensionality reduction and clustering, can successfully discern subcategories it was not trained on. The code and datasets for this paper can be found on GitHub.

2. Dataset

The dataset for our project was sourced from the Reddit Self-Post Classification Task (RSPCT) available on Kaggle. This dataset was created with the aim of developing a comprehensive and substantial text classification problem with a large number of classes, devoid of the label sparsity typically associated with similar datasets.

The dataset consists of over 1.013 million self-posts, each sourced from one of 1,013 distinct subreddits, providing 1,000 examples per class. Each data entry includes the subreddit to which the post belongs, along with the title and content of the self-post. This substantial corpus of data was leveraged to explore the ability of our fine-tuned DistilBERT model to distinguish between various categories and subcategories based on post content.

Substantial effort went into the dataset's construction to minimize overlap in content among selected subreddits. For the creation of this dataset, approximately 3,000 subreddits were manually annotated, a selection criterion for which subreddits were included. Alongside this, a top-level category and subcategory were identified for each subreddit, and a reason was provided for the exclusion of certain subreddits from the dataset.

The dataset creators also provide a blog post for a more comprehensive understanding of the dataset, its creation, and its possible use cases.

The provided top-level categories served as the labels for the supervised fine-tuning of our DistilBERT model. The model's ability to encode information about unseen subcategories was then examined using a combination of dimensionality reduction techniques, clustering, and few-shot learning.

3. Methods

Our methodology included several key steps, starting from fine-tuning the DistilBERT model, performing dimensionality reduction, clustering, and finally, evaluating the clustering results with a specific objective. We also utilized the FAISS library for efficient similarity search of nearest neighbors. Here, we provide a more detailed description of each step:

3.1 Fine-Tuning DistilBERT with Adapters

DistilBERT is a transformer-based model that is a smaller, lighter, and faster version of the much larger BERT model. It is designed for speed and efficiency without significantly compromising the performance. Thus, DistilBERT effectively learns a compact representation of the larger BERT model, leading to similar performance but with lower computational requirements.

We utilized this efficient model and fine-tuned it using adapters. Adapters are small modules inserted into the pre-existing layers of the neural network. These modules are trained to adapt to the specific task at hand while keeping the original pre-trained weights of the model intact. This fine-tuning technique offers several benefits: it enables the reuse of pre-trained models without major modifications, reduces the risk of catastrophic forgetting, and requires fewer resources than fine-tuning the entire model.

The combination of DistilBERT and adapter training enabled us to build an incredibly efficient pipeline. The distilled nature of DistilBERT ensured computational efficiency, while the adapter modules allowed for task-specific fine-tuning without a substantial increase in computational requirements. This approach demonstrates a promising strategy for effective large-scale NLP tasks, mitigating the computational cost typically associated with them.

3.2 Dimensionality Reduction and Clustering

For clustering the reduced-dimensional data, we employed the KMeans algorithm. For efficient similarity search and nearest neighbors retrieval, we utilized the FAISS (Facebook AI Similarity Search) library. FAISS enables efficient comparison and retrieval of high-dimensional vectors, aiding our clustering efforts and contributing to the performance of our subsequent zero-shot learning.

3.3 Clustering Objective Evaluation

For evaluating the quality of the clustering in our experiment, we adopted a variety of metrics and methods that provide different insights into the structure and validity of the clusters. Specifically, we used the following scores:

- **Normalized Mutual Information (NMI):** A measure that compares the mutual information between the clustering and the true labels, normalized by the entropy of the labels.

- **Adjusted Rand Index (ARI):** An adjustment of the Rand Index that accounts for chance, providing a measure of the similarity between two data clusterings, irrespective of permutations.

- **Fowlkes-Mallows Index (FMI):** A geometric mean of precision and recall, offering a balance between the precision and the sensitivity of the clustering.

- **The Elbow Method using Distortion and Inertia:** A method for determining the optimal number of clusters by fitting the clustering model to the data for a range of clusters and identifying the ‘elbow’ or the point where adding more clusters does not provide a significant reduction in distortion or inertia.

3.4 Subcategories prediction using Zero-Shot Learning

In the context of our project, we employ a zero-shot learning approach following the clustering process. This method enables the model to classify a given sample to its respective subcategory without having seen any examples of those subcategories during training. By leveraging the information provided by the cluster to which the sample belongs, our approach is particularly suited to our task, where the model must deduce the relationships between unseen subcategories.

Initially, we attempted to classify samples by searching for their nearest neighbors within the entire dataset. This process involved performing clustering on the trainset and assigning each sample from a new data set to a cluster based on calculating the minimal centroid distance to the existing clusters derived from the trainset.

However, we refined this approach by focusing our search only within the assigned cluster for each test sample. For each sample in the test set, we identified the 5 nearest neighbors solely within the cluster it was assigned to. These neighbors were found based on similarity or distance in the embedding space, created by our fine-tuned DistilBERT model. To efficiently find the nearest neighbors, we utilized the FAISS library, enabling swift retrieval of the closest points in our high-dimensional embedding space. This localized search within the assigned cluster reduced the time and computation by a factor of 1000, while still yielding good classification results with approximately the same accuracy.

The final classification for the test sample is then determined by taking a majority vote from the subcategories of the identified 5 nearest neighbors within the assigned cluster. In essence, the most common subcategory among the nearest neighbors is chosen as the predicted subcategory for the given sample. Here, we tried to weigh the neighbors by their proximity, but there was no significant improvement.

4. Discussion

In this study, we sought to understand if a DistilBERT model, fine-tuned on category labels, could generate representations encapsulating information about subcategories it was not explicitly trained on. This examination resides at the intersection of representation learning and clustering, two crucial components of machine learning, with significant implications for tasks requiring large-scale data categorization, such as text classification, information retrieval, and content recommendation.

Our results suggest that even when the model is not directly trained on subcategories, it is capable of capturing some level of semantic information related to these subcategories. This is an important insight as it implies that high-quality representations can carry nuanced semantic details, and these details can be effectively extracted with the right set of analytical tools and techniques.

This finding carries some implications for representation learning. Moreover, it suggests that these representations can be considered as a form of unsupervised learning, generating rich features that can be harnessed for tasks the model was not explicitly trained on.

The use of clustering further underscores the ability to group these representations in meaningful ways, revealing the underlying structure in the data that corresponds to subcategories. It demonstrates the utility of modern clustering techniques like KMeans, and dimensionality reduction techniques as UMAP, in analyzing high-dimensional data, providing a pathway for discovering previously unidentified patterns and relationships.

Our experiment also demonstrates that substantial results can be achieved without a prohibitive computational cost. We used DistilBERT, a smaller, faster variant of the larger BERT model, and trained it using adapters, a technique that allows for task-specific fine-tuning without altering the original pre-trained weights. This approach significantly reduces the computational resources required, making similar experiments accessible to smaller research teams or organizations with limited resources.

5. Results

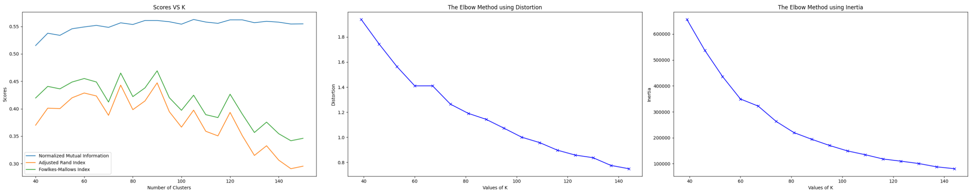

Figure 1: Analisys of choosing k for the KMeans algorithm. Embeddings were generated by the model after fine tuning. k chosen is 80

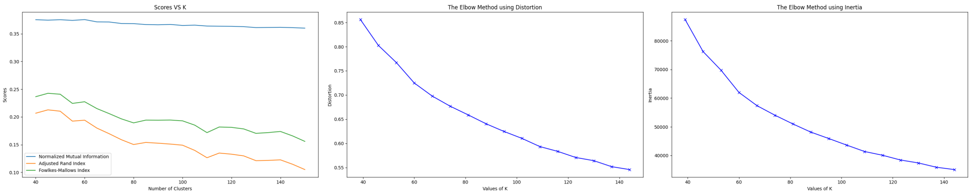

Figure 2: Analisys of choosing k for the KMeans algorithm. Embeddings were generated by the model before fine tuning. k chosen is 60

Table 1: Clustering-based Classification Results for Different Scenarios

Metric	Before Fine Tuning (Associated Cluster)	After Fine Tuning (Entire Dataset)	After Fine Tuning (Associated Cluster)
Accuracy	0.2623	**0.6314**	0.6227
Precision	0.2693	0.6293	**0.6280**
Recall	0.2629	**0.6305**	0.6228