Approximating Manifold Projected Hierarchical Clustering with Neural Nets

Armen Manukyan

Master's Thesis

Advisor: Vahan Arsenyan

Applied Statistics and Data Science

Faculty of Mathematics and Mechanics

Yerevan State University

Yerevan, 2024

Abstract

In this thesis, we hypothesize that it is possible to effectively approximate sequential dimensionality reduction and clustering outcomes via a neural network-based framework, with the objective of enhancing inference speed without compromising precision. Our approach synergistically combines PCA[1]UMAP[2]HDBSCAN[3] algorithms stack into a unified framework that generates labels for the selected data embeddings. We validate the adaptability of our approach through comprehensive experimental evaluation, concentrating on the frameworks ability to consistently detect cluster formations under varying algorithmic conditions and to ascertain the ideal network depth required to meet pre-established performance benchmarks on selected datasets. Although this methodology is applicable to any embeddable data, our experiments specifically utilize textual datasets to meet the urgent demands for effective topic modeling and swift processing.

Contents

1 Introduction

Clustering stands as an indispensable tool in the unsupervised analysis of various domains, particularly textual data, with widespread applications in the fields of news categorization, social media insight extraction, and topic modeling. The ability to dissect voluminous and unstructured datasets into thematically distinct clusters allows for enhanced data comprehension and utilization.

Embedding models like MPNet [4], particularly its 'multi-qa-mpnet-base-dot-v1' implementation provide nuanced semantic representations for a variety of applications. This leap in embedding capabilities enables the effective use of dimensionality reduction and clustering techniques, making it possible to efficiently distill and categorize these intricate representations into meaningful groups.

While the integration of dimensionality reduction techniques like UMAP and clustering algorithms such as HDBSCAN has enhanced our ability to understand and categorize data, these methods are inherently non incremental, requiring refitting to incorporate new data points − a significant bottleneck for dynamic datasets typical in diverse sectors. Our study addresses this challenge head-on, aiming to create a neural network that approximates the clustering capability of these methods for greater adaptability to new data with improved inference speed.

The computational demands of navigating the extensive pipeline from embedding generation to dimensionality reduction and finally clustering are substantial. Existing methods, mostly reliant on CPU-based computations, encounter notable scalability challenges. This is due to the intricate process of generating detailed embeddings, the resource-intensive demands of UMAP for dimensionality reduction, and the rigorous density calculations that HDBSCAN requires for effective clustering. In response, our proposed neural network model seeks to mitigate these computational inefficiencies, offering a streamlined approach to processing large scale textual datasets with improved speed and flexibility.

The integration of predefined labels, such as sentiment scores or upvote counts, is a common practice in the processing and analysis of textual datasets. These labels are often considered proxies for ground truth, serving as indicators for the underlying tone or popularity of the content. However, the utility of such labels is intrinsically limited as they provide a unidimensional view of the text, which may not fully encapsulate the text's multifaceted nature, particularly in the financial domain where context and thematic relevance play critical roles.

In our research, we posit that textual datasets inherently contain multiple latent clusters that extend beyond sentiment gradations or popularity metrics. These clusters could represent

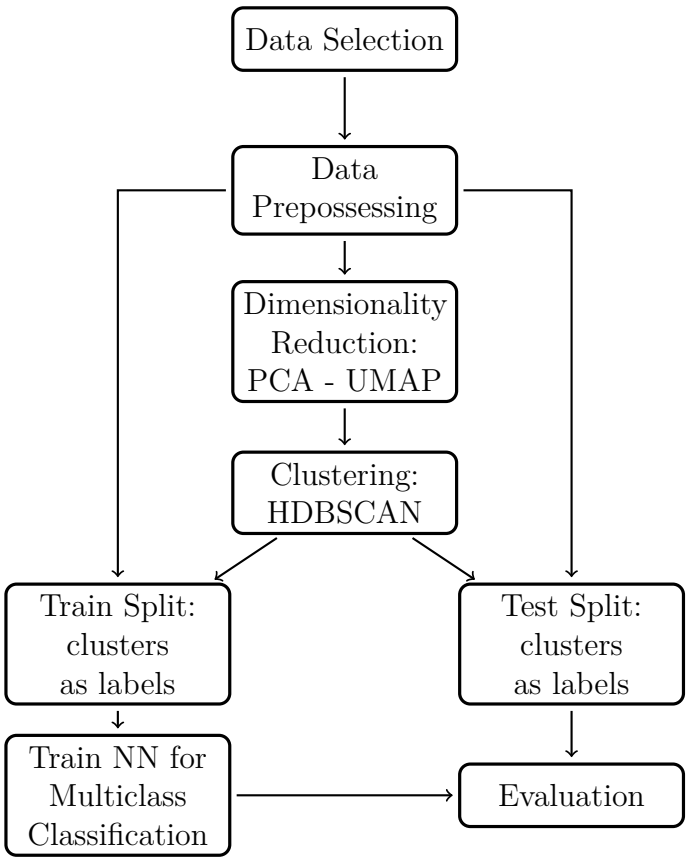

Figure 1.1: The workflow begins with relevant textual datasets selection. The data undergoes Preprocessing to achieve the desired sentence embeddings. Dimensionality Reduction follows, utilizing PCA and UMAP techniques. The data is then subjected to Clustering through HDBSCAN, identifying latent themes and structures. Post-clustering, the dataset is divided into Train and Test Splits, where identified clusters are used as labels to guide the Neural Network training. The final step involves Evaluation of predictions for sentence embedding and cluster pairs.

underlying themes, topics, or perspectives that are prevalent in discourse of consideration. The number of such clusters can be as varied as the dataset itself, potentially ranging up to the full size of the dataset in certain scenarios and influencing the ability to approximate them with the NN model at hand. This perspective paves the way for an exploration into the granular structure of datasets, unlocking a more nuanced understanding that sentiment scores alone may not reveal.

Our clustering approach, therefore, does not conflict with the conventional ground truths but rather complements them by uncovering additional dimensions of analysis. By utilizing unsupervised learning techniques to discern these latent clusters, we can potentially enrich the predictive models and analytical tools used particularly within textual analysis. The resulting clusters from our methodology can serve as an alternative or an adjunct set of labels, offering a richer, more complex view of the data that could enhance subsequent machine learning tasks, including the classification task that we aim to solve with these obtained clusters.

The pipeline delineated in Figure 1.1 encapsulates this approach, ensuring that the process from data preprocessing to neural network classification is not only cognizant of the existing labels but is also proficient in identifying and leveraging the intrinsic clustering within the data. This dual awareness ensures that our model is grounded in the practical realities of datasets while being ambitiously forward-looking in its quest to extract deeper, more informative insights from the textual corpus.

While our experiments are initially tailored to the textual domain, the underlying principles and techniques of our approach have broad applicability across various data domains. This adaptability suggests that our model could be effectively extended to other domains such as audio, video, and sensor data, where rapid and dynamic clustering inference is similarly crucial.

2 Related Work

Dimensionality reduction stands as a cornerstone for managing high dimensional data, such as sentence embeddings. The necessity of transforming these dense, multidimensional vectors into a more manageable, lower dimensional space is of high importance for enhancing computational efficiency and algorithmic performance. Two significant methods in this arena are PCA and UMAP, each with its own set of advantages and unique applications.

2.1 Textual Embeddings

In the landscape of NLP, embedding models such as MPNet have marked a considerable advancement in capturing the nuanced semantic meanings of text. This progress enables improved management of subsequent challenges like semantic search or document clustering.

MPNet, advances beyond previous embedding models by combining *Masked Language Modeling (MLM)* used by BERT[5] with *Permuted Language Modeling (PLM)* from XLNet[6]. It overcomes their limitations by utilizing the dependency among tokens predicted through *PLM* and adding auxiliary position information, offering a comprehensive text representation. The shift from traditional embedding techniques is significant. Earlier models relied on *MLM* without accounting for the interdependency among masked tokens. XLNet introduced *PLM* to capture this dependency but lacked complete positional information during pre-training. MPNet integrates the strengths of both, providing a unified and effective embedding method. MPNet's detailed embeddings, combined with proper dimensionality reduction and clustering has the potential for discovery of meaningful categories for textual data.

2.2 Dimensionality Reduction for Textual Embeddings

PCA is a linear dimensionality reduction technique that simplifies data by transforming it into principal components. These components are linear combinations of the original variables, particularly if v is an eigenvector (principal component), the transformed data along this component is Xv, which is a linear combination of the columns of X. The first principal component is the direction that maximizes the variance of the projected data. Each subsequent component, in turn, maximizes variance under the constraint that it is orthogonal to the preceding components. PCA is considered a good choice for its simplicity and effectiveness in reducing dataset dimensions, making it an excellent preliminary step for data analysis. However, its linear nature means it might not capture complex relationships within more intricate datasets, such as textual.

Unlike PCA, UMAP is a non-linear method that excels in maintaining both the local and global data structure. This allows the algorithm to be very competetive in complex domains such as data from simulations detailing biomacromolecules' interactions over time [7]. Given UMAP's demonstrated efficacy in preserving intricate data structures within

complex domains, it naturally motivates an exploration of its applicability and potential advantages in the textual data domain, where maintaining semantic relationships is equally crucial.

UMAP employs a non-linear method to reveal intricate structures embedded within sentences, offering a distinct advantage for datasets that require the maintenance of relational subtleties. Central to UMAP's methodology is its focus on local structure preservation. It achieves this by constructing a high-dimensional graph based on local neighborhood relationships, ensuring that semantically akin sentences are proximate in the embedded space. The process is initiated by creating a weighted graph for each data point, connecting it to its k nearest neighbors to mirror the local configuration. The connections, or edges, are assigned weights that represent the similarity between data points, calculated using a local Gaussian kernel as follows:

$$w_{i,j} = \exp\left(-\frac{d(x_i, x_j)}{\sigma_i}\right)$$

Here, $d(x_i, x_j)$ denotes the distance between points x_i and x_j, and σ_i is a parameter responsible for the scale of the distances. Through its optimization process, UMAP aims to minimize the disparity between the high dimensional and low dimensional representations. This ensures that thematically similar sentences or documents are located near each other in the low dimensional space, even when they are distantly placed in the original high dimensional setting. This objective is achieved by minimizing the cross entropy between the high dimensional graph and its low dimensional counterpart, typically employing stochastic gradient descent. The optimization function is given by:

$$C(Y) = \sum_{i,j} v_{ij} \log\left(\frac{1}{1 + a\|y_i - y_j\|^2 b}\right) + (1 - v_{ij}) \log\left(1 - \frac{1}{1 + a\|y_i - y_j\|^2 b}\right)$$

Where $v_{i,j}$ signifies the likelihood of i and j being contextually related in the high-dimensional space, with y_i and y_j as their mappings in lower-dimensional space. The parameters a and b are dynamically determined based on the dataset to balance the focus on local versus global textual contexts. This formulation allows UMAP to effectively encapsulate and retain the essential topological and geometric relationships inherent in high-dimensional textual datasets within a condensed, lower-dimensional space. The paper shows that the overall complexity of the algorithm is dominated by the nearest neighbor search required for the graph construction and is approximately $O(N^{1.14})$ for N data points. Additional limitation of UMAP arises in its application to new or streaming data. UMAP's reliance on a fitted model based on the initial dataset means that adding even a single new datapoint requires refitting the entire model, rendering it impractical for tasks requiring inference with continuously updated data. This aspect is critical to consider in dynamic environments. Our approximation strategy aims to address both the speed and refitting bottlenecks that currently limit the effectiveness of this powerful algorithm.

Employing PCA to initially reduce data dimensions before applying UMAP merges the strengths of both methods. PCA's efficiency and UMAP's nuanced semantic structure preservation can be combined to process embedded sentences more effectively. This approach not only leverages PCA's rapid dimensionality reduction capabilities but also capitalizes on UMAP's detailed structure preservation, thereby providing a balanced method for handling complex sentence embeddings. However, it's important to acknowledge a potential drawback of this approach, where PCA's initial dimensionality reduction might strip away some of the dataset's non-linear intricacies that UMAP could have otherwise utilized.

While both PCA and UMAP offer significant benefits in dimensionality reduction for textual embeddings, their integration into a combined approach aims to optimize the balance between computational efficiency and the preservation of semantic structures. With the already reduced dimensionality, UMAP can possibly concentrate more on uncovering and preserving the nuanced semantic relationships between data points, rather than expending resources on navigating through less informative dimensions. Starting with a lower-dimensional space makes it easier for UMAP to capture the underlying structure, as there's less "empty" space to navigate and fewer irrelevant dimensions to distract from the meaningful patterns, given PCA retains the total variance along different dimensions of initial data. This synergy, however, is not without its drawbacks. The process can be computationally intensive, particularly in real time or dynamic dataset scenarios prevalent in sectors like finance, where data constantly evolves. The challenge lies in maintaining the adaptability of these methods to accommodate new data without exhaustive reprocessing, prompting a quest for more efficient solutions that retain the benefits of both PCA and UMAP while mitigating their limitations.

In the landscape of dimensionality reduction techniques, methods such as t-SNE and manifold learning algorithms represent the toolset available beyond PCA and UMAP. These approaches, each with their distinct mathematical foundations and algorithmic implementations, contribute to a versatile repertoire for handling high-dimensional data. Techniques like t-SNE, renowned for its ability to preserve local structures at the cost of global ones, and manifold learning algorithms, which seek to uncover the underlying manifold structure of the data, provide alternative strategies for addressing the complexness of textual embeddings.

The efficacy of these dimensionality reduction techniques, however, depends on the nature of the dataset and the specific objectives of the analysis. For instance, while t-SNE excels in visualizing clusters within data, it may not always scale well to very large datasets or preserve the global structure as effectively as UMAP. Similarly, manifold learning algorithms, with their diverse assumptions about data geometry, may vary in performance depending on the inherent structure of the data. This underscores the importance of a nuanced understanding of each method's strengths and limitations, guiding their application in a complementary manner to achieve the best possible outcomes.

Furthermore, the comparative analysis of these techniques reveals a common theme: the

balance between preserving local versus global data structures and the trade-off between computational efficiency and the fidelity of representation. The choice of method often boils down to the specific requirements of the task at hand, whether it be the need for rapid preprocessing, the preservation of detailed semantic relationships, or the facilitation of intuitive data visualizations.

This synergy of PCA and UMAP combination not only encapsulates the strengths of both methods, but also sets a precedent for future explorations in embedding analysis and clustering applications. The collaboration between these two embodies a balanced approach, addressing the tradeoffs between computational demands and the quality of dimensional reduction.

2.3 Clustering for Reduced Embeddings

As we aim to approximate actual clustering results, selecting an appropriate method to be used after reducing dimensionality of the textual embeddings is crucial. HDBSCAN, in particular, is highlighted for its capability to efficiently and effectively cluster reduced embeddings from high-dimensional textual data. It s an advancement of DBSCAN, and is adept at identifying clusters of various densities, a feature beneficial for textual data clustering due to the semantic similarities among data points. Unlike traditional methods, HDBSCAN does not require pre-specification of the number of clusters, as it determines clusters based on data density, offering detailed insights into the data's structure.

HDBSCAN can be considered as a modification of the DBSCAN algorithm. The former is a widely recognized clustering algorithm characterized by its ability to discover clusters of arbitrary shape and its robustness to noise. It operates on the premise that a cluster in a dataset is defined by a high density of points within a particular region, surrounded by a region of lower density. Unlike many other clustering algorithms, DBSCAN requires minimal domain knowledge to define its input parameters, making it particularly useful for cases where the true structure of the data is unknown. The algorithm introduces the concept of the ϵ - neighborhood for a point, which encompasses all points within a radius ϵ of a given point. Clusters are formed by connecting points that are closely packed together (within the mentioned radius), while points that lie alone in low-density regions are marked as noise and not included in any cluster. This approach allows DBSCAN to efficiently identify clusters of varying shapes and sizes. The effectiveness of cluster formation depends on 2 parameters: the ϵ - the distance threshold that defines the neighborhood for any point, and $MinPts$ - the minimum number of points required to form a dense region. A point is considered a core point if its ϵ-neighborhood contains at least $MinPts$, a border point if it is in the neighborhood of a core point but has fewer neighbors than $MinPts$, and noise otherwise. Clusters are formed by recursively connecting core points that are reachable from each other, with border points being included in the clusters of their corresponding core

points. Mathematically, the ϵ-neighborhood of a point is defined as:

$$N_\epsilon(p) = \{q \in D \mid \text{dist}(p, q) \leq \epsilon\}$$

here D is the dataset and $\text{dist}(p, q)$ is the distance between our point in consideration p, and any other point q from the dataset.

In contrast, HDBSCAN does not select clusters based on a global ϵ threshold, but creates a hierarchy for all possible ϵ values with respect to $MinPts$ as minimum cluster size. This allows for the discovery of clusters of different densities and makes HDBSCAN well suited at handling data with clusters of varying densities, which DBSCAN might struggle with due to its single global density parameter. The changes introduced by HDBSCAN are the following:

Core Distance - defined as the minimum distance, ϵ, required for a point to qualify as a core point, with this distance being unique to each point. This adaptability allows HDBSCAN to manage clusters of varying densities effectively.

Mutual Reachability Distance - This metric is calculated between any two points as the maximum of their core distances and their actual Euclidean distance. This definition ensures closer association of points within denser clusters compared to those situated outside, enhancing cluster formation. The Mutual Reachability Distance is given by

$$d_{\text{mreach-}k}(p, q) = \max(d_{\text{core-}k}(p), d_{\text{core-}k}(q), \text{dist}(p, q))$$

where $d_{\text{core-}k}(q)$ represents the $CoreDistance$ of point p with respect to $MinPts$, and $\text{dist}(p, q)$ denotes the Euclidean distance between p and q

Condensed Cluster Hierarchy - By pruning the hierarchical tree of clusters, HDBSCAN simplifies the cluster structure to retain only significant clusters. A cluster's significance is determined by its stability across a range of density scales, with stability indicating the cluster's persistence against changes in density thresholds.This ensures that only meaningful clusters are identified, while those formed from noise or insignificant density variations are disregarded.

Thus HDBSCAN's methodological enhancements to DBSCAN allows to dynamically identify clusters across varied density scales, which effectively overcomes the challenge of selecting a global density threshold.

In practice, processing textual data through embedding, dimensionality reduction, and clustering can be streamlined using specialized libraries. One such library is BERTopic [8], which combines these steps into a cohesive workflow. Users can easily select and configure their preferred methods for each stage within a single framework, enhancing the experimentation process. While BERTopic significantly simplifies the workflow, making it more accessible and manageable, it's important to remember that it primarily facilitates the

process without directly solving issues related to processing speed or real-time data inference. Essentially, BERTopic serves as a powerful tool to efficiently navigate through the complex pipeline of generating embeddings, reducing dimensionality, and forming meaningful topic clusters.

2.4 End to end Methods/ Other approaches

In the following we review some of the deep clustering techniques used to untangle complex data representations. While these methods are promising, they often come with inherent challenges such as computational demands, dependencies on data types, and the accuracy of the learned representations.

Autoencoder-based Clustering are renowned for their ability to learn nonlinear mappings. A notable enhancement in this area is the Deep Embedding Network for Clustering (DEN) [9], which improves the performance of Deep Autoencoders (DAE)[10] by introducing two main constraints. The first is the *Locality-Preserving Constraint*, which ensures that embedded features of closely related data points remain similar in the embedded space, thus preserving the clustering structure:

$$E_g = \sum_{i,j \in k(i)} S_{ij} \| f(x_i) - f(x_j) \|^2$$

where $f(.)$ is the embedding function, S_{ij} is similarity measure and $k(i)$ is the set of $k-nearest$ neighbors of point i. And the *Group Sparsity Constraint* , which aims to organize the features of the data in such a way that only a few, but important, features are activated for each cluster:

$$E_s = \sum_{i=1}^{N} \sum_{g=1}^{G} \lambda_g \| f^g(x_i) \|$$

where $f^g(x_i)$ denotes the group of features for the $i-th$ data point associated with the $g-th$ group, λ_g is the weight that promotes sparsity, N represents the number of datapoints and G the number of groups. These enhancements, while improving clustering accuracy, add to the computational complexity, especially for large datasets

Other approaches in the autoencoder category vary in effectiveness depending on the chosen architecture and the dimensionality of the initial feature space, which may not be oriented towards optimal clustering outcomes.

GNN-based Clustering utilize the powerful capabilities of graph structures[11] to process data, making them suitable for scenarios like social networks and biological structures. GNNs update node states iteratively until stability is achieved, adhering to a global transition function based on the fixed-point theorem, which ensures the uniqueness and consistency of the results. Advanced methods in this domain, such as those employing attention mechanisms and gating, refine this process further. For example, the Graph Convolutional Network (GCN)[12] method leverages convolution operations to efficiently aggregate information

across the graph.

Clustering enhancements in GNNs, like DRGC[13] and DLC[14], build on traditional methods by incorporating deep learning to capture non-linear relationships within graph data. Notably, DLC speeds up the process by avoiding costly eigen-decompositions, while Graph Autoencoders excel at embedding graph data, maintaining structural and node similarities to improve clustering accuracy.

However, the efficiency of GNN-based methods hinges on the proper construction of graphs, which can be computationally intensive and may require data domain expertise, making them less suited for quick inference tasks. Compared to autoencoder-based clustering, GNNs excel in handling relational data, providing a distinct advantage, although autoencoders may be more straightforward and less resource-intensive when dealing with non-relational data.

For a comprehensive exploration of these deep clustering methodologies, further details and extensive analysis are available in the seminal work Deep Clustering: A Comprehensive Survey[15].

3 Industry Applications

Our approach can potentially have practical applications in multiple sectors. Below we will discuss some of the applications to demonstrate how the clustering approximation can be integrated to address the need for fast inference in different domains.

3.1 High Frequency Trading

In the dynamic environment of High Frequency Trading (HFT), the ability to swiftly and accurately interpret market-impacting news is critical. Our NN-based method stands out by quickly clustering incoming news articles into themes or sentiments that influence specific trading tickers or industries.

The process initiates with a selectively curated assortment of news sources. The NN is trained on *(embedding, labels)* pairs, where *embeddings* are vectors derived from articles using an embedding model like MPNet, typically in $\mathbb{R}^{768}$, and *labels* are the categories established from initially applying our clustering stack to these embeddings.

Once deployed, the NN can efficiently organize incoming news into clusters that reflect distinct market sentiments or themes, such as "growth potential" across various clusters. This rapid categorization is achieved through fast inference, and based on the NN's metrics, can confidently predict the clustering results as if the new article had been included in the original dataset and processed through the entire clustering algorithm stack.

These insights can then be integrated into trading algorithms, enabling dynamic strategy optimization in real-time, tailored to the detected sentiments.

Incorporating this detailed, NN-driven news clustering into HFT operations can significantly enhance the responsiveness of trading strategies to market movements, which is vital in a field where milliseconds can significantly impact outcomes.

3.2 Call Center Operations

In the field of call center operations, effective management and fast response to customer interactions are very important. Our method can significantly enhance the ability to classify and prioritize customer calls, leveraging voice-based embeddings to streamline response strategies and operational workflows.

The procedure initiates with the collection of transcribed voice data from customer interactions. These transcripts are transformed into voice embeddings using voice recognition models like Wav2Vec 2.0[16], typically resulting in vectors in $\mathbb{R}^{768}$. Afterwards the collection of the obtained voices' embeddings undergoes PCA→UMAP→HDBSCAN clustering algorithm stack to obtain relevant categories. This preliminary categorization gives us the necessary *(embedding, labels)* pairs to initiate the NN training.

Upon deployment, the NN swiftly organizes incoming calls into distinct clusters that represent common concerns or unique cases. This efficient sorting enables customer service agents to quickly address prevalent issues and escalate more complex or unusual cases as needed. Such rapid categorization not only enhances response times but also aids in identifying outliers in customer interactions, potentially signaling deeper issues with products or services.

The clustering insights can further be utilized to refine call routing protocols and adapt training materials, aligning them with the most common challenges and preparing agents for rare or critical concerns. Implementing the NN driven clustering method in call center settings has the potential to improve the efficiency and quality of customer service, ensuring optimal resource allocation and enhancing customer satisfaction.

3.3 Inventory Management

In the environment of inventory management, particularly within warehouses that handle a diverse range of items, rapid and accurate categorization of inventory can be essential to streamline storage and retrieval processes, which are alternatively conducted by warehouse managers. Our NN based clustering method can optimize this task by immediately analyzing and clustering image scans of incoming inventory, enabling the system to recommend optimal storage locations based on these classifications. This approach can insure that similar items are grouped together, enhancing accessibility and aiding efficient warehouse structure.

The process starts with the collecion of high resolution images of relevant inventory items. These images are transformed into detailed embeddings using an image embedding transformer, such as the Vision Transformer[17] for example. These embeddings then undergo clustering algorithm stack to obtain relevant categories, which again will lead us to the necessary *(embedding, labels)* pairs to initiate the NN training.

Once deployed, the system utilizes the NN to assess incoming inventory in real time, efficiently grouping each item into a cluster that conforms to specific storage protocols. The insights gained from the clustering process can refine inventory management strategies, enhancing predictive stocking capabilities and reducing the likelihood of overstocking or shortages. By automating the initial categorization of items upon their arrival, the method significantly reduces the need for manual sorting and placement, thereby improving operational efficiency and reducing labor costs.

Overall, by integrating our clustering approach into inventory management systems, businesses can transform traditional warehouse operations into more responsive, efficient, and cost effective frameworks.

3.4 Other Practical Use Cases

As demonstrated in the preceding cases involving textual, voice, and image data, our neural network based clustering method shows broad applicability across different data domains. The capability to perform rapid clustering holds significant potential for various other industries that can benefit from accelerated data processing. Here are additional domains where the rapid clustering can be of high benefit:

Healthcare Data Analysis: Rapid clustering of patient data, such as medical imaging or genetic sequencing, can aid in quicker diagnosis and personalized treatment planning.

Social Media Monitoring: In the context of social media analytics, quickly clustering user-generated content into thematic or sentiment groups can provide immediate insights into public opinion trends or emerging issues.

Environmental Monitoring: For environmental studies, fast clustering for a data coming from sensors monitoring air quality, water quality, or wildlife movements can possibly lead to immediate response strategies to ecological changes or any other extreme cases.

Educational Resource Management: In the sphere of education, clustering learning materials and student responses can help in quickly identifying suitable educational resources tailored to diverse learning needs and performance levels, which may in turn assist in teaching strategies and course plan recommendations.

Each of these applications showcases the flexibility of the NN based clustering method to adapt to different scenarios where speed and accuracy are of high importance. By extending the capabilities of our approach beyond the initial focus areas, industries can leverage fast, accurate data processing to enhance operational efficiency and decision making processes.

4 Experiments

4.1 Datasets

For the empirical evaluation of our proposed NN based clustering approach, we selected two diverse and substantial textual datasets: the AG News dataset and the TweetEval dataset. These datasets not only vary in the type and complexity of text they contain but also in the intricacy of the classification tasks they entail. This diversity is critical for testing the adaptability and effectiveness of our model across different textual contexts.

4.1.1 AG News Dataset

The AG News dataset[18] consists of approximately 160,000 news articles from over 2000 sources, curated by ComeToMyHead. Structured into four primary categories, it provides a controlled environment to test our model's clustering efficacy. Here, our model's objective is to replicate and refine the dataset's inherent topical clustering, achieved via fitting initial clustering stack to the textual datapoints in isolation.

4.1.2 TweetEval Dataset

In contrast, the TweetEval dataset[19], comprising about 120,000 tweets, poses a complex challenge with its seven heterogeneous tasks, which are irony, hate, offensive, stance, emoji, emotion, and sentiment analysis. Each of these tasks, particularly the emoji classification with 20 distinct primary classes, serves as a multifaceted classification problem reflecting the dynamic and nuanced nature of social media data. Particularly, we have selected to fit the clustering stack on the 100,000 tweets with 20 distinct emojis as classes, to identify latent clusters that may not necessarily align with predefined emojis, and subsequently to evaluate our NN's ability to approximate the clustering results.

4.1.3 Yelp Reviews Dataset

The Yelp Review dataset[20], curated by Xiang Zhang from the Yelp Dataset Challenge 2015, consists of approximately 700,000 reviews distributed across five star ratings. For our analysis, we specifically selected texts corresponding to 5-star reviews to maintain a comparable dataset size with AG News and TweetEval and to identify latent clusters corresponding to the highest rated restaurants or other businesses. This selection of 130,000 reviews allows us to explore the nuances of top rated feedback and how initial clustering can reveal inherent patterns within highly positive reviews, and how afterwards this patterns can be approximated. Each review is accompanied by a label indicating its star rating, and the reviews are formatted with special characters escaped appropriately. By focusing on the highest ratings, we aim to evaluate clustering stack's ability to detect and our NNs

ability to approximate the detected clusters that mirror the qualitative aspects of customer experiences, thus offering insights into the characteristics of top-performing businesses.

In the case of all datasets the actual labels provided along with the texts were not used with the corresponding sentence emmbedings, aiming not to have any possible influence on the text sentence itself.

4.2 Methodological Approach to Neural Network Design

In our experimental setup, we have configured a neural network architecture with a fixed design of five hidden layers, which provides a robust framework for learning complex patterns. The only change in the architecture between experiments lies in the dynamic adaptation of the last layers dimensionality, tailored to align with the number of clusters predicted by our initial dimensionality reduction and clustering stack.

By adjusting only the last layer's dimensionality based on the number of clusters, we aim to retain the structural integrity and learning capacity of the neural network while allowing for sufficient flexibility to accommodate the varied cluster configurations that emerge from the initial stack.

4.3 Treatment of Outliers and Their Impact

A notable observation from the experiments involves the consistent presence of a significant number of outliers in the results of the initial clustering stack fitting, specifically when applying the HDBSCAN algorithm to reduced-dimensional vectors. This occurrence of outliers is intrinsic to HDBSCAN's sensitivity to density variations and its tendency to segregate sparse data points as outliers. In our experiments, these outliers were treated as a distinct class during the neural NN training phase, which had profound implications on the overall model performance.

Treating outliers as a separate class aimed to capture the inherent variability and potential anomalies, providing the NN with a comprehensive learning scope. This approach not only informed the neural network's ability to generalize across varied textual contexts but also highlighted critical areas for further refinement, particularly in enhancing the model's robustness to anomalies and sparse data distributions.

However, the inclusion of a substantial outlier class introduced challenges, primarily related to the imbalance between the classes, as the outlier class often represented a significant proportion of the data points. To address this, we implemented a weighted loss function during the NN training phase to effectively manage class imbalance. The percentage of outliers between experiments and the change along with other parameters is illustrated for three of our datasets in Figures 4.1, 4.2 and 4.3.

4.3.1 Parameter Tuning: HDBSCAN's Cluster Selection Epsilon

To explore the interplay of different labels for same text embeddings, obtained from varying clustering algorithm parameters, and the NN's performance on predicting these labels, we have specifically varied the $cluster - selection - epsilon$ parameter of HDBSCAN. This parameter adjustment influences the merging behavior of the initial clusters, providing a spectrum of cluster resolutions from finer to coarser groupings, without altering other critical parameters such as the number of components in PCA and UMAP.

Figures representing the experimental results for the AG News and Tweet Eval Emoji datasets illustrate how adjustments in HDBSCANs $cluster - selection - epsilon$ correlate with changes in both the F1 score of the neural network and the number of clusters detected from the initial sack fitting.

For the AG News dataset, as illustrated in Figure 4.1, increasing the epsilon value consistently reduces the cluster count from 116 to just 3, while impacting the Macro F1 score, which reaches its peak at an epsilon value of 0.15 before diminishing. This trend highlights an optimal clustering granularity, where the neural network attains its highest classification performance in terms of the F1 score, indicating a balanced trade-off between cluster resolution and the model's generalization capabilities across these clusters

Similarly, for the Tweet Eval Emoji dataset, we observe a performance improvement as the epsilon value increases from 0.05 to 0.2, stabilizing the Macro F1 score while reducing the number of clusters. This trend demonstrates the model's adaptability to varying cluster densities, with an optimal cluster configuration emerging at higher epsilon values.

An intriguing observation emerges when examining the results between the epsilon values of 0.15 and 0.20 within our experimental framework. It is hypothesized that an intermediate epsilon value, potentially leading to a configuration of approximately 20 clusters, might offer a "Close to Peak" performance for our neural network model. Ourur current selection of the $cluster - selection - epsilon$ did yield just a bit more than and just a bit less than 20 clusters within this specified range. Consequently, cluster counts both above and below this target were realized at epsilon settings outside the 0.15 to 0.20 interval. This indicates that an optimal cluster count around 20 clusters might exist, which could enhance the neural network's efficiency in categorizing and deciphering the underlying patterns in the dataset. Such an alignment might mirror the exact correspondence between the inherent clusters and the 20 distinct emoji labels, leading to optimal neural network performance as depicted in Figure 4.2

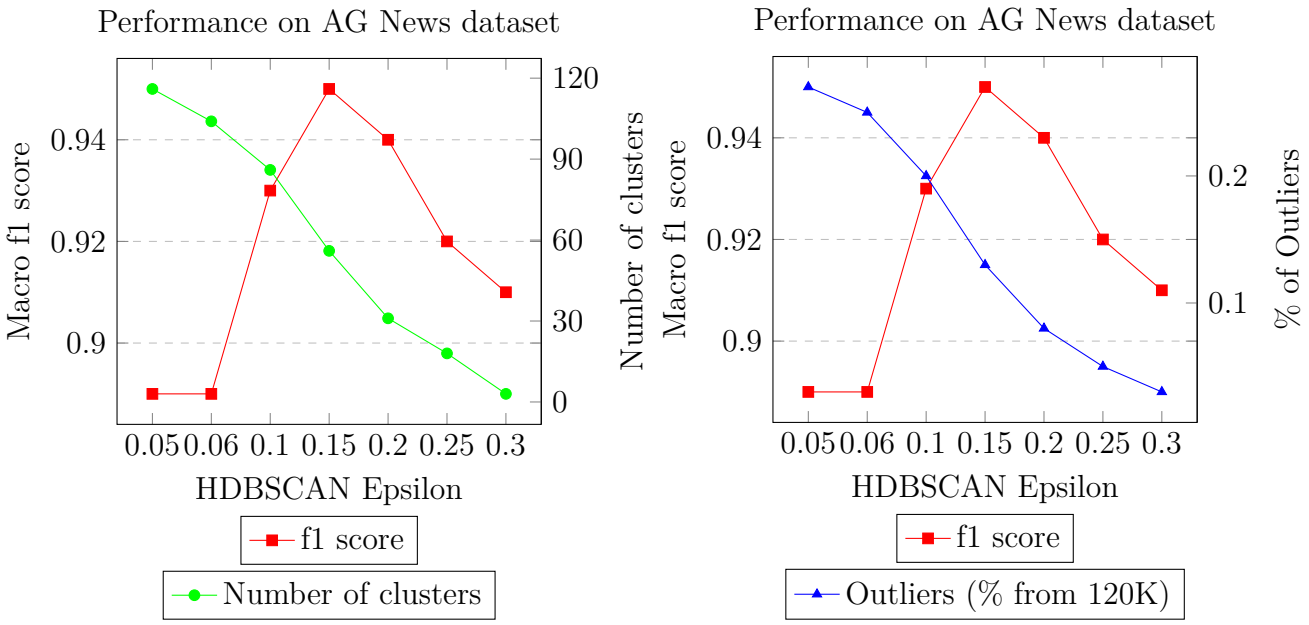

Figure 4.1: Comparison of NN's F1 score with Number of Clusters and Percentage of Outliers based on HDBSCAN Epsilon for AG News Dataset with 120K datapoints

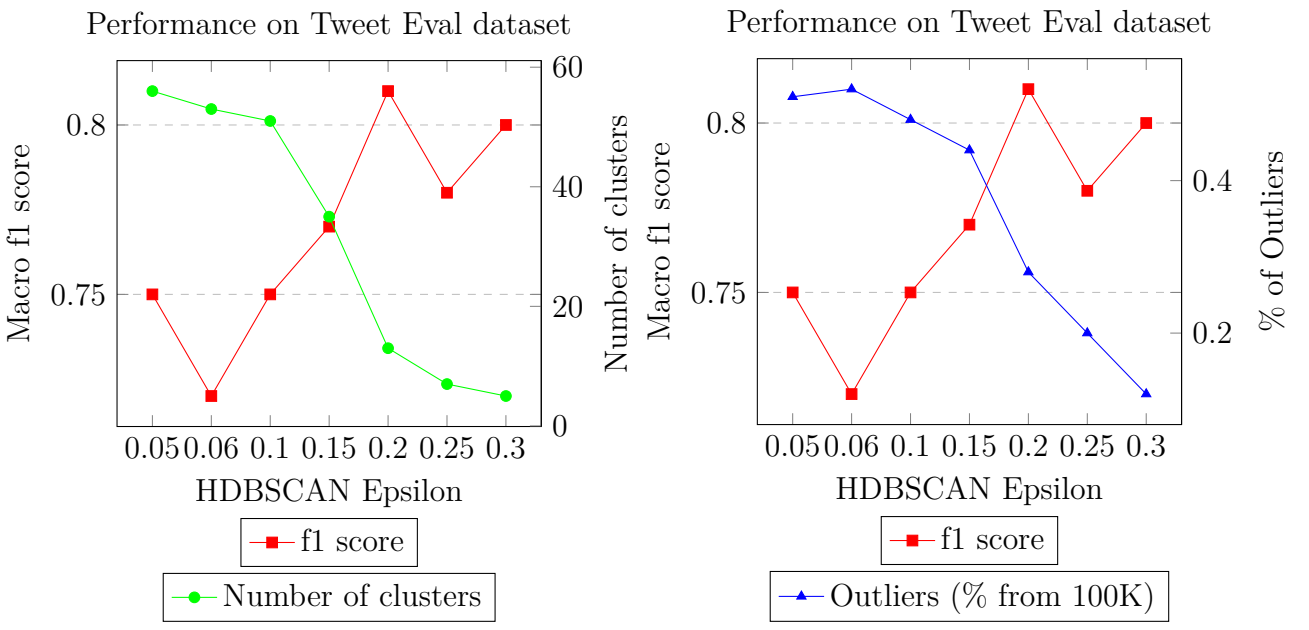

Figure 4.2: Comparison of NN's F1 score with Number of Clusters and Percentage of Outliers based on HDBSCAN Epsilon forr Tweet Eval Emoji Dataset with 100K datapoints

For the 5-star reviews from the Yelp dataset, as illustrated in Figure 4.3, increasing the epsilon value predictably results in a monotonic decrease in both the number of identified clusters and the percentage of outliers. This trend aligns with the inherent nature of HDBSCAN's cluster merging approach. Notably, our network, constructed on clusters derived from various configurations, demonstrates optimal performance at an epsilon value of 0.15 for $cluster - selection - epsilon$.

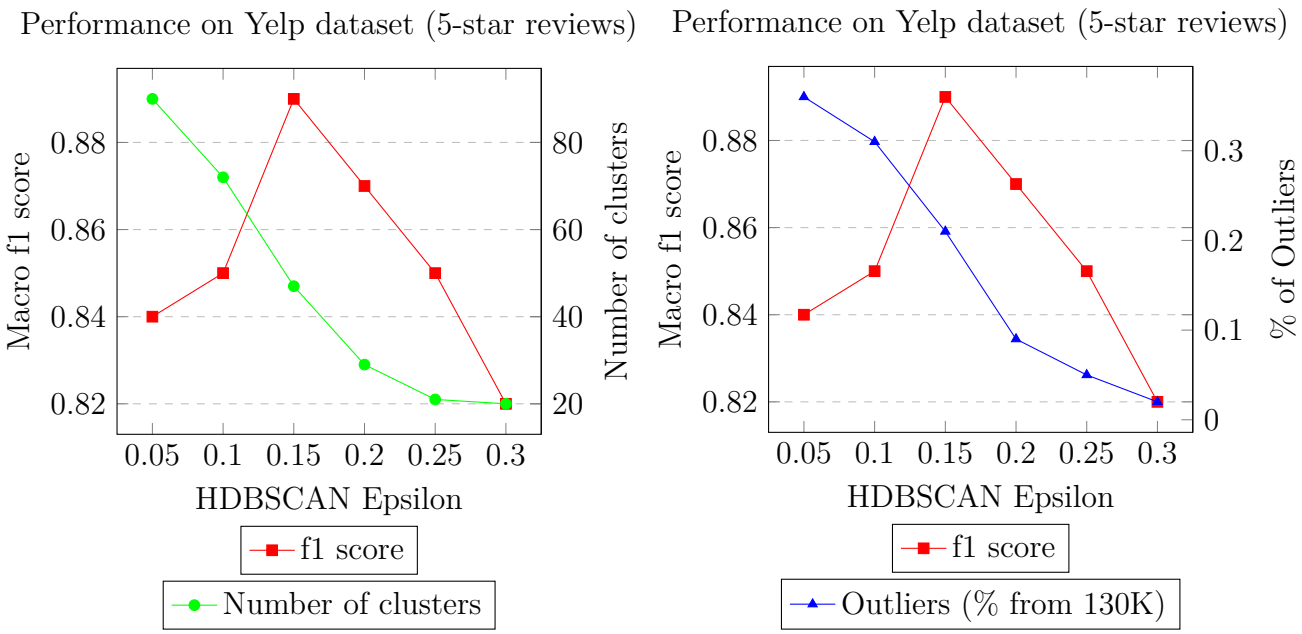

Figure 4.3: Comparison of NN's F1 score with Number of Clusters and Percentage of Outliers based on HDBSCAN Epsilon for Yelp Dataset (5-star reviews) with 130K datapoints

4.3.2 Parameter Tuning: UMAP's reduced dimensions

To fine tune our neural network's efficacy, we investigated the influence UMAP's dimensionality reduction has on clustering outcomes and, consequently, on the network's learning performance. With the optimal $cluster - selection - epsilon$ parameter already identified, we varied UMAP dimensions while keeping other variables constant to isolate the effects of this specific parameter.

Our analysis began with a UMAP dimensionality of 32, which served as our baseline. With this setting, the neural network achieved a macro F1 score of 0.95 and an accuracy of 0.93, identifying 56 distinct clusters in the AG News dataset. Notably, the outlier class, which accounted for 13% of the data, showed a precision of 0.78, a recall of 0.80, and an F1 score of 0.79.

To explore the range of potential dimensionalities, we varied the UMAP dimensions to 16, 48, 64, and 80, and observed the corresponding effects on clustering behavior and classification performance.

These experiments, presented in Tables 4.1, 4.2 and 4.3 illustrate that changes in UMAP dimensions can significantly influence cluster configurations but have a minimal impact on the overall accuracy and macro F1 score of the neural network. This suggests that the network has robust learning capabilities across a range of dimensional reductions, adapting to various cluster densities without substantial performance loss.

For the TweetEval dataset, we observed that increasing the final dimensions prior to clustering from 48 to 64 resulted in a drop in network performance. We believe this is due to the nature

UMAP Dimensions	Clusters	NN Accuracy	NN Macro F1	Outlier Precision	Outlier Recall	Outlier %
16	55	0.93	0.94	0.82	0.73	13.6%
32	56	0.93	0.95	0.78	0.80	12.9%
48	48	0.93	0.94	0.79	0.78	13.1%
64	42	0.93	0.94	0.76	0.79	13.4%
80	46	0.92	0.95	0.76	0.77	13.8%

Table 4.1: Performance Metrics for Varying UMAP Dimensions on AG News Dataset

of tweets in the dataset, which are short and often include symbols. These characteristics potentially introduce more noise in higher dimensions, adversely affecting the network's ability to separate class boundaries effectively.

UMAP Dimensions	Clusters	NN Accuracy	NN Macro F1	Outlier Precision	Outlier Recall	Outlier %
16	18	0.80	0.80	0.76	0.65	32.3%
32	13	0.79	0.81	0.62	0.72	28.2%
48	18	0.79	0.80	0.69	0.72	32.0%
64	12	0.79	0.75	0.68	0.72	32.4%
80	12	0.78	0.76	0.65	0.75	31.6%

Table 4.2: Performance Metrics for Varying UMAP Dimensions on Tweet Eval Dataset

Interestingly, for the Yelp 5-star reviews, similar to the AG News dataset, using 16 UMAP dimensions before cluster generation achieved comparable neural network performance as higher-dimensional embeddings. This optimal dimensionality facilitated faster HDBSCAN clustering without compromising the network's effectiveness, making it the preferred choice for efficient processing.

UMAP Dimensions	Clusters	NN Accuracy	NN Macro F1	Outlier Precision	Outlier Recall	Outlier %
16	47	0.83	0.86	0.81	0.43	19.4%
32	47	0.85	0.89	0.73	0.58	20.6%
48	41	0.84	0.86	0.72	0.60	21.1%
64	38	0.84	0.87	0.70	0.62	21.4%
80	38	0.83	0.87	0.72	0.60	22.0%

Table 4.3: Performance Metrics for Varying UMAP Dimensions on Yelp 5-star reviews

The consistent performance across different dimensions, especially in the treatment of outliers, underscores the NN's ability to generalize well across varied data representations. By maintaining performance levels despite variations in the number of clusters, these findings suggest that within the explored range, dimensionality adjustments do not significantly

hinder the NN's learning ability. This indicates an inherent flexibility in the model to accommodate different levels of data abstraction, which is valuable for practical applications where dimensionality reduction parameters may need to be tailored based on specific storage or computational constraints.

4.3.3 Increasing the Network Depth

In our exploration of network architecture variations, we extended the depth of our neural network from the original five hidden layers by incrementally adding two and then three intermediate layers, resulting in two new architectures. These deeper networks were trained using the cluster-embedding pairs derived from the initial algorithm configurations that produced the best-performing five-layer network for each dataset.

We maintained all other training parameters consistently to isolate the impact of increased network depth. The results indicated a nuanced effect on performance. For the Yelp Reviews dataset, the macro F1 score slightly decreased from 0.89 to 0.87. In the case of the AG News dataset, the performance remained stable with a macro F1 score of 0.95. However, for the TweetEval dataset, the macro F1 score dropped more significantly from 0.81 to 0.75.

These observations suggest that the increased network depth did not consistently enhance performance. Particularly, the presence of outliers in the clustering algorithm's output appears to present substantial challenges for the neural network in learning effective decision boundaries. This issue is more pronounced in the TweetEval dataset, likely due to its inherent complexity and noise.

Interestingly, the AG News and Yelp Reviews datasets exhibited similar responses to changes in both network depth and UMAP dimensionality reductions, hinting at a potential underlying similarity in their data structures or clustering behaviors. This further underscores the need for tailored approaches when optimizing neural network architectures for different types of textual data and clustering outcomes.

5 Conclusion

In this thesis, we have demonstrated the viability of a neural network based framework for approximating sequential dimensionality reduction and clustering outcomes within the domain of textual data, aiming to achieve rapid inference without sacrificing precision of the chosen clustering sequence.

Our experimental evaluation uncovered three key insights into the efficiency of our proposed framework. Firstly, we observed that the optimal range for HDBSCAN's $cluster-selection-epsilon$ parameter was consistently between 0.15 and 0.2 across all datasets. Clusters obtained within this range led to the best performing neural network, ensuring that the labels derived from the clustering stack had the greatest potential to be separable by the same network.

Secondly, our experiments with UMAP demonstrated that varying the final dimensionality before clustering within the range of 16 to 80 did not significantly impact the performance of our network. Consequently, we advocate for using the lower end of this range to expedite the initial fitting process without detriment to the clustering efficacy.

Lastly, we found that increasing the depth of the neural network did not yield performance improvements. The network's efficacy is more influenced by the presence of an outlier cluster, which complicates the application of global performance metrics such as the macro F1 score. This finding underscores that merely augmenting network depth is insufficient to enhance performance and highlights the need for careful consideration of the dataset characteristics and the clustering parameters.

Future work

Building on the findings of this thesis, future research will explore the extension of the proposed neural network-based framework to other data domains, such as audio, images, and video. These complex data types present unique challenges in dimensionality reduction and clustering due to their inherent high-dimensional nature and the subtleties of their contextual information. It would be insightful to adapt and test the sequential stack for these modalities, potentially integrating domain-specific preprocessing steps like spectrogram conversion for audio data and convolutional neural network layers for image data.

References

[1] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2016.

[2] Leland McInnes and John Healy. Umap: Uniform manifold approximation and projection for dimension reduction. *Journal of Open Source Software*, 3(29):861, 2018.

[3] Claudia Malzer and M. Baum. A hybrid approach to hierarchical density-based cluster selection. *arXiv preprint arXiv:1911.02282*, 2019. Available at arXiv:1911.02282.

[4] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding, 2020. Available at arXiv:2004.09297.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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*, pages 4171–4186, 2019.

[6] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, pages 5753–5763, 2019.

[7] Francesco Trozzi, Xinlei Wang, and Peng Tao. Umap as a dimensionality reduction tool for molecular dynamics simulations of biomacromolecules: A comparison study. *Journal of Physical Chemistry B*, 125(19):5022–5034, 2021.

[8] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure, 2020. Available at arXiv:2203.05794.

[9] Peihao Huang, Yan Huang, Wei Wang, and Liang Wang. Deep embedding network for clustering. In *CVPR*, pages 1532–1537, 2014.

[10] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[11] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[12] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[13] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014.

[14] Ming Shao, Sheng Li, Zhengming Ding, and Yun Fu. Deep linear coding for fast graph clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[15] Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, Philip S. Yu, and Lifang He. Deep clustering: A comprehensive survey. *IEEE*. Member, IEEE, Fellow, IEEE, Member, IEEE.

[16] Alexei Baevski, Henry Zhou, Abdel-rahman Mohamed, and Michael Auli. Wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020. Available at arXiv:2006.11477.

[17] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and D. Tao. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 2020.

[18] ComeToMyHead. Ag news dataset, 2015.

[19] Francesco Barbieri, Jose Camacho-Collados, Luis Anke, and Leonardo Neves. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *Proceedings of Findings of EMNLP*, pages 1644–1650, 2020.

[20] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.