Optimized-Embedded Cluster & Tune : Boost Cold Start Performance in Text Classification

Sorn Chottananurak
KAIST
sorn111930@kaist.ac.kr

Abstract

In the field of natural language processing, the challenge of insufficient labeled data in specific domains often impedes the effective fine-tuning of Large Language Models (LLMs) like BERT, a phenomenon known as the *cold start problem*. Prior research on domain-adaption has shown that intertraining on domain-specific data between pre-training and fine-tuning stages can enhance model's performance. Addressing the cold start problem, Cluster & Tune (Shnarch et al., 2022) tackles this by inter-training BERT using pseudo-labels generated from clustering during the intermediate training phase. We propose Optimized-Embedded Cluster & Tune (OECT), a novel algorithm taking a step further from Cluster & Tune by incorporating three crucial elements: feature extraction, appropriate clustering techniques, and soft pseudo-labels. We rigorously tested our method on both topical and non-topical datasets. Our findings demonstrated a significant improvement in accuracy, particularly in scenarios with a limited number of labeled instances, showcasing the efficacy of our proposed methods in mitigating the cold start problem.

1 Introduction

In natural language processing, the availability and quality of labeled data are crucial factors that significantly influence the performance in the fine-tuning phase of text classification tasks. However, in real-world scenarios, there is a significant scarcity of labeled data. Gathering sufficient labeled examples is often hindered by issues like data confidentiality, annotation quality, and the large number of categories, making extensive labeling impractical in many real-world cases. Using pretrained LLMs to train on few labeled instances often results in poor performance in target task, a phenomenon called *cold start problem*.

Prior works have extensively explored the importance of pretraining large language models and the critical role of transfer learning in enhancing their effectiveness for downstream tasks (Ziser and Reichart, 2018; Shnarch et al., 2018). Additionally, employing a pretrained model for intermediate supervised tasks requiring high-level reasoning in various domains has been shown to be beneficial during the fine-tuning stage (Pruksachatkun et al., 2020). Multi-phase pretraining on domain-related and then task-related data can lead to performance gain (Gururangan et al., 2020). This is partly because task-domain data often differ significantly from the general corpora used in pre-training models like BERT. Intermediate training on domain-specific datasets may aid the model in learning contextualized representations and terminologies not present in general corpora (Whang et al., 2020).

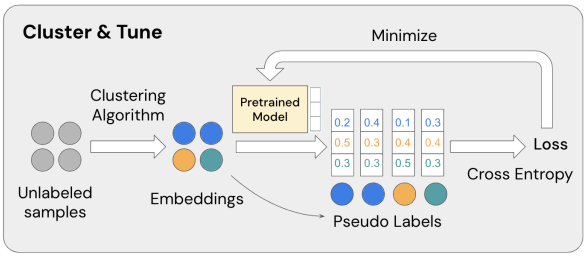

Figure 1: A pipeline of employing the intermediate unsupervised task described in Cluster&Tune paper.

Our baseline paper "Cluster and Tune"(Shnarch et al., 2022), builds on the notion of task-adaptive pretraining (Gururangan et al., 2020) in scenarios with few labeled instances. It employs unlabeled target samples during the intermediate training phase as shown in Figure 1. Utilizing sequential bottleneck clustering, the method clusters unlabeled samples to create pseudo labels for the pretrained model. The model is then trained to minimize the loss between its output and these pseudo labels using cross-entropy loss, a step crucial for enhancing model performance.

Our method builds upon this unsupervised intermediate task further through a series of preliminary

studies. These studies concentrate on selecting optimal clustering algorithms, developing more effective loss functions, and leveraging latent feature representations of the target-domain data. Based on these investigations, we proposed our novel method, Optimized-Embedded Cluster & Tune (OECT), by integrating three essential components: feature extraction, refined clustering methods, and the utilization of soft pseudo-labels. OECT significantly improves average accuracy across 5 topical and 3 non-topical datasets compared to our baseline paper. Our main contributions include:

- Undertaking three preliminary studies across three areas of interest—optimal clustering algorithms, effective loss functions, and latent feature representations—to enhance LLM performance during the cold start problem.
- Proposing OECT, a novel inter-training algorithm that combines three vital elements: feature extraction, enhanced clustering methods, and the application of soft pseudo-labels.
- Carrying out extensive experiments that demonstrate the superior performance of OECT in comparison to the vanilla Cluster & Tune across five topical and three non-topical datasets.

2 Preliminary Study

We have conducted preliminary studies based on our 3 main ideas, titled: *Cluster Optimization*, *Entropy Minimization*, and *Embedding Clustering*.

2.1 Cluster Optimization

Our first proposal involved enhancing the clustering algorithm. The original study predetermined the cluster count at 50 without incorporating any optimization measures. Hence, we suggested alternative clustering algorithms, including DBSCAN, Affinity Propagation, and Mean-Shift, which eliminate the need for predefining the number of clusters. We also use K-means from the baseline paper. Comprehensive explanations of each clustering algorithm are provided in Appendix C.

The outcomes of our first proposal are presented in Table 1. According to the table, DBSCAN emerges as the most proficient clustering algorithm. Specifically, on topical datasets, it demonstrates a notable average accuracy improvement, elevating from the baseline of 27.4% to 56.1%. Conversely, for non-topical datasets, it yields a modest increase from the baseline of 81% to 86.9%. OPTICS and the original algorithm rank as the second and third best performers, respectively. This illustrates that opting for a suitable clustering algorithm in inter-training possibly leads to a potential enhancement in the classification task.

Method	Topical	Non-Topical
Original Paper		
BERT (baseline)	27.4	81.0
BERT + SIB	54.4	85.7
Ours		
BERT + Mean-Shift	49.0	80.3
BERT + AP	52.1	84.0
BERT + K-means	53.9	85.2
BERT + OPTICS	55.2	84.7
BERT + DBSCAN	**56.1**	**86.9**

Table 1: Average classification accuracy from the first preliminary study "Clustering Optimization" on 5 topical datasets and 3 non-topical datasets. The results are averaged on 3 different random seeds.

2.2 Entropy Minimization

Our second exploration centered on introducing the entropy minimization approach. The intention was to transition from the current clustering algorithm to an alternative algorithm specifically designed to minimize entropy loss as shown in Figure 2. Our objective was to establish a new intermediate unsupervised classification task, where the emphasis was on maximizing the model's prediction confidence for unlabeled data. The comprehensive explanation of entropy loss is available in Appendix B.

The experimental results obtained from this approach reveal some advancements compared to BERT is shown in Table 2, with an average accuracy of 38.1% on topical datasets and 82.4% on non-topical datasets. Nevertheless, it is important to note that this method falls short of producing an improvement when compared to the outcomes reported in the original paper.

The poor performance of entropy minimization can be due to the technique's inclination to prompt the model to make predictions with elevated confidence. This is evident in Figure 3 illustrating the predicted classes of unlabeled training data from Yahoo! Answering dataset, where approximately at the 10,000$^{\text{th}}$ sample, the model exhibits excessively high confidence for certain classes, leading to a stop in the prediction of both class 7 and class 4.

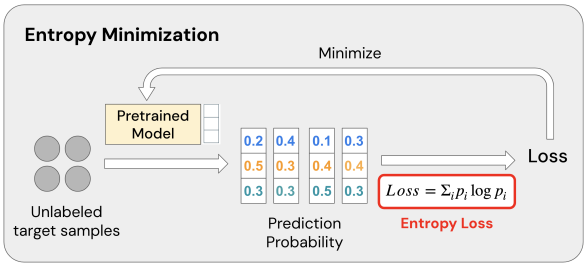

Figure 2: A pipeline of Entropy Minimization approach

Method	Topical	Non-Topical	Run-time (s)
Original Paper			
BERT (baseline)	27.4	81.0	**598**
BERT + C&T	**54.4**	**85.7**	751
Ours			
BERT + EntMin	38.1	82.4	**598**

Table 2: Average classification accuracy from the second preliminary study "Entropy Minimization" on 5 topical datasets and 3 non-topical datasets. The results are averaged on 3 different random seeds.

2.3 Embedding Clustering

Our third approach focused on enhancing the feature extraction process. In contrast to the original methodology, which solely relies on a bag of words as the data representation, we introduced a new pipeline. In this revised process as shown in Figure 4, a pre-trained model is initially employed to extract features from raw data before being input into the clustering algorithm. Furthermore, we introduced a new loss function, defined as the ratio of intra-distance to inter-distance within each cluster. This modification was implemented to guide the model in training such that the extracted features become more distinguishable in the embedding space. The comprehensive explanation of the ratio loss is available in Appendix B.

As a result shown in Table 3, this approach has resulted in a substantial enhancement compared to our previous techniques and has surpassed the performance of the original paper in terms of accuracy. Specifically, the accuracy has risen to 59.0% and 87.7% on topical and non-topical datasets, respectively. However, a notable drawback is the increased runtime, which is now five times longer than that of the original, approximately extending from 10 to 50 minutes.

3 Methodology

In addition to the three earlier contributions outlined in the preliminary study section, we introduce

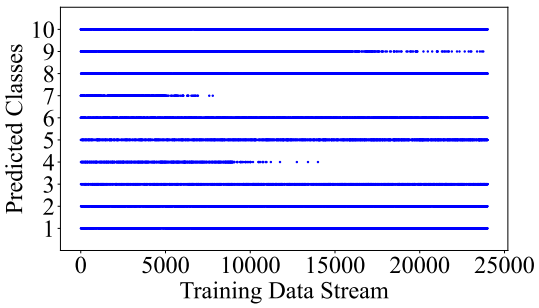

Figure 3: Model Collapsation during intermediate-training on Yahoo! Answering using Entropy loss. The trained model starts to predict only a few classes even if the data sequence is Independent and Identically Distributed (I.I.D.)

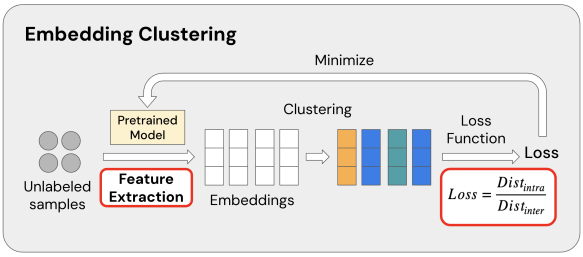

Figure 4: A pipeline of Embedding Clustering approach

a comprehensive approach that integrates key findings from the preliminary study and incorporates additional ideas to represent our final methodology.

Our final proposal, named Optimized-Embedded Cluster & Tune (OECT), consists of three primary components, namely *Feature Extraction*, *Alternative Clustering Algorithm*, and *Soft Pseudo-Labels*, as shown in Figure 5.

Feature Extraction. In line with the methodology of the third preliminary study, an initially utilized pre-trained model extracts features from raw data before feeding them into the clustering algorithm. This enhances the clustering algorithm's performance by providing a more meaningful representation of extracted features in a lower dimension.

Method	Topical	Non-Topical	Run-time (s)
Original Paper			
BERT (baseline)	27.4	81.0	**598**
BERT + C&T	54.4	85.7	751
Ours			
BERT + EmbClust	**59.0**	**87.7**	2963

Table 3: Average classification accuracy from the third preliminary study "Embedding Clustering" on 5 topical datasets and 3 non-topical datasets. The results are averaged on 3 different random seeds.

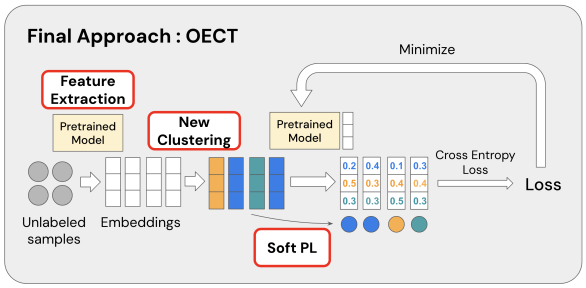

Figure 5: A pipeline of our final approach, OECT

Dataset	Train	Test	#classes
Topical			
Yahoo! answers	15K	3K	10
DBpedia	15K	3K	14
20 newsgroups	10.2K	7.5K	20
AG's news	15K	3K	4
ISEAR	5.4K	1.5K	7
Non-Topical			
SMS spam	3.9K	1.1K	2
Subjectivity	7K	2K	2
Polarity	7.5K	2.1K	2

Table 4: Dataset details

Alternative Clustering. Knowing that an appropriate clustering algorithm used in inter-training could potentially improve the classification task, we explored several alternative clustering methods as we did in the first preliminary. These methods encompassed DBSCAN, Affinity Propagation, Mean-Shift, and K-means. Comprehensive explanations of each clustering algorithm are provided in Appendix C.

Soft Pseudo-Labels. We introduces soft pseudo-labels derived from the clustering algorithm as our third component. This strategic choice is made from our expectation that soft pseudo-labels will offer a richer information source compared to the original hard pseudo-labels. A comprehensive explanation of soft pseudo-labels can be found in Appendix D.

In contrast to involving entropy loss, we have decided to retain the original cross-entropy loss as our primary loss function, as our second preliminary study suggests that entropy loss leads to inferior overall performance.

4 Data & Experiments

4.1 Datasets

We conducted evaluations across a total of eight datasets the same as the original paper, as described in the table Table 4. They are categorized into five topical datasets and three non-topical datasets, covering diverse classification tasks and domains. The topical datasets, characterized by a high-level distinction related to their content, include Yahoo! Answers (Zhang et al., 2016), which separates answers and questions to types; DBpedia (Zhang et al., 2016); CC-BY-SA which differentiates entity types by their Wikipedia articles; AG's News (Zhang et al., 2016); CC-BY-SA which categorize news articles; 20 newsgroups (Lang, 1995), which classifies 20 Usenet discussion groups; and ISEAR (Shao et al., 2015); CC BYNC-SA 3.0, which considers personal reports for emotion. Conversely, the non-topical datasets, which focus on the way sentences are written, include SMS spam (Almeida et al., 2011), which identifies spam messages; Polarity (Pang and Lee, 2005), which includes sentiment analysis on movie reviews, and Subjectivity (Pang and Lee, 2004), which categorizes movie snippets as subjective or objective. The sources of datasets directly used in this project are listed in Appendix A.

To mitigate computational costs, particularly for the larger datasets (DBpedia, AG's News, and Yahoo! Answers), we further reduced the size of the train/test sets to 15,000/3,000 instances, respectively. This reduction was achieved through random sampling from each set, and all runs and methods utilized the trimmed versions. Moreover, we implemented a 70%-10%-20% train-valid-test split for our datasets.

4.2 Experiment setting

In our primary experiments, we evaluate the performance of fine-tuning BERT-based models for a target task across various intermediate training configurations. We establish two baselines: (i) BERT without intermediate training and (ii) BERT with Cluster & Tune (BERT + C&T).

Fine-tuning samples : In every configuration, the fine-tuning for the target task is executed on each dataset, utilizing a fine-tuning budget of 64 labeled examples. The experiment is iterated three times, employing distinct random seeds for each repetition.

Inter-training : Upon conducting intermediate training, it was executed using the unlabeled training set for each dataset. We configured it to 50 for siB and K-means clustering as outlined in the original paper.

BERT hyper-parameters : All settings commenced from the BERT base model (110M param-

Method	Original	Reproduced (Ours)
Topical Datasets		
BERT	29.3	28.4
BERT + C&T	54.0	54.5
Non-Topical Datasets		
BERT	82.6	81.0
BERT + C&T	85.4	85.7

Table 5: Average classification accuracy from paper replication on 5 topical datasets and 3 non-topical datasets. The results are averaged on 3 different random seeds.

Method	Topical	Non-Topical	Run-time (s)
Original Paper			
BERT (baseline)	27.4	81.0	**598**
BERT + C&T	54.4	85.7	751
Ours			
BERT + OECT	**58.9**	**88.3**	743

Table 6: Average classification accuracy from the final approach "OECT" on 5 topical datasets and 3 non-topical datasets. The results are averaged on 3 different random seeds.

eters). Both BERT inter-training and fine-tuning were executed using the Adam optimizer (Kingma and Ba, 2014), adhering to standard configurations: a learning rate of 3×10^5, a batch size of 64, and a maximal sequence length of 128. Fine-tuning was arbitrarily set to span 10 epochs, mirroring the original study's approach.

For inter-training based on clustering outcomes, a single epoch was employed for two reasons. First, prolonged training over the clusters might excessively redirect the model's focus towards learning the cluster divisions, which in our context serves as an auxiliary task, not the primary target. Second, from a practical standpoint, single-epoch training is preferred due to its minimal run-time demands.

5 Results

5.1 Replication Results

By following the exact procedure outlined in the original paper, we observed consistent improvements when applying the cluster&tune approach to the BERT model across both types of datasets, as shown in Table 5. This indicates the successful replication of our paper's findings.

5.2 Final Improved Approach

The application of our final proposal, OECT, which involves feature extraction, the utilization of a new clustering algorithm, and the incorporation of soft pseudo labels, resulted in a significant improvement from the baseline and was achieved within a comparable computational time. The result of our final improved approach is illustrated in Table 6.

6 Discussion

Improvements. As shown in Table 6, our ultimate approach proves to be both efficient and effective, demonstrating a notably higher accuracy compared to the original paper. Specifically, we observed an improvement from 54.4% to 58.9% in topical datasets and from 85.7% to 88.3% in non-topical datasets under conditions of limited labeled data. Importantly, this enhanced performance was achieved while maintaining a similar average run-time. The results from the previous section along with additional studies offer three intriguing points for discussion.

Computational Efficiency. Firstly, the comparison between Table 3 and Table 6 highlights that our final solution effectively addresses the issue of exploding computational time, achieving nearly five times faster processing in our third preliminary study while maintaining a comparable level of accuracy. This improvement can be attributed to the position of the clustering algorithm in the pipeline. In the third preliminary study, the clustering algorithm is embedded within each gradient update process, necessitating computation in every epoch and consuming a substantial amount of time. In contrast, our final proposal relocates the clustering algorithm outside the gradient update process, allowing it to be independently calculated once at the beginning. This adjustment results in significantly reduced computational time.

Effectiveness of feature extraction with K-means. Secondly, we have conducted an ablation study to highlight the incremental gains achieved with different clustering algorithms. Interestingly, unlike our preliminary study 1 where the feature extraction process was applied initially, we observed that clustering the extracted features using OPTICS and DBSCAN did not yield favorable results. Instead, K-Means emerged as the most effective clustering algorithm, as evidenced by the results presented in Table 7. The potential reason behind this finding lies in the feature extraction process, which turns high-dimensional input into a lower-dimensional representation. This mitigates the curse-of-dimensionality problem associated with K-Means and boosts the quality of clus-

Method	Topical	Non-Topical	Run-time (s)
Original Paper			
BERT (baseline)	27.4	81.0	598
BERT + C&T	54.4	85.7	751
Ours			
BERT + FeatEx + OPTICS	55.1	86.0	764
BERT + FeatEx + DBSCAN	55.4	85.9	742
BERT + FeatEx + SIB	57.6	87.1	757
BERT + FeatEx + K-Means	**58.8**	**88.0**	747

Table 7: Average classification accuracy from the ablation study "Effectiveness of feature extraction with K-means" on 5 topical datasets and 3 non-topical datasets. The results are averaged on 3 different random seeds.

Method	Topical	Non-Topical	Run-time (s)
Original Paper			
BERT (baseline)	27.4	81.0	598
BERT + C&T	54.4	85.7	751
Ours (Each Component)			
BERT + K-means	53.9	85.2	731
BERT + Soft PL	54.5	86.1	740
BERT + FeatEx	57.6	87.1	756
Ours			
BERT + OECT	**58.8**	**88.0**	743

Table 8: Average classification accuracy from the ablation study "Contribution of each component" on 5 topical datasets and 3 non-topical datasets. The results are averaged on 3 different random seeds.

ters. To ensure comprehensive coverage, a detailed discussion of the curse-of-dimensionality problem in K-Means is included in Appendix D.

Contribution of each component. In our final ablation study, we aim to analyze the incremental gains associated with each component of our ultimate solution, as depicted in Table 8. Notably, K-means clustering exhibited a dependence on other components and failed to achieve improvement when introduced alone. On the contrary, components such as soft pseudo-labeling and feature extraction demonstrated the potential to yield improvements even when introduced independently in the pipeline. This observation leads us to infer that feature extraction serves as the most important component. The rationale behind this is that the original pre-trained model already encompasses well-designed feature extraction layers. Using these layers to extract features from raw input gives us much more information. Consequently, clustering on these embedded features is effective, resulting in higher-quality clusters and contributing to improved final accuracy.

6.1 Limitations and Future Plan

Despite the progress made in our project, certain limitations persist. As highlighted in the preceding section, our methodology heavily relies on feature extraction. The dependency on the feature extraction layer needs to be further investigated. To illustrate, experimenting with various pre-trained networks would provide insights into the algorithm's robustness if the feature extraction layer performs poorly.

Additionally, extending our observations beyond the BERT language model and diversifying the tasks beyond classification could contribute valuable findings. We can also investigate an impact on other languages such as Korean and Thai.

7 Conclusion

In summary, our work addresses the cold start problem prevalent in various language models by introducing an intermediate unsupervised task between the pre-training and fine-tuning phases in a pipeline. Our original paper presented a clustering and pseudo-labels algorithm for this intermediate task, and our successful replication of those results served as a foundation. Building upon this, our proposed solution involved three key components: an alternative clustering algorithm, soft pseudo labels, and an additional feature extraction process. As a result, our method demonstrated performance improvements across both types of datasets in scenarios where labeled data is scarce. Notably, these enhancements were achieved within a comparable average run-time.

References

Tiago A Almeida, José María G Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 259–262.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work?

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Bo Shao, Lorna Doucet, and David R Caruso. 2015. Universality versus cultural specificity of three emotion domains: Some evidence based on the cascading model of emotional intelligence. *Journal of Cross-Cultural Psychology*, 46(2):229–251.

Eyal Shnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it blend? blending weak and strong labeled data in a neural network for argumentation mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–605, Melbourne, Australia. Association for Computational Linguistics.

Eyal Shnarch, Ariel Gera, Alon Halfon, Lena Dankin, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2022. Cluster tune: Boost cold start performance in text classification.

Taesun Whang, Dongyub Lee, Chanhee Lee, Kisu Yang, Dongsuk Oh, and HeuiSeok Lim. 2020. An effective domain adaptive post-training method for bert in response selection.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level convolutional networks for text classification.

Yftah Ziser and Roi Reichart. 2018. Pivot based language modeling for improved neural domain adaptation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1241–1251, New Orleans, Louisiana. Association for Computational Linguistics.

A Datasets

These are some links for downloading the datasets. We used the same links provided by the original paper.

Polarity :
http://www.cs.cornell.edu/people/pabo/movie-review-data/.

Subjectivity :
http://www.cs.cornell.edu/people/pabo/movie-review-data/.

CFPB :
https://www.consumerfinance.gov/data-research/consumer-complaints/.

20 newsgroups :
http://qwone.com/~jason/20Newsgroups/, https://scikit-learn.org/0.15/datasets/twenty_newsgroups.html.

AG's News, DBpedia, and Yahoo! answers :
https://pathmind.com/wiki/open-datasets.

SMS spam :
http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/

ISEAR :
https://www.unige.ch/cisa/research/materials-and-online-research/research-material/.

B Definitions of Loss Functions

Entropy Loss The expression Entropy Loss $= -\sum_{i=1}^{N} p_i \log(p_i)$ defines the concept of entropy loss, a fundamental measure used to quantify uncertainty within a probability distribution. This equation assesses the level of unpredictability or disorder associated with potential outcomes within a system or model. Here, Entropy Loss represents the degree of uncertainty, where $-\sum_{i=1}^{N}$ iterates over N distinct events or outcomes in the probability distribution. Each p_i denotes the probability of a specific outcome, ranging from $i = 1$ to N. The logarithm function $\log(p_i)$ is applied to each individual probability p_i, emphasizing the diminishing impact of higher probabilities on the entropy loss calculation. This mathematical formulation penalizes the model more for uncertain or less probable outcomes, contributing to higher entropy loss when predictions lack confidence. In summary, the entropy loss equation offers a quantitative means to evaluate uncertainty within probability distributions, providing crucial insights into prediction uncertainty and information content across various fields, including machine learning and information theory.

Clustering Ratio Loss The ratio Loss $= \frac{\text{Inter-Distance}}{\text{Intra-Distance}}$ serves as a crucial metric in clustering or classification tasks, quantifying the separation and distinctiveness of clusters or classes within a feature space. The inter-distance signifies the distance between different clusters or classes, often computed as the distance between cluster centroids or class means. Conversely, the intra-distance represents the distance within each cluster or class, typically calculated as the average distance between data points within the same cluster or class. A smaller loss value denotes well-separated clusters or classes, indicating larger distances between clusters (inter-distance) compared to the distances within clusters (intra-distance). Conversely, a higher loss value implies that clusters or classes are closer together or less distinct. Therefore, this ratio-based measure provides valuable insights into the effectiveness of clustering or classification algorithms by evaluating the separation and distinctiveness of different clusters or classes within the feature space.

C Clustering Algorithms

The Sequential Information Bottleneck (sIB) is a clustering algorithm rooted in the Information Bottleneck (IB) principle, primarily used for unsupervised learning and data compression tasks. sIB aims to extract relevant information from sequential data while discarding redundant or less important details. Built upon the Information Bottleneck concept, it optimizes a trade-off between compression and retention of relevant information. Tailored for sequential data, such as time-series or sequential observations, sIB iteratively processes data to uncover meaningful clusters. Through information-theoretic principles, it refines clusters to capture significant information while minimizing redundancy, using a trade-off parameter to control the granularity of formed clusters. Its iterative approach continually adjusts cluster boundaries to encapsulate information-rich segments, converging when further iterations offer minimal improvements. Applied in natural language processing, bioinformatics, signal processing, and sequential data analysis, sIB distills essential information from sequences, providing a structured method to extract meaningful patterns and clusters.

K-means clustering stands as a widely-used unsupervised machine learning algorithm designed to partition a dataset into K distinct, non-overlapping clusters. Employed across diverse domains like data mining, pattern recognition, and image segmentation, the algorithm operates through iterative assignment and refinement of cluster centroids until convergence. It initiates by randomly selecting K cluster centroids from the dataset, signifying the cluster centers. Data points are then allocated to the nearest centroid based on a distance metric, commonly employing Euclidean distance, forming initial clusters. Subsequently, centroids are recalculated as the mean of all data points within each cluster, becoming new cluster centers. This iterative process continues, reassigning data points and updating centroids until a stopping criterion, often a set number of iterations or minimal centroid change, is met. K-means aims to minimize within-cluster variance, optimizing the sum of squared distances between data points and their respective centroids. Crucial to its performance is the choice of K, often determined through methods like the elbow method or silhouette analysis. While efficient and scalable for large datasets, K-means can be sensitive to initial centroid placements and assumes clusters of similar sizes and spherical shapes. Variants such as K-means++ and mini-batch K-means address these limitations, enhancing its applicability across various contexts. Overall, K-means clustering offers a simple yet robust approach for clustering data based on similarity.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm widely used for identifying clusters in datasets without the need for a predetermined number of clusters. It excels in discovering clusters of arbitrary shapes and effectively handling outliers (noise). The algorithm relies on two primary parameters: Epsilon (ϵ) and MinPts. Epsilon defines the radius around a point, establishing its neighborhood, while MinPts sets the minimum number of points required within the ϵ radius for a point to be considered a core point. DBSCAN categorizes points as core, border, or noise points: core points have at least MinPts neighbors within ϵ, border points are within ϵ of a core point but lack sufficient neighbors, and noise points are outliers. Its workflow involves selecting a point, expanding its cluster by adding reachable

points within ϵ, and repeating this process until all points are processed. While effective in identifying clusters of various shapes and handling noisy data by categorizing outliers, DBSCAN's performance relies heavily on proper parameter settings, making it sensitive to ϵ and MinPts choices. It might encounter challenges with clusters of differing densities or irregular shapes and can be impacted by dataset dimensionality. Nevertheless, DBSCAN finds applications in spatial data analysis, image processing, and anomaly detection, particularly in scenarios where clusters lack clear separation or have complex structures. Careful parameter tuning and consideration of the dataset's characteristics are crucial for optimal clustering results with DBSCAN.

Affinity Propagation (AP) is a clustering algorithm that discovers clusters by simulating message passing among data points to identify exemplars (representative points) and their associated clusters autonomously, without requiring the prior specification of the number of clusters. It operates through the iterative updating of Responsibility and Availability matrices. The Responsibility Matrix (R) measures the suitability of one point to be an exemplar for another, while the Availability Matrix (A) evaluates the evidence for a point to choose another as its exemplar. The algorithm initializes these matrices and iteratively updates them based on accumulated messages exchanged between data points. This message-passing scheme continues until a stable set of exemplars emerges. Exemplars and clusters are then identified by selecting points with high values in both the Responsibility and Availability matrices. This method allows Affinity Propagation to handle varying cluster sizes and automatically infer the number of clusters based on internal data interactions. It finds applications in diverse fields like bioinformatics and social network analysis, effectively identifying representative data points and forming clusters through message passing and similarity assessment. However, its computational demands increase with larger datasets, and sensitivity to initial conditions may affect clustering outcomes, necessitating careful parameter tuning and resource considerations for optimal results, particularly in more extensive datasets.

Mean-Shift is a clustering algorithm that identifies clusters by detecting density peaks or modes in the probability density function of the data. It begins by estimating the kernel density function of the dataset, treating each data point as a potential cluster center. The algorithm iteratively shifts these points towards the mode of the density function until convergence, adjusting each point's position based on the weighted average of its neighbors within a specified bandwidth. Points that converge to the same mode belong to the same cluster. Unlike K-means, Mean-Shift does not require the number of clusters to be predetermined, and it can identify clusters of various shapes and sizes. However, its performance is highly sensitive to the bandwidth parameter used in kernel density estimation; a small bandwidth may result in too many fine-grained clusters, while a large bandwidth may merge distinct clusters. Additionally, it can be computationally demanding, particularly with larger datasets, as it iterates until convergence for each point, making it less efficient for high-dimensional or extensive datasets. Nonetheless, Mean-Shift is effective in scenarios where the underlying structure of the data is unknown or where clusters have non-linear shapes, finding applications in image segmentation, object tracking, and other domains requiring unsupervised clustering of data. Careful selection of the bandwidth parameter is essential for achieving optimal clustering results with Mean-Shift.

D Additional Terms

Soft pseudo-labels generated by clustering algorithms involve assigning probabilistic or soft labels to data points, indicating the likelihood or probability of a point belonging to different clusters rather than assigning a hard, definitive label. This process typically follows clustering algorithms like K-means, DBSCAN, or hierarchical clustering, where data points are grouped into clusters based on specific criteria. Soft labels are derived by computing the probabilities or degrees of membership of data points to various clusters, representing their confidence or certainty in association with multiple clusters. These labels offer a probabilistic representation, indicating a point's likelihood of belonging to each cluster, allowing for nuanced interpretations of

uncertainty or potential secondary memberships. Soft pseudo-labels find applications in scenarios like semi-supervised learning, providing additional information for unlabeled data, and in transfer learning, aiding adaptation to new domains. While flexible and robust in handling complex data distributions, their effective utilization requires consideration of clustering parameters and models capable of handling probabilistic or soft targets during training.

Curse of dimensionality in K-means poses challenges in high-dimensional spaces, significantly impacting the performance of K-means clustering. In such spaces, characterized by a large number of features or dimensions, data becomes sparser, and the notion of proximity becomes less reliable as points are equally distributed, making distances less discriminative. This increased sparsity and equal spreading of distances make it challenging to define meaningful clusters accurately. Moreover, the computational complexity of K-means escalates in higher dimensions, becoming computationally expensive and sometimes impractical. The need for exponentially more data as dimensions increase further exacerbates the challenge, making data collection challenging. The curse of dimensionality also introduces risks of overfitting, where the model may capture noise instead of meaningful patterns, impacting generalization to new data. To address these issues, dimensionality reduction techniques like Principal Component Analysis (PCA) or feature selection, along with the use of appropriate distance metrics or clustering algorithms robust to high dimensions, such as density-based methods, are often employed in K-means clustering.