Automatic Text Labeling Method Based on Large Language Models

Chenwu Li [1,2] and Henry Dyke A. Balmeo [1]
[1]Graduate School, University of the East, Manila, Philippines
[2]GuangDong Polytechnic, Guangdong,China
henrydyke@gmail.com,lichenwu05@gmail.com

***Abstract*—With the increasing demand for large amounts of training data for model development, this paper proposes LLM4Label, an automatic text labeling method based on large language models, to assist human labelers in annotating text data. LLM4Label first selects the most representative seed data using a clustering algorithm based on text similarity. It then constructs prompt dialogues with few-shot prompts to stimulate the language model's performance on entity labeling tasks, enabling it to automatically and efficiently label more data. Finally, LLM4Label introduces human feedback to correct uncertain labeling results and retrains the model with the corrected annotations. Experiments show that LLM4Label achieves high-quality labeled data at low human labeling cost. The proposed method provides an effective way to obtain sizable and high-quality annotated datasets with minimal manual effort, which can strongly support downstream natural language processing tasks.**

***Index Terms*—data automatic annotation, large model, prompt engineering, text similarity clustering algorithm**

I. Introduction

In various fields, there is a vast amount of unprocessed natural language text, which contains knowledge of immense value for aspects such as intelligence analysis and strategic planning deployment. However, training a high-performing domain-specific model traditionally requires the collection, cleansing, and labeling of a substantial amount of data, with the initial data annotation and preparation phase consuming significant time and human resources. To enhance the efficiency of the data preparation stage, it is necessary to explore suitable methods for the automatic annotation of text. Nevertheless, due to the complex challenges posed by the specialized knowledge requirements and the complexity of label categories among other issues, the task of automating natural language text annotation presents intricate challenges. Therefore, investigating appropriate methods for automatic data annotation, with the aim of acquiring more high-quality labeled data for model training with minimal annotation costs, holds substantial research value.

II. Research Background

A. Automatic Data Annotation

In the field of information processing, challenges such as the scarcity of available corpus data, the rarity of expert knowledge, and the difficulties associated with annotating sample data are prevalent [1], [2]. Successful machine learning models are often predicated on the foundation of extensive, high-quality training data [3]. However, the growing shortage of labeled, high-quality textual data is increasingly becoming one of the key issues limiting the improvement of model performance [4], [5]. To address this challenge, methods of automatic data labeling based on autonomous learning have been proposed [6], [7], aiming to make the labeling process more efficient. The common procedure for automatic data labeling involves initially selecting a small, representative subset of data from the original domain data through a data selection model (classification model) for manual annotation, which serves to initialize a labeling model. Subsequently, this model is applied to the unlabeled original dataset. If the labeling model determines the confidence of the annotation result to be high, the label is retained; otherwise, the low-confidence annotation results are presented to annotators for a second judgment. This subset of data is then re-entered into the labeling model to enhance its annotation efficacy. Through a number of iterations, a substantial and reliable labeled dataset is ultimately obtained.

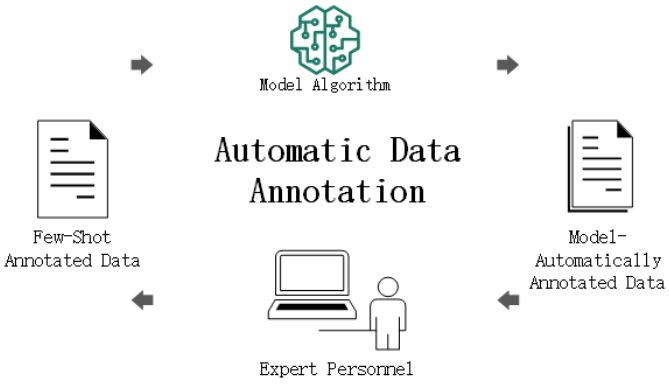

Fig. 1. Data automatic labeling flow chart

Through the aforementioned automated annotation process, annotators are only required to manually label data during the initialization of the labeling model and the correction of the model's outputs. While this significantly reduces the quantity of data that annotators need to label, the process involves updating the model, leading to a lengthy waiting period for the next set of low-confidence results [7], [8]. There is a need to devise reasonable methods to assist the data annotation iteration process in an automated manner, thereby further enhancing the efficiency of automatic data labeling methods.

B. Large Language Models and Prompt Engineering

Distinct from traditional pre-trained models, Large Language Models (LLMs) are characterized by their significantly larger training datasets, parameter counts, and computational requirements. These models exhibit emergent properties and possess profound contextual understanding capabilities [9], [10]. Due to their robust natural language understanding abilities, LLMs, leveraging open-source large models, have facilitated a rich variety of downstream task scenarios. For instance, ChatGPT [11]has demonstrated remarkable conversational capabilities. While large models offer numerous advantages, their extensive parameter size and computational demands significantly increase the costs associated with storing and deploying fully fine-tuned models for each downstream task. Consequently, Prompt Engineering has emerged [12] as a relatively new discipline aimed at developing a minimal set of prompts to optimize large model outputs. This approach seeks to efficiently employ LLMs for applications and research topics that are either more complex or domain-specific [13], [14].

Prompt Engineering encompasses methods such as few-shot prompting, chain-of-thought prompting, and active prompting. Few-shot prompting [15], based on contextual learning, guides the model's generation by providing examples, allowing the model to learn how to perform tasks from a single example or necessitating additional prompts for more challenging tasks. Chain-of-Thought Prompting [16] enhances the model's reasoning capabilities by supplying examples of the thought process, combining this with few-shot prompts to enable the model to answer more complex questions through reasoning. As the fixed set of manually annotated examples provided to the model initially might not remain the most effective over time, Active Prompting [17] was proposed. This approach involves human participation in the chain of thought process, selecting the most uncertain answers through uncertainty calculation algorithms for human re-annotation, and basing further reasoning on these annotations.

III. LLM4Label Method

To assist in the automated text annotation process and obtain a larger quantity of high-quality labeled entity data with minimal annotation cost, this paper introduces LLM4Label, a text data automatic annotation method based on large models. Initially, a clustering algorithm based on similarity calculations selects the most representative samples in the dataset for manual annotation by annotators, serving as seed domain knowledge for the large model's learning. Subsequently, few-shot prompt engineering techniques are utilized to enhance the large model's performance in entity annotation tasks, enabling the model to automatically and efficiently label a larger volume of sample data. Finally, a human feedback mechanism is introduced to manually correct results of the large model with low confidence levels, with these corrected results fed back to the model for further learning. This method aims to obtain a greater quantity of high-quality labeled data with the least human effort and cost. The main problems addressed by this method are as follows: This paper focuses on two key problems:

1) How to reasonably select representative seed data for annotation?
2) How to leverage large language models for automatic data annotation?

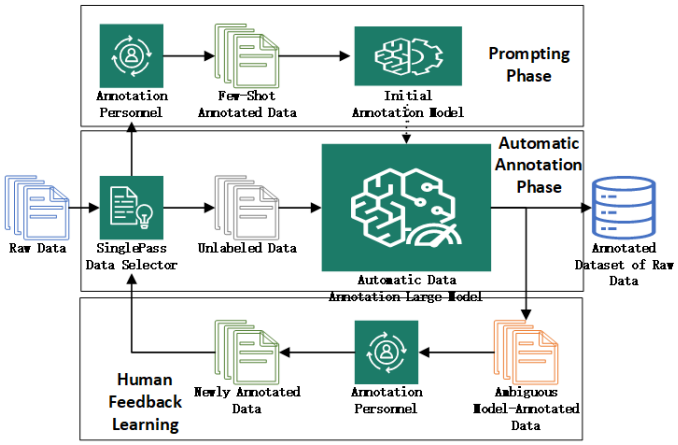

Fig. 2. LLM4Label pipeline

A. Data Selector

Given the vast reservoir of knowledge possessed by large models, the key to the method lies in how to select a small yet representative subset of sample data for annotation by human annotators.

The first step in the LLM4Label approach involves constructing a data selector. This process employs the SinglePass [23] clustering algorithm to explore the intrinsic characteristics of the data, selecting representative samples within each cluster for annotation. This method effectively utilizes the natural properties of the data to reduce the number of annotations required while choosing samples that are highly representative, thereby enhancing the large model's performance capabilities. The SinglePass (single iteration) clustering algorithm is an iterative clustering method that is memory-efficient, suitable for large datasets, and does not require a predetermined number of clusters. The algorithm starts at a starting point, iteratively traverses samples in the dataset, and assigns them to the nearest cluster based on the distance to the current cluster center. If the nearest cluster does not meet the preset conditions, SinglePass creates a new cluster with the sample as its center and continues the iterative traversal until the algorithm converges.

From each obtained cluster, a certain sampling ratio is used to select data to form the seed dataset for the large model to learn from. Annotators only need to focus on annotating this seed data to complete the data preparation process.

B. Automatic Data Annotation Based on Large Models

To inspire large models to complete automatic annotation tasks, it is necessary to introduce prompts within a limited set of example dialogues to guide the large model in generating data annotation results. LLM4Label, leveraging few-shot prompts, stimulates the large model's contextual learning

```
Algorithm 1: SinglePass Clustering Algorithm
Data: Annotated Dataset
Result: Clusters
Input: Dataset
Output: Clustering Results
clusters = []
cluster_centers = []
threshold = 0.5
foreach sample in Annotated Dataset do
    nearest_cluster = None
    min_distance = Infinity
    foreach cluster_center in cluster_centers do
        distance = calculate_distance(sample, cluster_center)
        if distance < min_distance then
            min_distance = distance nearest_cluster = cluster_center
    if min_distance < threshold then
        assign sample to nearest_cluster
        new_center = update_cluster_center(sample, nearest_cluster)
        cluster_centers[nearest_cluster] = new_center
    else
        create new cluster and set sample as its center
        new_center = [sample]
        clusters.append(new_center)
        cluster_centers.append(sample)
```

Fig. 3. SinglePass Clustering Algorithm

capabilities, achieving better performance in automatic data annotation tasks.

The construction method for few-shot prompts involves analyzing the seed data output by the data selector and the task objectives and label information in the manually annotated labels to form the prompt text in the dialogue prompts. The prompts, text from the seed data, and the manual labels of the seed data together constitute a set of prompt dialogues. These dialogues are then input into the large model for learning, completing the initialization process of the large model in the text data annotation task.

C. Human Feedback Learning for Data Annotation Results

In response to the annotation results output by the large model, a human feedback loop process is designed to re-annotate the uncertain annotation results of the large model and provide them again for the model to learn. Through this cycle, the uncertain outputs of the large model can be corrected, aligning the model's outputs more closely with the requirements of automatic entity annotation tasks. Human feedback enables more efficient use of limited annotation data to address domain knowledge, aiming to reduce the burden of annotation.

IV. Experiments

A. Datasets

This method collected publicly available information from open sources such as news, Weibo, public accounts, and journals, which after screening, formed the experimental dataset. Initially, the data underwent preprocessing to remove textual noise, such as URLs and special symbols. Subsequently, researchers annotated the dataset, identifying and labeling twelve categories of fields within it, resulting in the experimental annotated dataset, the details of which are as shown in Table I.

B. Baselines

To validate RQ1, LLM4Label compares its performance with that of currently proven, efficient, and fast classification algorithms like K-means, LDA, and DBSCAN through experimental trials. To assess the effectiveness of automatic data annotation implemented by large models, the entity extraction model Spert is chosen for comparison.

1) K-means: The K-means algorithm is a classic clustering algorithm [18], [19], aimed at dividing a dataset into a pre-specified number of clusters. The K-means algorithm requires a predetermined number of clusters, and the clustering process involves

TABLE I
Dataset Description

Category Name	Description	Quantity
Personnel	Individuals appearing in the text	248
Positions	Positions within texts, including government and research positions	147
Countries	Country names within the text	87
Institutions	Institutions, including government and research institutions, within the text	519
Bases	Base entities within the text	47
Maritime Areas	Maritime area entities within the text	34
Ports	Port entities within the text	60
Vessels	Naval gun-type entities within the text	441
Aircraft	Entities such as drones, helicopters, fighters, and bombers within the text	141
Trucks	Entities such as mini, medium, and large trucks within the text	84
Cars	Entities of various car brands within the text	93
Equipment	Various named equipment entities within the text	34

This ensures that data points within a cluster have a high degree of similarity, while the similarity between clusters is low. The advantage of the K-means algorithm lies in its simplicity and ease of understanding, making it effective for small to medium-sized datasets. The basic idea of the K-means algorithm is to divide data points into K clusters, with each cluster's center being the average of all data points within that cluster. The algorithm's implementation steps are: 1) Select the number of clusters k=3, 2) Initialize center points, 3) Assign remaining data points, 4) Update center points, and repeat steps 3 and 4 until the cluster centers no longer change significantly or a predetermined number of iterations is reached.

2) LDA: LDA is a widely used generative probabilistic model in the field of topic modeling [20], aiming to discover latent topic structures within text corpora to aid in understanding text associations and topic distributions. The core idea of LDA is to assume that each document follows a Dirichlet distribution over topics, and each topic's distribution over words also follows this distribution. Based on this, LDA links document-topic distributions and topic-word distributions together. Through Bayesian inference, the algorithm infers the topic distribution of documents and the word distribution of topics. The steps of the LDA algorithm include 1) Parameter

initialization, 2) Distribution initialization, 3) Sampling, 4) Parameter updates and repeat sampling updates, iterating until the model converges. In the experimental process of this method, each sample data is treated as a document input into the clustering algorithm for experimentation, with the clustering theme's empirical parameter set to 3.

3) DBSCAN: DBSCAN is a classic density-based clustering algorithm aimed at identifying data points with similar densities and dividing them into different clusters. Its distinctive feature is the ability to discover clusters of any shape and its robustness to noise data [21]. The working principle of DBSCAN is based on four key concepts: core objects, direct density reachability, density reachability, and density connectivity. Based on these concepts, DBSCAN's implementation steps include selecting any unvisited data point, determining core objects, expanding clusters, finding new core objects, and repeating these steps until all data points have been visited.

4) Spert: Spert is an attention model for joint entity and relation extraction based on spans [22], also conducting joint experiments on entity extraction and relation extraction tasks. Its notable contributions include: first, introducing a pretrained lightweight model, Bert, to represent text for subsequent inference; second, incorporating text span information as one of the classifiers' inference bases; third, employing a powerful negative sampling mechanism to process the dataset, making the Bert model more sensitive to text spans. In the experiments, all annotated data are divided into training and testing data in an 8:2 ratio, with training data used for model fine-tuning and testing data for evaluating the performance of the fine-tuned model. The model loads the Chinese pretrained Bert model, bert-base-chinese, for text representation, with the candidate phrase generation process conducted under a maximum span empirical parameter of 30 (ensuring generated candidate entities do not exceed 30 in length), training batch size set to 2, reading two training data at a time, and completing model fine-tuning after 10 rounds of learning. The method is trained and tested in a cuda environment, and under the same model parameter settings, experiments are repeated five times, with the average of the five experimental results taken as the final model performance metric.

C. Automatic Data Annotation Based on Large Models

After data selection, 13 seed data samples were obtained through SinglePass clustering model sampling. The selected annotated data is parsed, with text concatenated with prompts in the question, "Identify entities within the text and return the data in the specified format. Identified entity categories include: personnel, positions, countries, institutions, bases, maritime areas, ports, vessels, aircraft, ammunition, weapon systems, detection systems." As shown in Figure 4, a total of 26 prompt dialogues were constructed. These dialogues were input into the large model for learning. The question prompt template indicates the task for the large model (identify entities) and the types of entities to identify (12 categories). The answer template specifies the format for the large model's return results, including the entity phrase and the category of the determined entity. For unlabeled data, the large model learns the extraction target and patterns from the prompt template and returns the extraction results in the style of the prompt template answers. The constructed prompts and questions are input into the large model for prediction, with identification results having a confidence level above 0.6 considered as the large model's output.

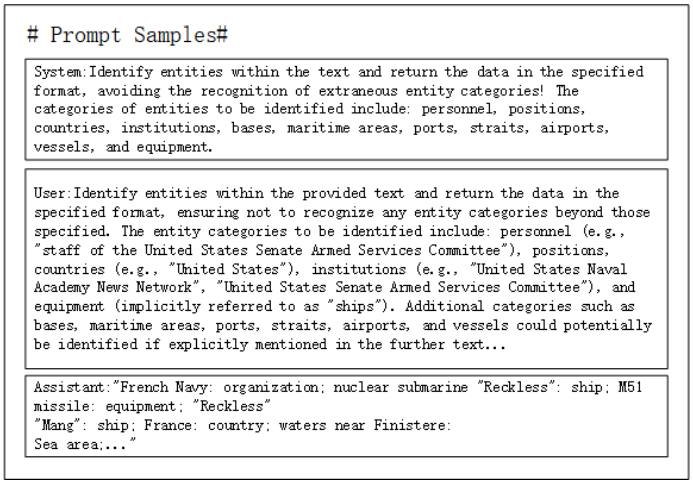

Fig. 4. Example of LLM Prompt

D. Evaluation Method for Annotation Results

The method uses Precision (P), Recall (R), and F1-Score values to measure experimental results. Precision measures the proportion of correctly identified results by the model out of all the model's output results, Recall measures the proportion of correctly identified labels by the model out of all actual entity labels, and F1-Score combines Precision and Recall, providing a comprehensive performance metric. These metrics comprehensively evaluate the model's performance on the dataset, helping to determine the quality of entities automatically identified by the large model.

After learning from the prompt templates, the method inputs questions without answers and obtains the large model's output. The large model's return data will follow the format of the answer template.

V. Results and Analysis

The experiments evaluated the performance of LLM4Label across twelve entity categories, as shown in Table II.

A. RQ1 Validation of Data Selector Effectiveness

To validate the role of the data selector in RQ1, the paper opted for experiments with three different selection algorithms.

Compared to K-means, LDA, and DBSCAN, LLM4Label achieved an F1-Score of 0.8324, outperforming the baseline methods by 30.28%, 46.82%, and 28.08% in terms of comprehensive performance metrics. The experiment demonstrates LLM4Label's higher performance and efficiency in clustering tasks on the dataset.

TABLE II
LABEL RESULTS

Category	P	R	F
Personnel	0.6176	0.6	0.6087
Positions	0.1765	0.375	0.24
Countries	0.7222	0.6842	0.7027
Institutions	0.6316	0.6857	0.6575
Bases	0.5	0.0714	0.125
Maritime Areas	0.125	0.0833	0.1
Ports	0.5	0.5	0.5
Vessels	0.5161	0.64	0.5714
Aircraft	1	0.9231	0.96
Trucks	1	1	1
Cars	0.3333	0.3333	0.3333
Equipment	0.3333	0.2	0.25
Sum	**0.8837**	**0.8354**	**0.8324**

TABLE III
EXPERIMENTS RESULTS

Data Selector	P	R	F1
Kmeans	0.5872	0.5146	0.5296
LDA	0.4633	0.3207	0.3642
DBSCAN	0.6242	0.5122	0.5516
LLM4Label	**0.8837**	**0.8354**	**0.8324**
Sum	**0.8837**	**0.8354**	**0.8324**

B. RQ2 Validation of Large Model Automatic Annotation Method

To compare the superiority of large models in entity recognition, the method selected traditional deep learning models for fine-tuning with fully annotated data and compared the results with those annotated by large models using few-shot prompts.

TABLE IV
EXPERIMENTS RESULTS

Method	P	R	F1
Spert	0.7385	0.7981	0.7668
LLM4Label	**0.8837**	**0.8354**	**0.8324**
Sum	**0.8837**	**0.8354**	**0.8324**

The experimental results demonstrate that large models, stimulated by few-shot templates, have achieved performance surpassing that of fine-tuned pre-trained models, showcasing the credible and stable capability of large models in entity recognition tasks. Simultaneously, it has been proven that with the robust support of large models, automatic data annotation tasks can be accomplished using a small number of annotated samples.

VI. CONCLUSION

LLM4Label achieves automated annotation of textual data through sample selection, few-shot prompt engineering techniques, and a human feedback mechanism. The method offers several advantages, including reducing annotation costs, enhancing annotation quality, increasing the volume of annotated data, and fully leveraging limited expert knowledge. The LLM4Label method presents an innovative approach by comprehensively utilizing large models, manual annotation, and feedback mechanisms to obtain more high-quality entity-annotated data with minimal human resource cost. This method is expected to provide robust support for information extraction and analysis in the field and holds potential significance for addressing the time-consuming and labor-intensive issues associated with data annotation work.

REFERENCES

[1] Gormley, M. R., & Mitchell, M. (2015). Temporal information extraction from narratives. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 1915-1920.

[2] Banko, M., & Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, 26(2), 26-33.

[3] Zhang, X., Zhao, J. and LeCun, Y., 2015. Character-level convolutional networks for text classification. Advances in neural information processing systems, 28.

[4] Ratinov, L., & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. Proceedings of the Thirteenth Conference on Computational Natural Language Learning, 147-155.

[5] Luo, L., Yang, J., and Zhang, J. (2015). Joint named entity recognition and disambiguation. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 879-888.

[6] Michael, J., 2006. Where's the evidence that active learning works?. Advances in physiology education.

[7] Prince, M., 2004. Does active learning work? A review of the research. Journal of Engineering Education, 93(3), pp.223-231.

[8] Cohn, D., Atlas, L. and Ladner, R., 1994. Improving generalization with active learning. Machine learning, 15, pp.201-221.

[9] Würsch, Maxime, Andrei Kucharavy, Dimitri Percia-David, and Alain Mermoud. L̇LM-Based Entity Extraction Is Not for Cybersecurity.̈(2023).

[10] Smith, Shaden; Patwary, Mostofa; Norick, Brandon; LeGresley, Patrick; Rajbhandari, Samyam; Casper, Jared; Liu, Zhun; Prabhumoye, Shrimai; Zerveas, George; Korthikanti, Vijay; Zhang, Elton; Child, Rewon; Aminabadi, Reza Yazdani; Bernauer, Julie; Song, Xia (2022-02-04). Üsing DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model. arXiv:2201.11990

[11] ChatGPT: Optimizing Language Models for Dialogue. OpenAI. 2022-11-30. Retrieved 2023-01-13.

[12] Diab, Mohamad; Herrera, Julian; Chernow, Bob (2022-10-28). Štable Diffusion Prompt Book̈(PDF). Retrieved 2023-08-07. Prompt engineering is the process of structuring words that can be interpreted and understood by a text-to-image model. Think of it as the language you need to speak in order to tell an AI model what to draw.

[13] Jiang, Dongfu, Xiang Ren, and Bill Yuchen Lin. "LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion." ArXiv preprint arXiv:2306.02561 (2023).

[14] Lee, Gibbeum, Volker Hartmann, Jongho Park, Dimitris Papailiopoulos, and Kangwook Lee. "Prompted LLMs as Chatbot Modules for Long Open-domain Conversation." arXiv preprint arXiv:2305.04533 (2023).

[15] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S., 2020. Language models are few-shot learners. Advances in neural information processing systems, 33, pp.1877-1901.

[16] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems. 2022 Dec 6;35:24824-37.

[17] Diao S, Wang P, Lin Y, Zhang T. Active prompting with chain-of-thought for large language models. arXiv preprint arXiv:2302.12246. 2023 Feb 23.

[18] Krishna, K., and M. Narasimha Murty. Ġenetic K-means algorithm.ÏEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29, no. 3 (1999): 433-439.

[19] Bock, Hans-Hermann. "Clustering methods: a history of k-means algorithms." Selected contributions in data analysis and classification (2007): 161-172.

[20] Yu, Hua, and Jie Yang. Ä direct LDA algorithm for high-dimensional data—with application to face recognition.Ṗattern recognition 34, no. 10 (2001): 2067-2070.

[21] Schubert E, Sander J, Ester M, Kriegel HP, Xu X. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Transactions on Database Systems (TODS). 2017 Jul 31;42(3):1-21.

[22] Jiang, Zi-Hang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. C̈onvbert: Improving bert with span-based dynamic convolution.Ädvances in Neural Information Processing Systems 33 (2020): 12837-12848.

[23] Bailey, Donald G., and Christopher T. Johnston. S̈ingle pass connected components analysis.Ïn Proceedings of image and vision computing New Zealand, pp. 282-287. 2007.