Cluster-Based Corrective Filtering for Class Specific Keyword Sets

Maria Nakhla

Abstract

The extraction of domain-specific keywords from textual data, a critical application within Natural Language Processing (NLP), has gained substantial importance in the contemporary data-driven landscape. The research concern is that there is a paramount chance of extracting keywords, which deviate from the core domain meaning. This is due to possibility of nth child keywords relations being introduced, which do not directly relate to the main domain goal. Thus, further keyword filtering is a crucial step to guarantee all keywords actually belong to the target domain. The methodology utilized consists of two main steps. The first one is clustering; in this phase multiple clustering techniques are investigated, and specially using a convex hull approach. Then comes the second step to get rid of outliers.Various techniques have been tested such as Isolation Forest and Local Outlier Factor. Text-Embeddings similarity measuring techniques with utilization of WordNet and ConceptNet are also involved as a final step. Furthermore, the utilized techniques are evaluated using recall, precision and F1-score, as well as with domain experts help for further evaluations. The results are quite promising using the convex hull clustering approach. The hybrid method combining three powerful tools which are clustering, outlier detection, and semantic similarity has proved its ability of getting rid of irrelevant class-specific keywords.

1 Introduction

1.1 Motivation & Previous Work

The CreateData4AI (CD4AI) [1] is one of the projects that motivated carrying out this research, as it represents an innovative solution which focuses on transforming unstructured text into structured, annotated datasets that are systematically classified according to specific features. The process encompasses several crucial steps, including **Keyword Extraction**. This involves the extraction of keywords and keyphrases, guided by classes or labels predefined by domain experts.

The main concern of this research is building a general tool to detect and remove outliers from a pool of domain-specific words, which would help CD4AI project by introducing a sub-step after Keyword Extraction to filter the extracted keywords.

1.2 Research Questions

This Research Topic focuses on investigating the following research questions:

1. Which clustering approaches currently exist that could be utilized to cluster keywords based on relevance to a class?
2. What are possible outlier detection methods that could also help to achieve a more class-specific keyword set?
3. Could different methods be combined for better results?
4. In which ways can the resulting filtered keywords set be evaluated?

   After answering these questions, cleaning and filtering the extracted keywords from the irrelevant generated ones would become possible and finally achieving outliers-free classes

2 Background

In this chapter, the main techniques which were proposed to solve the research problem will be discussed. These techniques includes exploring possible clustering methods, outlier detection methods, and last but not least the text-embedding models, which could to be utilized.

[1] CD4AI

2.1 Clustering Methods

Hierarchical Clustering is a method of cluster analysis that seeks to build a hierarchy of clusters. This approach is distinct in its use of an agglomerative algorithm, which starts with each data point as a single cluster and then successively merges clusters until one single cluster remains or a certain criterion is met. Ward's method has been specifically utilized. In the case of unlabeled data, both the methods complete method and ward method performed well, but Ward's method outperforms complete linkage method, specially in case where clusters are overlapping. Ward's method is particularly efficient in minimizing the variance within each cluster[1]. Thus, it is advantageous for identifying the precise number of clusters, which in this research case will vary from one class to another.

K-Means Clustering is a widely-used method for partitioning a dataset into K distinct, non-overlapping subgroups or clusters. In the research, k-means clustering was utilized for its computational efficiency and ease of interpretation. However, it assumes spherical clusters and it is sensitive to the initial placement of centroids[2], which can sometimes lead to suboptimal clustering solutions. As a result, it was a struggle for the specified use case, because it is very hard to point out the estimated number of clusters for each class.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) stands out from other clustering methods due to its ability to find arbitrarily shaped clusters and its robustness to outliers. It works based on two key parameters: ε (epsilon), a distance measure that defines the neighborhood around a data point, and *minPts*, the minimum number of points required to form a dense region. DBSCAN is particularly useful in the research for handling noise and identifying outliers. However, determining the appropriate values for ε and *minPts* per class was challenging, and the algorithm struggles with varying densities[3].

Latent Semantic Analysis (LSA) Though not traditionally viewed as a clustering method, Latent Semantic Analysis (LSA) can be employed for clustering in the context of text data[4]. However, it is not widely used for that purpose. In the research, LSA was tested to group related keywords within each class, revealing the hidden thematic structure in the text data.

2.2 Outlier Detection Methods

Isolation Forest (ISO) is an effective method for detecting outliers in high-dimensional datasets, such as text embeddings. Unlike distance-based methods, ISO works by isolating anomalies instead of profiling normal data points. It randomly selects a feature and a split value between the maximum and minimum values of the selected feature, recursively partitioning the dataset. This results in a forest of isolation trees. Outliers are expected to have shorter paths in these trees, as they are easier to isolate. The ISO method is computationally efficient and scalable, making it suitable for large text datasets. However, its performance can depend significantly on the choice of parameters and the nature of the dataset[5].

Local Outlier Factor (LOF) is particularly adept at identifying outliers in datasets with clusters of varying densities, which is a common characteristic in text embeddings. LOF works by measuring the local deviation of a given data point with respect to its neighbors. Points with a substantially lower density than their neighbors are considered outliers. This method is advantageous in datasets where the notion of an outlier is not globally defined but contextually relevant to local clusters[6]. The main limitation of LOF is its sensitivity to the choice of parameters, like the number of neighbors, which can significantly affect its detection accuracy.

Coresets offer a novel approach to outlier detection in large datasets, including text embeddings. A coreset is a small, representative subset of the original data that approximately preserves the properties of the entire dataset. By applying clustering or other analysis techniques to the coreset instead of the full dataset, outliers could be identified more efficiently. This method is particularly useful in reducing computational complexity and time, making it viable for large-scale text data. However, constructing an appropriate coreset that accurately represents the original data's distribution, including outliers, can be challenging[7].

Z-score is a statistical method, which is employed for outlier detection in class-keywords embeddings analysis. It involves measuring the number of standard deviations a data point is from the mean of the dataset[8]. Data points with a Z-score above a certain threshold are flagged as outliers.

This method is straightforward and effective for datasets with a distribution close to normal. However, its efficacy is reduced in datasets with skewed distributions or when the data has multiple dimensions with different scales which happens to be the current research problem.

These outlier detection methods have been instrumental in the analysis of text embeddings, each contributing uniquely to the identification of anomalies in the data. The choice of method depends on the specific characteristics of the dataset and the nature of the outliers, which are aimed to be detected.

2.3 Text Embedding Models

In the research, various state-of-the-art text-embedding models are employed to transform textual data into numerical embeddings. These embeddings are then utilized in clustering and outlier detection methods. The models shown in Table 1 have been pivotal.

Model	Details
jina-embeddings-v2-base-en	Advanced for semantic similarity, specialized in English
all-mpnet-base-v2	MPNet architecture, efficient in sentence context understanding, specialized in English
paraphrase-MiniLM-L12-v2	MiniLM optimized for multilingual paraphrase identification
gbert-large-paraphrase	BERT-based, fine-tuned for paraphrase identification with cosine similarity, specialized in German

Table 1: Text Embedding Models Comparison

3 Methodology

In this section, the outliers detection pipeline will be gone through step by step.

3.1 Data Translation

As a first step, each class has seed keywords briefly representing the class and general keywords in german, which are needed to be filtered, and remove irrelevant keywords from. Both german and english keywords have been tested to work with. However, some problems have occured, one of them is that german text embedding models are not as powerful as the ones trained on Engish. Thus, it was a crucial step to translate keywords first from German to English.

Several translators have been tried out, of which some of them had accurate translations but took a lot of time. There were others which had characters limit for free translations, so it was challenging to choose the best translator in the current use case. In Table 2, listed are the main experimented translators including name, and if there are characters limit for free translations. As a result of this step, the translated English keywords are obtained for all classes and then generate keywords embeddings.

3.2 Application of Clustering Methods

Currently, clustering is performed with the keyword embeddings, which are generated for each class. Majority of the experiments have been done with Hierarchical clustering methods, because it did not have the constraint of specifying the number of clusters, which was totally unknown and vary from one class to another. However, there was another challenging parameter that is the distance-threshold. The threshold controls the number of clusters so that as the threshold decreases, the number of clusters increases.

First, **the distance-threshold is set to be a constant**, but that was not the best solution as it would differ from one class to another. Then, another idea was to start with a very small threshold and keep increasing it in a slow rate until the number of clusters per class does not exceed a certain number. Moreover, recursive clustering was another approach to test. **Recursive clustering** is all about having an initial distance-threshold and generate the clusters, then check the density of each cluster if a cluster's density is less than a density-threshold, re-cluster that specific cluster with modification to

Translator	Limited
deep translator-google	No
deepL	Yes
translators 5.8.9-google	No
translators 5.8.9-lingvanex	No

Table 2: Different Translator Services

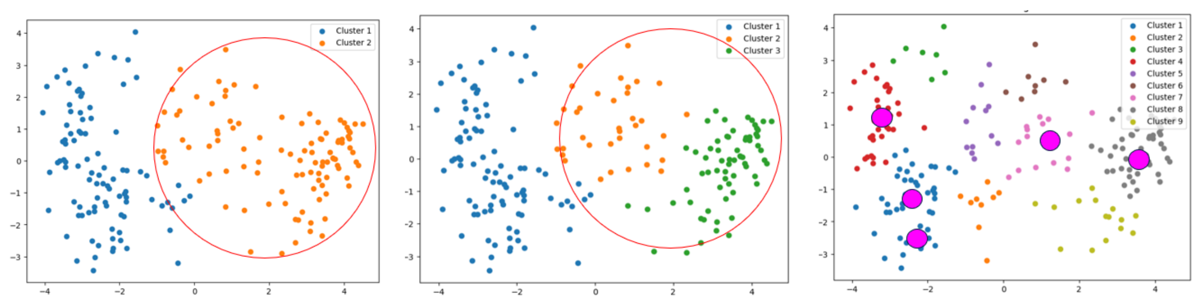

Figure 1: Recursive Clustering Illustration

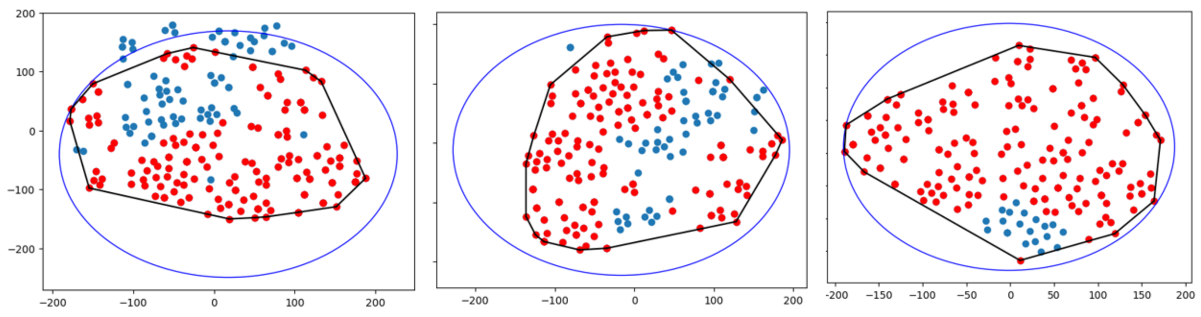

Figure 2: Convex Hull Approach with Recursive Clustering Illustration

the distance-threshold to be greater than the one used before. As a result, it is highly guaranteed to have well-defined clusters for each class. Figure 1 shows visualizations of recursive clustering technique.

Figure 1 shows how first clustering iteration step clusters the keywords. Then, it shows how after the orange cluster had a density less than the density-threshold was re-clustered again into two clusters. Moreover, it illustrates how the final result after all of the clustering iterations looks like. The purple dots represent the seed keywords existing in some of the clusters which in one of the approaches were considered the class-related keywords and the other clusters were considered as outliers.

The question of how outliers are defined after the clustering step is done will be discussed next. An initial idea was to consider a cluster as an outlier if its number of **members are less than a specific fixed number**. The number of members for each cluster are checked, if it is less than that specific number, then all of those keywords members are considered as outliers, and so on. Another idea was to also **remove furthest 25% members of each cluster**. Moreover, another approach was to **keep only the clusters which contain any of the class seed keywords as a member of it**. A comparison of mean embeddings of those clusters considered as outliers with the mean embeddings of the class seed keywords has been done, so that if they are within a margin, then do not consider that cluster as an outlier.

Furthermore, a solution using a **convex hull** or a **circle** was involved. This is achieved by enclosing all of the clusters which include a seed keyword with convex hull, so that some of the clusters which do not include any seed keywords still exist inside the convex hull. The keywords outside the convex hull are considered outliers in this case. Another addition was to do this process recursively, after all of outliers outside the convex hull are removed, the keywords are re-clustered again and apply same procedure until no more keywords embeddings lie outside of the convex hull. Same process applies to the circle same as the convex hull, both approaches were tested. Figure 2 further illustrate the process.

For approaches using the convex hull and circle, t-Distributed Stochastic Neighbor Embedding (t-SNE) was used as a dimensionality reduction method to be able to visualize the clusters and to handle all calculations. It is a highly effective method, particularly well-suited for the visualization of high-dimensional datasets. In the context of semantic keyword analysis, t-SNE can be instrumental in visualizing and understanding complex relationships among keywords. It works by converting similarities between data points into joint probabilities and aims to minimize the difference between these joint probabilities (or similarities) in the high-dimensional space and the low-dimensional space.

Figure 2 illustrates how first clustering iteration looks like and how the convex hull and circle enclose the clusters, which include any of the class seed keywords (represented in red). Thus, all of the keywords located outside of the convex hull will be considered as outliers and be removed. Moreover, it shows how the relevant keywords are re-clustered from the iteration before and again there are still some keywords located outside of the convex hull and will again be considered as outliers and be removed. Furthermore, it shows how the final iteration contains no more keywords outside of the convex hull, so iterations are stopped and collect all outliers from all iterations before.

3.3 Application of Outlier Detection Methods

Methods like Isolation Forest and Local Outlier Factor have been tried out. Both methods have same parameters however, the most important one is the contamination. The contamination controls the outliers percentage to be detected which was very hard to choose. One of the approaches was to retrieve outliers percentage per class from the clustering step and then feed it into the contamination parameter. Another approach was to choose the same contamination to all classes like 20% or 50% (the maximum contamination). The results will be shown in the Experiments and Results section.

3.4 Application of Keywords Similarity Methods

As a last filtering assurance step, keyword similarity methods like the familiar one which is the cosine similarity have been utilized.

Cosine similarity between two keywords embeddings $\mathbf{A}$ and $\mathbf{B}$ is given by:

$$\text{cosine similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| \times ||\mathbf{B}||}$$

where $\mathbf{A} \cdot \mathbf{B}$ is the dot product of vectors $\mathbf{A}$ and $\mathbf{B}$, and $||\mathbf{A}||$ and $||\mathbf{B}||$ are the magnitudes of vectors $\mathbf{A}$ and $\mathbf{B}$, respectively.

However, some of the keywords like *Mine* and *Mining* were noticed of not outputting high similarity scores. Another example is having *Natural Gas* as one of a class seed keywords and also cosine similarity could not relate that *Methane* and *Natural Gas* should be of high similarity scores. So two solutions to that problem were proposed.

The first idea was to use WordNet which is a large lexical database of English, developed and maintained by Princeton University. It's widely used in computational linguistics and natural language processing. It was used to derive synonyms for class seed keywords, so that when applying cosine similarity, there would be higher chances of similarity between actually similar keywords.

The other solution was to use ConceptNet which is a semantic network designed to help computers understand the meanings of words that people use. It is part of the larger Open Mind Common Sense project, which was started at the Massachusetts Institute of Technology (MIT) Media Lab. The main strength about ConceptNet in the problem, is that it is able to retrieve words which are related to a specific keyword. So for example it was able to get *Methane* as a word related to *Natural gas* as input.

3.5 Retrieval of Filtered Keywords

Three main approaches have been pointed out, to be able to remove outliers or irrelevant keywords of class-specific keyword sets. But there is still a missing question of how those three approaches are connected to finally retrieve the outliers for each class, which will be discussed in this subsection.

First, the clustering step is done and as an output a list of outliers X is retrieved with respect to the utilized clustering method. Second, a list of outliers Y is extracted from the outlier detection methods step using LOF and ISO. Then a union set of both is generated to finally have one proposed list of outliers Z.

$$Z = X \cup Y$$

The final step is to apply cosine similarity among the class seed keywords and their synonyms using WordNet or their related words using ConceptNet against the previously proposed list of outliers Z to finally get a filtered outliers list O after the cosine similarity, and obviously choosing a threshold.

4 Experiments and Results

In this chapter, an example dataset will be revealed. Moreover, the evaluation metrics of which the best practices are decided upon will be discussed, and also the results will be shown for all of the experimented practices.

Tables 3, 4, 5, 6, and 7 include all details about the experimented methods and their results, along with some performance related findings such as time to translate keywords for each translator.

4.1 Dataset

Dataset consists of 21 classes from class A to U, where each class represents a specific theme. A sample dataset for class **A** is presented. It includes the following fields in german with their corresponding text:

Name: *Land- und Forstwirtschaft, Fischerei*

Beschreibung in WZ2008: *Dieser Abschnitt umfasst die Nutzung der pflanzlichen und tierischen natürlichen Ressourcen. Dazu zählen Tätigkeiten wie Pflanzenbau, Tierzucht und Tierhaltung, Holzgewinnung und die Gewinnung anderer pflanzlicher und tierischer Erzeugnisse in land- oder forstwirtschaftlichen Betrieben oder in freier Natur.*

Potential Seed Keywords: *Landwirtschaft, Forstwirtschaft, Pflanzenbau, Tierzucht, Tierhaltung, Jagd, Holzgewinnung, Holzeinschlag, Veredlung landwirtschaftlicher Erzeugnisse, Fischerei, Aquakultur, Anbau von Pflanzen, Fallenstellerei*

Summary: *Dieser Abschnitt umfasst die Nutzung der pflanzlichen und tierischen natürlichen Ressourcen. Dazu gehören Tätigkeiten wie Pflanzenbau, Tierzucht und Tierhaltung, Holzgewinnung und die Gewinnung anderer pflanzlicher und tierischer Erzeugnisse in land- oder forstwirtschaftlichen Betrieben oder in freier Natur.*

Name+Summary+First5SeedKeywords: *Land- und Forstwirtschaft, Fischerei. Dieser Abschnitt umfasst die Nutzung der pflanzlichen und tierischen natürlichen Ressourcen. Dazu gehören Tätigkeiten wie Pflanzenbau, Tierzucht und Tierhaltung, Holzgewinnung und die Gewinnung anderer pflanzlicher und tierischer Erzeugnisse in land- oder forstwirtschaftlichen Betrieben oder in freier Natur. Dazu gehören: Landwirtschaft, Forstwirtschaft, Pflanzenbau, Tierzucht*

List of General Class Keywords to be Filtered: *["...flor", "ackerbau", "ackerbautreibender", "ackerbautreibendes", "ackerbauund", "ackerland", "Agr", "agrar", "agrarbereich", "agrarflächen", "Agrarier", "agrarstruktur", "agrarstrukturverbesserung", "agrarwirtschaft "agrarwirtschaftlicher", "Agrarwissenschaft", "agronomie", "Anbau von Pflanzen", "Angler", "Aquakultur", "Aquakulturen", "Argentum", "Bauern", "Bauerngut", "die Beine in die Hand nehmen", "Düse", "düsen", "eilen", "eingraben", "einpflanzen", "einsetzen", "erwerbsgartenbau", "Fallenstellerei", "Farmer", "Farmern", "Farmers", "farmerzeugnisse", "fegen", "fetzen", "Fischern", "Fischers", "Fischfang", "Fischfange", "Floren", etc.]*

Previous data illustrates how some keywords are not directly connected to the main class goal and description. An example of actual outliers are *"...flor"* and *"die Beine in die Hand nehmen"*

4.2 Evaluation Metrics

In this subsection dedicated to evaluation metrics, the performance of semantic keyword outlier detection is asessed by employing a robust suite of metrics. Specifically, recall, precision, and F1 scores are utilized to quantify the effectiveness of outlier detection algorithms. Recall, or the true positive rate, measures the algorithm's ability to correctly identify actual outliers within the dataset, providing insight into the sensitivity of the model.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where TP is the number of true positives and FN is the number of false negatives.

Precision, on the other hand, evaluates the proportion of true outliers among all identified outliers, reflecting the exactness of the algorithm.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Where FP is the number of false positives, which are actually not outliers.

The F1 score, which is the harmonic mean of precision and recall, serves as a balanced metric that considers both the precision and the recall to compute the test's accuracy.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

By integrating these three metrics, a holistic and nuanced evaluation of the model's performance is ensured, facilitating a thorough understanding of its strengths and limitations in detecting semantic keyword outliers.

Experiment Number	Translator	Text-Embedding Model	Hierarchical Clustering Method	Seed keywords Synonyms
1	deep translator-google	jina-embeddings-v2-base-en	< 15 members	No Cos Sim Step
2	deep translator-google	jina-embeddings-v2-base-en	< 20 members	No Cos Sim Step
3	deep translator-google	jina-embeddings-v2-base-en	< 10 members & remove furthest 25% of each cluster	No Cos Sim Step
4	deep translator-google	jina-embeddings-v2-base-en	recursive clustering based on cluster density & < 10 members	No Cos Sim Step
5	deep translator-google	jina-embeddings-v2-base-en	recursive clustering based on cluster density & keep only clusters containing any seed keyword	No Cos Sim Step
6	deep translator-google	jina-embeddings-v2-base-en	circle	No Cos Sim Step
7	deep translator-google	jina-embeddings-v2-base-en	convex hull	No Cos Sim Step
8	deepL	all-mpnet-base-v2	convex hull	WordNet
9	translators 5.8.9 – google	all-mpnet-base-v2	convex hull	WordNet
10	translators 5.8.9 – google & deepL	all-mpnet-base-v2	convex hull	ConceptNet
11	Lingvanex	all-mpnet-base-v2	convex hull	ConceptNet

Table 3: Experiments & Results done using translated English Keywords. **Notes: 1-** Please refer back to Subsection 3.2 for any clustering method description (Formatted in **Bold** Text style). **2-** Members mentioned in Hierarchical Clustering Method means clusters containing < number of members are considered outliers. **3-** Convex hull includes recursive clustering based on cluster density & recursive convex hull approach. **4-** Circle includes recursive clustering based on cluster density & recursive circle approach.

Experiment Number	Text-Embedding Model	Hierarchical Clustering Method	Seed keywords Synonyms
12	paraphrase-multilingual-MiniLM-L12-v2	convex hull	WordNet
13	gbert-large-paraphrase	convex hull	WordNet

Table 4: Experiments & Results done using German Keywords

Experiment Number	Recall	Precision	F1-Score
1	0.37	0.76	0.5
2	0.33	0.54	0.41
3	0.39	0.51	0.44
4	0.61	0.50	0.55
5	0.96	0.47	0.63
6	0.42	0.82	0.56
7	0.68	0.75	0.51
8	0.84	0.77	**0.80**
9	0.84	0.65	0.72
10	0.80	0.81	**0.80**
11	0.80	0.81	**0.80**
12	0.66	0.58	0.63
13	0.92	0.61	0.73

Table 5: Evaluating Average Results for all Methods in Tables 3 and 4

Experiment Number	Recall	Precision	F1-Score
14 (class A)	0.88	0.97	0.92
15 (class B)	0.15	0.82	0.25

Table 6: Evaluating Results for gpt3.5 trials on german keywords

5 Discussion

5.1 Takeaway Process Ideas

The first step was to choose which clustering method works best for the desired task. Hierarchical Agglomerative clustering, K-means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Latent Semantic Analysis (LSA) were all tested separately on a certain class. It was quite obvious after visualizing the results per method that, the Hierarchical clustering approach had the best results of all others. However, the results still needed to be improved.

Thus, other outlier detection methods were experimented such as Isolation Forest (ISO), Local Outlier Factor (LOF), Coresets, and Z-score. Although the first three methods showed promising results apart from the Z-score method, there was an api problem utilizing the Coresets method after a couple of tests. As a result, hierarchical clustering along with both Isolation Forest (ISO) and Local Outlier Factor (LOF) were decided to do some further tests with.

Translator	TTT/minute
deep translator-google	4.76
deepL	1.00
translators 5.8.9-google	5.71
translators 5.8.9-lingvanex	0.66

Table 7: Different Translator Services used for Performance Measuring

Lots of hybrid approaches were tried out. All of them are a combination of both hierarchical clustering outcome combined with ISO and LOF, but in different techniques for detecting the outliers in the clustering step. The ISO and LOF step was the same for all of the hybrid methods which is retrieving a union of results of both.

The first hybrid approach extracted the outliers based on the number of members within each cluster. As an example in Table 3 experiment 1, if a cluster has members less than 15, then all of its members will be considered as outliers. The second approach almost was similar to the previous one but with also removing furthest 25% of each cluster as shown in experiment 3.

The third hybrid approach involved recursive clustering based on cluster density, so simply re-clustering process stops when all clusters are of relatively high densities as in experiment 4. However, there was still a problem of considering clusters of members less than specific number as outliers, which is considered a limiting constraint, that it would be best if another general approach was found.

The fourth hybrid approach solved the obstacle that was faced in the previous approach, which is considering the number of cluster members limitation by only keeping the clusters which contain any seed keyword of that class, as in Experiment 5. However, there could be still other clusters which relate to the class in other clusters.

The fifth, and also sixth hybrid approaches made use of a circle and a convex hull enclosing all clusters which contain any class seed keyword as one of its members. Although the convex hull approach was not used before in any of the past researches of similar problem, it proved outstanding results when combined with WordNet or ConceptNet and involving cosine similarity as illustrated in Figure 2 and Subsection 3.5 Retrieval of Filtered Keywords.

5.2 Further Results Explanation

Table 3 shows and numbers all of the techniques experimented using translated english keywords, and the various hierarchical clustering aforementioned approaches, then followed by ISO and LOF. Seed Keywords Synonyms refers to the source used to retrieve class seed keywords synonyms to then be used for the cosine similarity calculation as a last step. Same for Table 4, but working on german keywords without translations.

Table 5 further includes the average results of recall, precision and f1-score based on expert manual annotation for 4 classes. The experiment number refers to Tables 3 and 4. Table 8 includes information about the annotated classes.

Class	Description	Number of Keywords
A	Involves exploiting natural plant and animal resources through farming, forestry, and fishing activities	476
B	Covers the extraction and processing of mineral resources, including fossil fuels and other minerals, through various mining methods and associated activities.	375
C	Deals with transforming raw materials from various industries into new goods, resulting in finished or semi-finished products	539
D	Concerns the provision of energy, particularly electricity, gas, and heat, through a network of infrastructure, regardless of its extent	231

Table 8: Annotated Classes Information

These results are the most interesting experimented approaches. Although experiments number 8,10, and 11 yield almost same results, experiment number 11 is recommended to be the best. That is due to the total execution time according to Table 2, which shows that 5.8.9-lingvanex translator has the fastest translation time 0.66 minute/class.

As illustrated before, Table 5 shows averaged results based on 4 classes described in Table 8. Thus, to avoid losing precision in average results, Table 9 explicitly breaks down the results per class for experiment 11.

Class	Recall	Precision	F1-Score
A	0.86	0.78	0.82
B	0.76	0.80	0.78
C	0.80	0.75	0.77
D	0.72	0.90	0.80

Table 9: Explicit Results Evaluation for Experiment 11 per class

It is also worth mentioning that gpt3.5 model has been tested to detect outliers from 2 classes as illustrated in Table 6. The prompt included the class name, seed keywords, summary, and the list of keywords to remove outliers from. However, there was a huge difference of both classes' results, which indicated that it is not reliable to use for this task for its high chance of hallucinations.

5.3 German to English Translations

Comparing results using german and english keyword shows that, english-text embedding models are better than the german ones despite that gbert-large-paraphrase model performed relatively well. The two main possible reasons for this are: **Data Availability:** English has a larger corpus of text available, which includes a wide range of domains and genres. This abundant data is crucial for the current research problem. **Complexity of the Language:** German, with its compound words and inflections, can be more complex in terms of syntax and morphology compared to English. This complexity can pose additional challenges for those kind of NLP related tasks

Furthermore, Table 7 lists the average Time To Translate (TTT/minute) all keywords per class of approximately 300 keywords each. Those results also supports using translators 5.8.9-lingvanex than other translators as it has translating speed of almost 7.5 times faster than google translate and 1.5 times faster than deepL.

6 Challenges & Future Work

Significant challenges were encountered, prominently in the domain of language translation and hyperparameter tuning. The reliance on translators markedly influenced the overall results. A notable fraction of keywords, originally in German, failed to be adequately translated, remaining in their original language. This presented a significant obstacle, as the incomplete or incorrect translation of keywords could lead to misinterpretation or misclassification, affecting the accuracy and reliability of the analysis.

Furthermore, the complexity of hyperparameter tuning presented another substantial challenge. Each stage, including clustering, outlier detection, and cosine similarity calculations, required intensive tuning of various hyperparameters. This process is both time-consuming and crucial for the effectiveness of the methodologies employed.

To address current challenges, future work should concentrate on two key areas:

Enhanced Translation Methods: Incorporating advanced neural machine translation models could substantially improve the accuracy and context-awareness of translations. Additionally, developing methods to verify translation accuracy and identify errors is crucial for data consistency.

Automated Hyperparameter Optimization: Utilizing automated hyperparameters tuning techniques can streamline the model optimization process, reducing manual effort, and potentially revealing more effective parameter configurations.

7 Conclusion

This research, forming a part of the larger Create-Data4AI (CD4AI) project, embarked on a quest to refine the process of extracting meaningful keywords from unstructured data. The principal objective was to enhance the relevancy and precision of these keywords to specific classes within diverse domains. The methodology adopted a two-pronged approach: advanced clustering techniques, including a novel convex hull method, and rigorous outlier detection strategies utilizing methods such as Isolation Forest and Local Outlier Factor.

One significant observation was the profound impact of language translation on the project's results. The challenge of translating lots of Class keywords from German to English underscored the necessity for accurate cross-lingual semantic analysis. Moreover, the process of hyperparameter tuning across various stages such as clustering, outlier detection, and similarity measurement added layers of complexity to the research.

The research's outcomes are promising, especially with the use of convex hull clustering. The hybrid approach of clustering, outlier detection, and semantic similarity analysis proved effectiveness in filtering out irrelevant class-specific keywords, enhancing the quality and applicability of the derived classes.

Future work should focus on improving translation mechanisms, perhaps leveraging advanced neural translation models, and exploring automated hyperparameter optimization to streamline and enhance the parameters tuning process. Such advancements are pivotal in elevating the project's ability to handle more complex, multilingual datasets and adapt to the ever-evolving landscape of semantic data analysis.

This study paves the way for more refined and accurate methods in data refinement, categorization and keyword analysis, offering valuable insights for further research and application in the field.

References

[1] Shweta Sharma, Neha Batra, et al. Comparative study of single linkage, complete linkage, and ward method of agglomerative clustering. In *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 568–573. IEEE, 2019.

[2] Md Zubair, MD Asif Iqbal, Avijeet Shil, MJM Chowdhury, Mohammad Ali Moni, and Iqbal H Sarker. An improved k-means clustering algorithm towards an efficient data-driven modeling. *Annals of Data Science*, pages 1–20, 2022.

[3] Ruitong Zhang, Hao Peng, Yingtong Dou, Jia Wu, Qingyun Sun, Yangyang Li, Jingyi Zhang, and Philip S Yu. Automating dbscan via deep reinforcement learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2620–2630, 2022.

[4] Nicholas Evangelopoulos, Xiaoni Zhang, and Victor R Prybutok. Latent semantic analysis: five methodological recommendations. *European Journal of Information Systems*, 21(1):70–86, 2012.

[5] Antonella Mensi, Alessio Franzoni, David MJ Tax, and Manuele Bicego. An alternative exploitation

of isolation forests for outlier detection. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops, S+ SSPR 2020, Padua, Italy, January 21–22, 2021, Proceedings*, pages 34–44. Springer, 2021.

[6] Omar Alghushairy, Raed Alsini, Terence Soule, and Xiaogang Ma. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing*, 5(1):1, 2020.

[7] Lingxiao Huang, Shaofeng H-C Jiang, Jian Li, and Xuan Wu. Epsilon-coresets for clustering (with outliers) in doubling metrics. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 814–825. IEEE, 2018.

[8] Vaibhav Aggarwal, Vaibhav Gupta, Prayag Singh, Kiran Sharma, and Neetu Sharma. Detection of spatial outlier by using improved z-score test. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 788–790. IEEE, 2019.