Boosting data findability: The role of AI-enhanced keywords

Kokila Jamwal[1]

Abstract

In today's data-driven world, finding relevant data in a vast expanse of information is increasingly important. Researchers have been exploring various methods to improve the findability, accessibility, interoperability, and reusability of data, for example, by using controlled vocabularies to enhance data findability. Although the use of controlled vocabularies is growing, challenges remain for findability when users provide their own keywords, known as user-defined keywords or do not provide keywords at all. Finding data in data archives based on metadata fields with user-defined or missing keywords is challenging, or even impossible. Here, we show the use of artificial intelligence (AI) techniques from the subfield of deep learning to automate the assignment of keywords using controlled vocabulary, leading to improved data findability. The main results demonstrate that AI automation performs well on the test set. In addition, we comapre our deep learning model against large language model (LLM) on the task of automated topic assignment. Automated topic assignments will reduce the time and effort required for data curation, enhancing data findability and usability for data producers and consumers. The application of AI to automate metadata assignment offers practical solutions for improving data findability and reusability, not only in research data archives but across various data-driven domains. Overall, this approach highlights the potential of AI in addressing data findability challenges, paving the way for more efficient and effective data discovery and utilization in the era of big data and information abundance.

Keywords

FAIR, user-defined keywords, text classification, AI, findability, controlled vocabulary

Introduction

Research data generation has seen an upward trend. Researchers generate research data to validate their findings and hence actively contribute to this upward trend (Steiner, 2023). In addition to this, new data collection methods like APIs and web scraping have added to the exponential growth in the volume of daily-generated research data. Due to this, managing research data has become more challenging. Metadata plays a crucial role in managing research data. Research data with rich metadata is easier to manage than research data with limited or no metadata. Supporting the idea of rich metadata, FAIR principles have caught the research community's attention. The FAIR principle focuses on making data findable, accessible, interoperable, and reusable.

Focusing on the findability aspect of FAIR principles, we investigate GESIS Search. GESIS Search[2] is a search platform from GESIS (GESIS – Leibniz Institute for the Social Sciences, based in Germany, is an infrastructure institute for the social sciences). The GESIS Search platform allows researchers to find

surveys and social science research data. GESIS Search provides necessary metadata fields along with research data in order to make research data findable for the researchers. In our paper, we focus on the 'Topics' metadata field. The 'Topics' metadata field generally takes in values from a controlled list of topics, known as controlled vocabulary (CV). Controlled Vocabularies improve data findability [3,4,5,6,7]. However, some studies in GESIS Search have either user-defined or no keywords, which leads to poor findability, as shown in Figure 1.

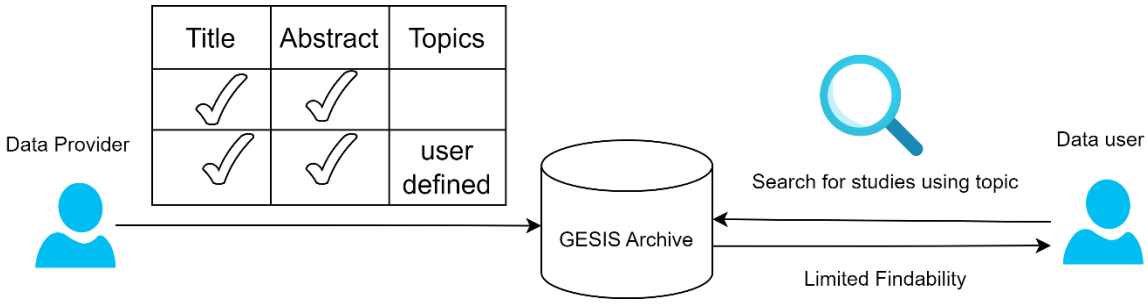

Figure 1: Limited Findability of research data due to missing and user-defined values in 'Topics' metadata field

In Figure 1, the data provider submits a study into the GESIS archive. However, the study might have missing or user-defined topics. When data users search for studies based on keywords in GESIS archive, the search results may be negative, leading to limited findability. There is a need to overcome this gap so that the findability of the studies can be further improved. For example, in one of the studies from GESIS Search, data depositors used ‘migration cost’ as a user-defined topic, which is not easily findable. However, replacing it with ‘migration’ from CV would make it more findable. Figure 2 shows the result of searching for a user-defined topic versus a CV topic on GESIS Search. There are no results for the user-defined topic, but the CV topics group similar studies together, making such studies more findable.

Artificial intelligence is beneficial in labeling data automatically. We specifically use the deep learning subfield of AI, which focuses on learning from complex data to make predictions. Leveraging AI techniques, we use an automatic topic classification model to label missing or user-defined keywords in the 'Topics' metadata field. These labels take values from a set of CV sources. In particular, the ‘Topics’ metadata field can have more than one label, making it a multi-label classification problem. As an input feature, we utilize text from the ‘Abstract’ metadata field and encode this text with a sentence transformer. The encoded abstract is then passed through a multi-label classification model, which outputs the labels from the CV sources. We will discuss the definitions and approach in detail in the following sections. Our work helps to improve data findability and reusability through AI techniques.

In summary, our contributions are as follows:

1) We investigate the presence of missing or user-defined keywords for the ‘Topics’ metadata field for GESIS research data.

2) We propose a deep learning-based multi-label classification model[1] for automatic labeling of the 'Topics' metadata field with values from CV. We evaluate the performance of our model based on various metrics such as precision (micro), recall (micro), F1 score (micro) and hamming loss, to measure the fraction of incorrectly predicted labels. These metrics will be discussed in detail in 'Experimental Setup' section.

3) We compare our deep learning-based classification model with a large language model (LLM), ChatGPT 3.5. Our evaluation results on multiple subsets of test data demonstrate that our model trained for topic classification performs better than the LLM.

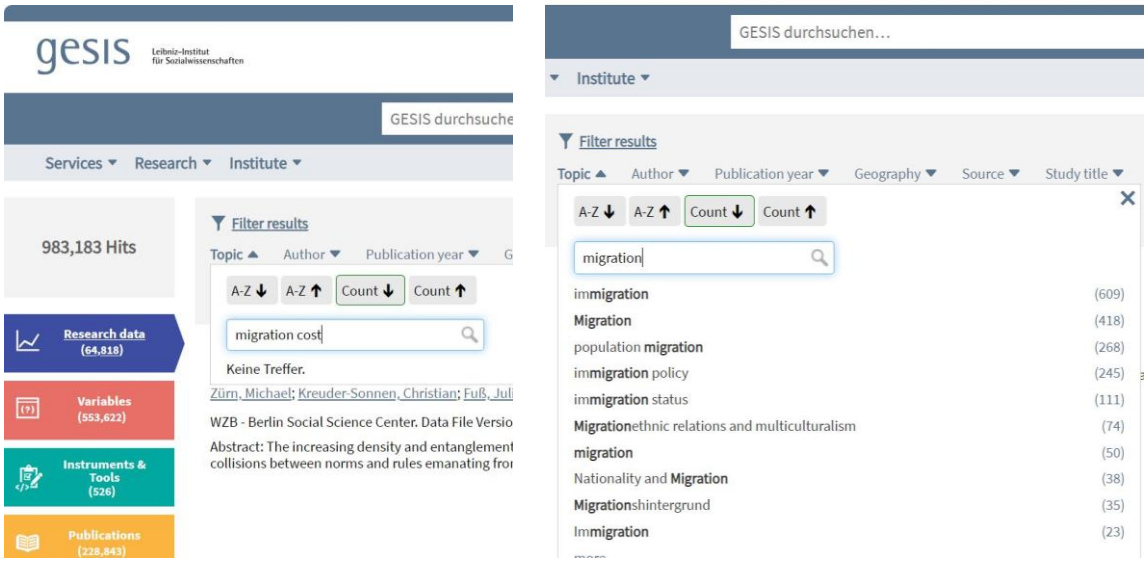

Figure 2: GESIS Search results based on filter 'Topic' for 'migration cost' vs 'migration'. ©GESIS

We structure the rest of our paper as follows: First, we formalize our problem statement. Then, we present our proposed multi-label topic classification model in the 'Approach' section. We then describe our 'Experimental Setup' section, including information on datasets, parameter settings and evaluation metrics. 'Evaluation Results' section assesses our approach using various metrics. In 'Use Case' section, we compare our model with LLM before wrapping up in 'Conclusion' section.

Definitions & Problem Formulation

We consider our topic assignment problem as a multi-label classification problem. Table 1 provides an example of a multi-label dataset.

Table 1 shows a multi-label dataset for movie reviews. Each movie review is an input instance of the dataset, and the corresponding set of labels in column 'Label' is the output label. Each output label can contain more than one value.

[1] https://github.com/kokila134/multi-label-classification

Movie review (X)	Label (Y)
"The movie is great mix of comedy and romance.",	["Comedy", "Romance"]
"I loved great action sequence in the climax of beautiful love story"	["Action", " Romance"]

Table 1: Example of multi-label dataset

Each study in GESIS Search contains metadata fields to describe the study. Figure 3 shows a study and its associated metadata fields in GESIS Search.

In this paper, we are interested in assigning CVs to the 'Topics' metadata field. For this, we only consider the 'Abstract' metadata field as an input feature as it generally contains the most textual information about the study.

Let a dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i \in X$ is an input instance and $y_i \subseteq Y$ is the set of labels associated with x_i.

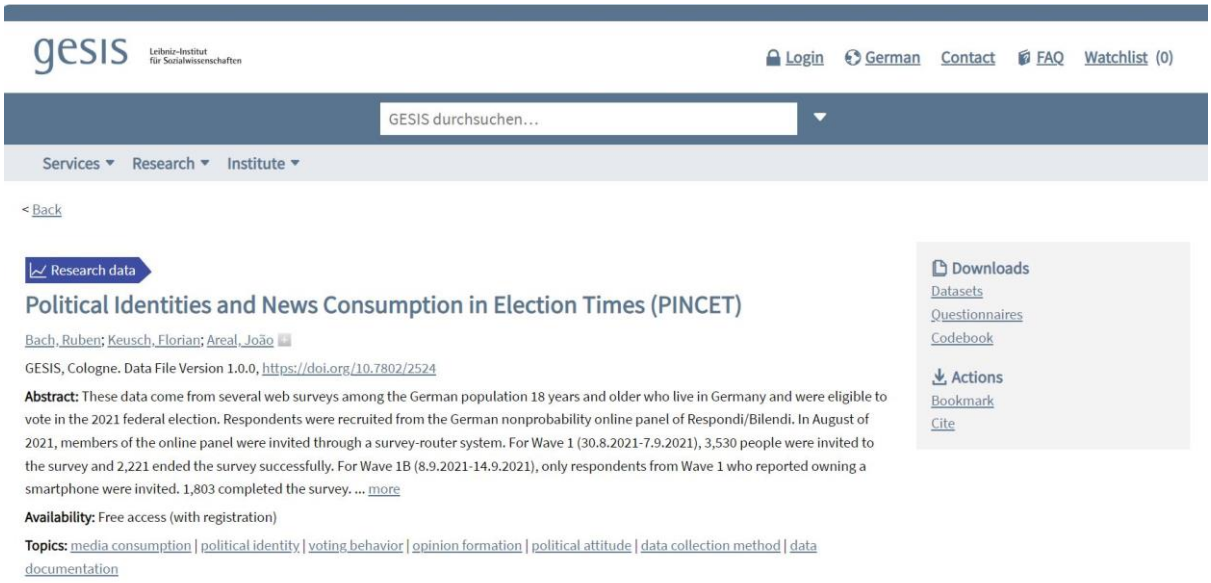

Figure 3: A study and its metadata fields in GESIS Search. ©GESIS

Input instance (x_i): Each input instance is the set of input features passed to a model for learning the patterns. An input instance is an entry in the dataset.
Output labels (y_i): Output labels, for each input instance, is a list of labels associated with the provided input instance.
Input feature: An input feature is an attribute of a specific input instance. In a dataset, especially one in a tablular format, an input feature refers to the value in a specific column for a given input instance (row).
The goal is to learn a function $f(x)$ that maps each input instance x to its corresponding set of labels y.

Multi-label classification ($f(x)$): Multi-label classification is a text classification problem in which each instance may belong to several predefined categories or classes simultaneously. In our case, these categories are values from CV sources.

Figure 4 shows a snapshot from our initial dataset, which contains two columns: Abstract and Label. Each row depicts a study in GESIS Search. An input instance and a list of output labels correspond to each study.

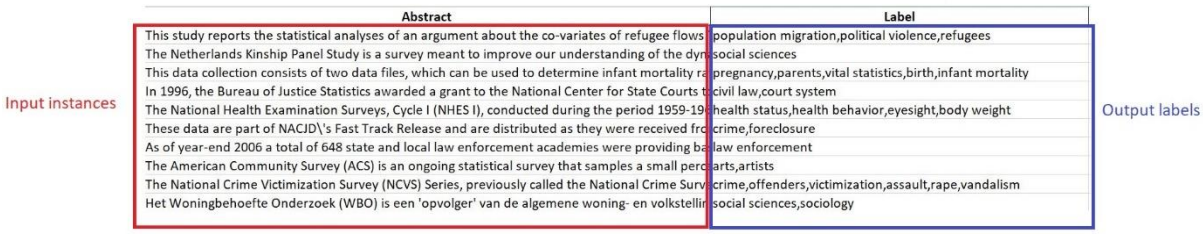

Figure 4: Snapshot of our initial dataset

Approach

In this section, we discuss our approach in more detail. We first discuss our data preprocessing pipeline in section 'Data preprocessing'. Next, we describe our embedding generation technique to embed the input instances. Finally, we discuss the multi-label classification model which outputs the class labels from CVs. The overall pipeline for our approach is illustrated in Figure 5.

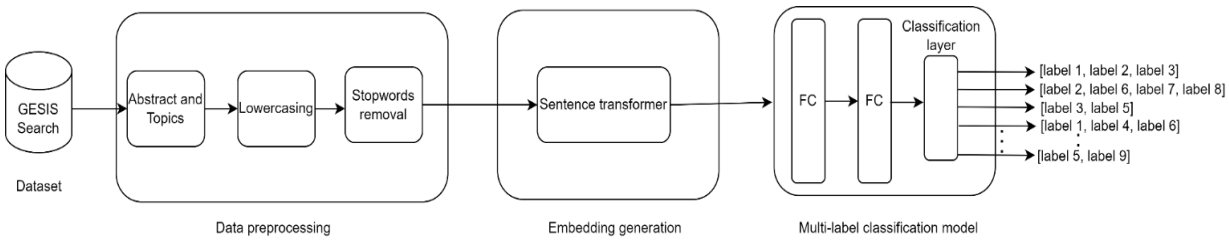

Figure 5: Overall pipeline for our approach

Data preprocessing

GESIS Search is a platform that allows researchers to find information about social science research data, publications on research data, and open-access publications. The studies in GESIS Search include metadata about the study or research data. Each study contains metadata fields in descriptive, methodological, and bibliographic metadata categories. In descriptive metadata, the studies contain 'Abstract' and 'Topics' metadata fields among others. The 'Abstract' metadata field allows the researchers to summarize the study they are publishing. The 'Topics' metadata field provides keywords from a set of controlled vocabulary (CV), which provide a better picture of the study. For the 'Topics' metadata field, the research users generally select more than one value from the CVs. However, some studies either have user-defined values in the 'Topics' metadata field or have no value. Considering that, we filter studies based on the number of CV topics. We select the studies containing more than one CV topics for training purposes. Later, we remove the duplicate studies at the end of the data preprocessing pipeline.

A controlled vocabulary is a controlled list of values from which the relevant values are selected for a specific metadata field. Multiple sources of controlled vocabularies have been used in social science research studies at GESIS, such as the STW Thesaurus for Economics and TheSoz thesaurus. In our paper, we select CESSDA Topic Classification, DPRex, ISO 3166-1 Country Codes, Kategorienschema

Wahlstudien, STW thesaurus for Economics, TheSoz thesaurus, and UNESCO thesaurus as the controlled vocabulary sources[8]. The values from these sources are combined to create a unified list. The duplicated values are removed and converted into lowercase for mapping to CV topics as shown in Figure 6.

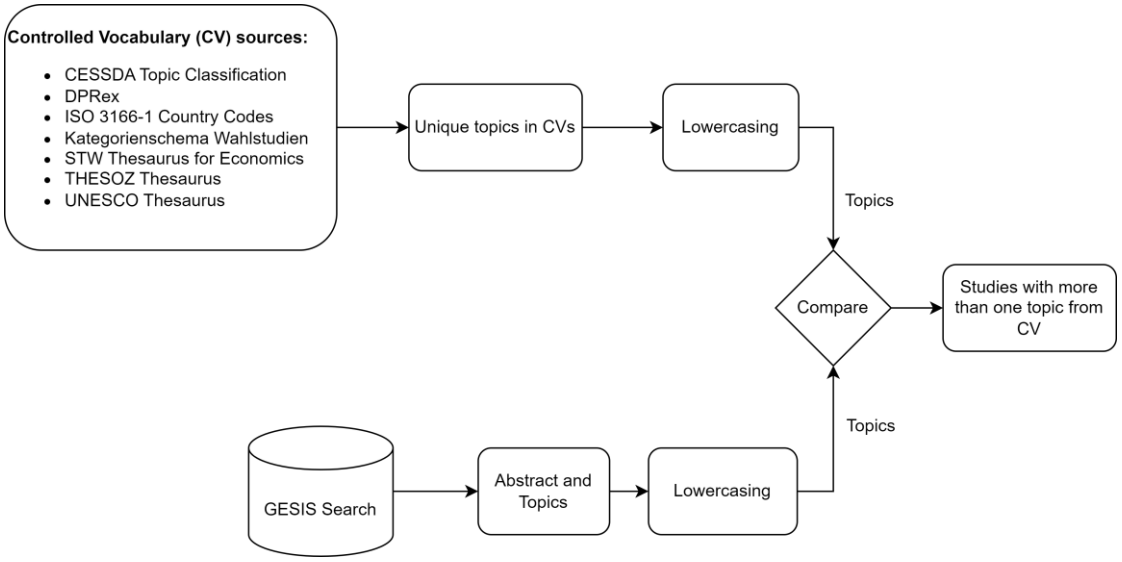

Figure 6: Data preprocessing pipeline

Embedding generation

Deep learning models do not understand raw textual data, so we need to convert the textual data into numerical form. Embeddings are numerical representations of data that captures semantics of the data, making it possible to understand real-world data effectively. In this paper, data in the 'Abstract' metadata field is in text format. We utilize the textual information from the 'Abstract' metadata field and capture its semantic meaning through our embedding generation component. To achieve this, we apply a sentence transformer (Reimers and Gurevych, 2019) to transform text in the 'Abstract' metadata field into embeddings. Since most of the text in the 'Abstract' metadata field across studies is multilingual (mostly in German and English), we utilize a multilingual sentence transformer called 'distiluse-base-multilingual-cased-v1' to generate the embeddings for 'Abstract'. These embeddings are vector representations that encapsulate the semantic essence of original sentences. Figure 7 depicts an example of embedding generation.

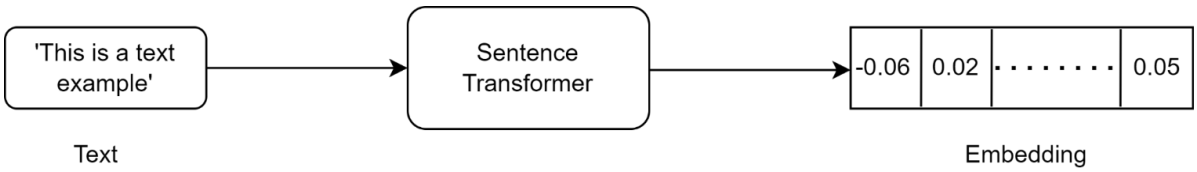

Figure 7: Sentence Transformer based text to embedding generation

Multi-label classification model

The multi-label classification model is the last component of our approach. It takes the embeddings generated for each study as input and passes them through a series of fully connected layers (FC) to

provide a list of output labels associated with the input instance. Figure 8 illustrates the multi-label classification model.

The multi-label classification problem differs from other classification problems[9]. The basic form of classification problem is the binary classification problem, where the output label for an input instance is only one of the two available output labels. Other than binary classification problem, there is multi-class classification problem. In a multi-class classification problem, an input instance can have only one associated output label from among more than two output labels.

Due to mutually exclusive classes in binary and multi-class classification, we deal with them differently than multi-label classification problems. Multi-label classification problems are generally dealt with through problem transformation or algorithm adaptation methods (Pant et al., 2019). As the name suggests, problem transformation methods transform the multi-label problem into a set of binary classification problems, which are then handled using algorithms for single-class classifiers. The algorithm adaptation methods adapt the algorithms and perform multi-label classification directly instead of

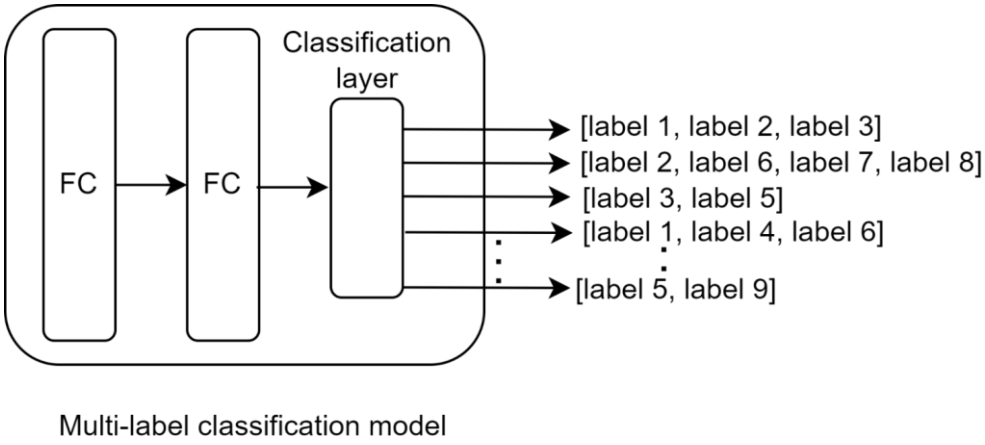

Figure 8: Multi-label classification model

simplifying the problem as a binary classification problem. We employ the problem transformation method by dividing the task into a series of binary classification problems. At the end of this component, we get the predicted output labels for each input instance.

Experimental Setup

In this section, we discuss our experimental setup including details on the datasets used, parameter settings, and the evaluation metrics used to evaluate the performance of our approach. All experiments are conducted on a laptop with 16 GB of RAM, and an Intel Core i7 processor. The software environment includes Python 3.11.9, TensorFlow 2.12.0, and other necessary libraries.

Datasets

We extract the dataset from GESIS Search via Elasticsearch[10]. Elasticsearch allows the users to store, search, and analyze huge volumes of data quickly. Each row in our dataset represents a research study, and the column 'Abstract' acts as a feature of the study. The 'Topics' is the set of labels associated with the study. For our experiments, we extract 64,791 studies from GESIS Search, which depicts the

number of instances in our dataset. Among these, the total number of studies with missing values in the 'Topics' metadata field is 15,776. Considering that, we select 44,417 studies containing more than one CV topics. Later, we de-duplicate the studies, resulting in 36474 studies at the end of the data preprocessing step. The number of unique labels from the CV sources equals 1,221. For more clarity on the dataset, we extract the following dataset properties for our multi-label dataset (Sorower, 2010).

- Distinct Label Set (DL): It is the total number of distinct label combinations in a dataset
- Proportion of Distinct Label Set (PDL): It is the measure of the number of distinct label sets per input instance
- Label Cardinality (LCard): It is the average number of labels per input instance. LCard is a measure of 'multi-labelledness'
- Label Density (LDen): It is measured as LCard normalized by the number of labels, which means LDen is the ratio of LCard to the number of distinct labels.

The values for each of these properties are depicted in Table 2.

Property	**Value**
Distinct Label Set (DL)	7,237
Proportion of Distinct Label Set (PDL)	0.198415
Label Cardinality (LCard)	5.966387
Label Density (LDen)	0.002699

Table 2: Properties of our multi-label dataset

Since Distinct Label Set (DL) and Proportion of Distinct Label Set (PDL) depict the diversity and coverage of labels in a dataset, they are essential to determine if the model needs to handle a wide range of labels or a more limited subset. The value for DL is 7,237 and since the number of unique studies is 36,474, value for PDL is 0.198415. Label Cardinality (LCard) and Label Density (LDen) provide information about the average label load per input instance. High LCard and LDen values indicate that dataset instances are likely to have multiple labels, necessitating robust multi-label handling capabilities in the model. Our dataset averages 5.966387 labels per instance, and the number of unique labels is 2,210, therefore the value of LDen is 0.002699.

Parameter Settings

Deep learning models use different hyperparameters. The hyperparameters control the learning process of a model. In order to get the best model results, these hyperparameters can be optimized using different techniques. We have employed one such hyperparameter optimization technique called grid search, which exhaustively searches the best parameter over all possible combinations of hyperparameters. During the multi-label classification model step, the output embeddings of size 512 each, created from the embedding generation step, are taken as input. The model consists of a series of fully connected (FC) layers. The fully connected layers consist of neurons. Neurons in deep learning models are nodes through which data and computations flow. Neurons receive one or more input signals, perform some calculations, and send output signals to the next layer of neurons. Based on the

hyperparameter optimization, the number of neurons in fully connected layer 1 (FC1) is 1,024, and in fully connected layer 2 (FC2), it is 256. The size or number of neurons in each layer represents the number of features leaned from the input data by this layer. Both fully connected layers use Rectified Linear Unit (ReLU) as the activation function. Activation functions add non-linearity to the model and help the model learn complex relationships in data. For the classification layer, the sigmoid function is used as the activation function, with an output size equal to the number of unique topics.

We split the dataset into training and testing sets for the multi-label classification problem, with 80% data in training and 20% in testing sets, respectively. The training set is passed to the multi-label classification model to learn patterns in the data, whereas the test set tests the model's performance. During training, we can set other parameters like batch size, learning rate, and number of epochs. Batch size is the number of training samples processed before the model's internal parameters are updated. The learning rate controls the rate or speed at which the model learns. The number of epochs signifies the number of times the entire training set is passed through the model. Based on the hyperparameter optimization for our model, the batch size is 128, the learning rate is set to 0.001 for the Adam (Adaptive Moment Estimation) optimizer. Optimizers are algorithms or methods used to change the attributes of neural network such as weights and learning rate for better learning by the model. The number of epochs is set to 200. We use early stopping as a regularization technique to avoid overfitting. Regularization techniques are methods used to improve a model's ability to generalize to new data by adding constraints to the learning process. Early stopping is a regularization technique used to prevent overfitting in training deep learning models. Overfitting occurs when a model becomes too specialized to the training data, capturing noise or irrelevant patterns, leading to poor generalization of unseen data. Early stopping monitors the model's performance on a validation set during training and halts the training process when the model's performance degrades, indicating it has started to overfit. If the model overfits, it will not perform well for the new unseen data.

Evaluation Metrics

The performance of our model for multi-label classification is evaluated using following metrics (Sorower, 2010):

- Micro-averaging: It is calculated by aggregating the contributions of all classes to compute the average metric. It considers the proportion of each class in the overall population.
    - Precision: The ratio of correctly predicted instances for a class to the total instances predicted as that class.
    - Recall (Sensitivity or True Positive Rate): The ratio of correctly predicted instances for a class to the total instances that belong to that class.
    - F1 Score: F1 Score is the harmonic mean of precision and recall, providing a single metric that balances both.
- Hamming loss: Hamming loss measures the fraction of incorrectly predicted labels (both false positives and false negatives) over the total number of labels.

Evaluation Results

In this section, we evaluated our model's performance through the evaluation metrics defined in section 'Evaluation Metrics'. The result of our evaluation is provided in Table 3. The value of all the

metrics used for evaluation ranges between 0 and 1. The higher the precision, recall, and F1 Score value, the better the model's performance. However, a lower value for hamming loss depicts better performance. The precision value is 68%, recall is 56%, F1 score is 61%, and hamming loss is 0.27%.

We asked human annotators to validate the performance of our model. We provided the annotators with a model-labeled sample of 20 instances, with ten random instances containing no label and ten random instances containing user-defined labels. The annotators got text in the 'Abstract' metadata field and the labels predicted by our model. We asked two human annotators to label the predictions per instance into two categories: the number of correct labels and the number of incorrect labels. The annotators indicated the number of labels they consider correct and the number they consider incorrect in the two categories, respectively. In case of disagreement between the two annotators, we asked a third annotator to resolve the disagreement. Ultimately, we took the labels generated by the third annotator after resolved annotations. Our model predicted 74% correct and 26% incorrect labels based on this sample. This experiment validated the performance of our model; however, we can further improve the performance by including more CV sources and input instances.

Metrics	Value
Precision (Micro)	0.6828
Recall (Micro)	0.5577
F1 Score (Micro)	0.6140
Hamming Loss	0.0027

Table 3: Evaluation results for our model

Use Case: A comparison of our model with LLM

We compared our model's performance with large language models (LLMs). As LLM, we used OpenAI's ChatGPT 3.5. We randomly took multiple samples, each containing 30 studies from the test data to carry out this experiment. To compare the performance of our model with LLM, we labeled the input instances using our model and using LLM with relevant labels from the CV. We compared our model with LLM during this experiment; the detailed comparison is in Table 4. We provided the ChatGPT with a prompt, including the problem statement, one 'Abstract' text at a time and the set of all possible values from CV. We asked ChatGPT to provide five labels for each 'Abstract' text. A sample prompt and result from ChatGPT is provided in the Figure 9. We averaged the performance of our model and LLM across the samples and carried out the Friedman test (Bogatinovski et al., 2022) to calculate the significance of our results. Table 4 provides the mean and standard deviation of our model and LLM across various metrics. According to the results, our model performed significantly better for the multi-label classification task on the test data samples.

Metrics	Value (Our model)	Value (LLM)
Precision (Micro)	0.85 ± 0.031	0.35 ± 0.02
Recall (Micro)	0.60 ± 0.10	0.31 ± 0.16
F1 Score (Micro)	0.69 ± 0.08	0.31 ± 0.08
Hamming Loss	0.02 ± 0.004	0.0528 ± 0.0012

Table 4: Comparison of evaluation results for our model vs LLM

User:

'Below is the list of labels, stored in "lol" variable:

['health status', 'larceny', 'disease', 'television news', 'aids', 'ethnicity', 'consumer attitudes', 'economic conditions', 'political change', 'pollution', 'school classes', 'legislation', 'government', 'opec', 'united states', 'information sources', 'drug use', 'european parliament', 'political participation', 'sports', 'quality of life', 'nationalism', 'presidency', 'social behaviour and attitudes', 'occupations', 'civil rights', 'alcohol', 'mental health', ...].

I am dealing with multi label classification. The aim is to classify the abstract into multi labels from "lol". Below is the abstract. Please label it with five relevant labels from the above list, "lol" in python list format. Do not make your own labels.

Abstract: The purpose of the National Health Interview Survey (NHIS) is to obtain information about the amount and distribution of illness, its effects in terms of disability and chronic impairments, and the kinds of health services people receive.....

Please select only the values from lol variable.

ChatGPT:

Based on the abstract provided and the list of labels, here are five relevant labels in Python list format:

relevant_labels = ['health status', 'disability', 'health care', 'social sciences', 'vital statistics']

Figure 9: Sample ChatGPT prompt and results

Conclusion

Researchers have focused on making their data FAIR. Data findability and reusability are among the main pillars of FAIR principles. Using controlled vocabulary improves findability. A controlled list helps to restrict the values for a particular metadata field, ensuring further data findability and reusability later. However, missing and user-defined topics hinder data findability and reusability.

We have devised an approach to automatically label studies for the 'Topics' metadata field. The 'Topics' metadata field labels are taken from a set of CV sources. Our topic classification model efficiently classifies the studies into multiple topic labels based on the 'Abstract' metadata field. Our model can help people across the whole lifecycle of research data. For data depositors, it can help make their data findable and reusable. Our model can help save the time and effort required for data curation and can easily manage larger volumes of data to help the data curators. Our model allows data users to enjoy improved findability with enhanced topic discovery. We compared our model with Chat GPT 3.5 for the same task, and our model proved to work better with the random sample from the test data. In the future, we will try to improve our classification model and examine the performance of different open-source LLM models for assigning topic labels to a study.

Acknowledgements

I would like to thank Libby Bishop (GESIS) for her suggestions and comments on the manuscript.

References

Bogatinovski, J., Todorovski, L., Džeroski, S., & Kocev, D. (2022). Comprehensive comparative study of multi-label classification methods. *Expert Systems with Applications*, 203, 117215.

Pant, P., Sai Sabitha, A., Choudhury, T., & Dhingra, P. (2019). Multi-label classification trending challenges and approaches. *Emerging Trends in Expert Applications and Security: Proceedings of ICETEAS 2018*, 433-444.

Reimers, N., Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In: EMNLP-IJCNLP 2019. ACL.

Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, *18*(1), 25

Steiner, G., (2023). The exponential growth of research data. *Analytical Science Magazine*, Vol. 3 - May/23 (https://analyticalscience.wiley.com/content/article-do/exponential-growth-research-data) [Accessed 27/06/2024]

Endnotes

[1] Kokila Jamwal, GESIS-Leibniz Institute for the Social Sciences, Unter Sachsenhausen 6, 50667 Cologne, Germany, kokila.jamwal@gesis.org

[2] GESIS Search (https://search.gesis.org/?source=%7B%22query%22%3A%7B%22bool%22%3A%7B%22must%22%3A%7B%22match_all%22%3A%7B%7D%7D%2C%22filter%22%3A%5B%7B%22term%22%3A%7B%22type%22%3A%22all%22%7D%7D%5D%7D%7D%7D&lang=en) [Accessed 15/05/2024].

[3] Metadata for Data Management: A Tutorial: Controlled Vocabularies (https://guides.lib.unc.edu/c.php?g=8749&p=44502) [Accessed 19/06/2024].

[4] Controlled vocabularies (https://www.cms.hu-berlin.de/en/dl-en/dataman-en/share/documentation/controlled-vocabularies) [Accessed 19/06/2024].

[5] Controlled Vocabulary for DAM (https://digitalassetmanagementnews.org/dam-guru-archive/controlled-vocabulary-for-dam/) [Accessed 19/06/2024].

[6] Controlled Vocabulary Policy (https://www.toronto.ca/wp-content/uploads/2022/05/8f0a-controlledvocabularypolicy2021.pdf) [Accessed 19/06/2024].

[7] Controlled Vocabulary (https://www.informedbyte.com/services/controlled-vocabulary) [Accessed 21/06/2024].

[8] GESIS Controlled Vocabulary Service (https://lod.gesis.org/en/) [Accessed 20/05/2024].

[9] Difference: Binary vs Multiclass vs Multilabel Classification (https://vitalflux.com/difference-binary-multi-class-multi-label-classification/#:~:text=To%20summarize%2C%20binary%20classification%20is%20a%20supervised%20machine,predict%20one%20or%20more%20classes%20for%20an%20item.) [Accessed 15/04/2024].

[10] Elasticsearch (https://www.elastic.co/elasticsearch) [Accessed 21/05/2024].