NLP based Conceptual Coding

Rishabh Bhonsle

Swetha Siripurapu

Advisor: Dr. Charles Gomez

Date: December 20, 2024

INDEX

ABSTRACT

The manual coding of interview transcripts is a slow and resource-intensive process that limits the ability of researchers to work with larger datasets. Currently, qualitative analysts manually assign conceptual themes to text using tools like **MAXQDA**, which requires significant time and coordination among multiple researchers. To overcome these challenges, this project, **NLP-Based Conceptual Coding**, aims to automate the conceptual coding process using Natural Language Processing (NLP) techniques. By reducing manual effort, this approach allows researchers to analyze a greater volume of qualitative data without the constraints of human scheduling for coding tasks. We utilize both foundational models, such as Term Frequency-Inverse Document Frequency (**TF-IDF**) and Bag of Words (**BoW**), alongside advanced pretrained models like **RoBERTa** and **DeBERTa**. These models are paired with **neural networks** to predict conceptual codes for different sections of interview transcript data. The transcripts contain insights into scientific careers, global collaborations, and researchers' perspectives. The manually coded data provided by analysts at The **Global Knowledge Lab** will serve as a benchmark to evaluate our model's accuracy. By combining traditional text encodings, neural networks, and advanced machine learning models, this project offers a faster, consistent, and scalable solution for conceptual coding, enabling the efficient analysis of large datasets and enhancing research capabilities.

1. INTRODUCTION AND LITERATURE REVIEW

BACKGROUND

Qualitative research involves analyzing textual data to derive themes, categories, and insights. This process, while essential for understanding nuanced perspectives, often relies on labor-intensive manual coding performed by researchers. For example, interview transcripts must be read, annotated, and grouped into fine-grained categories such as NLP Codes (e.g., Motivation, Ethics) and higher-level groupings such as Conceptual Codes (e.g., Politics, Morals, or Research Impact). Traditional manual methods are inherently subjective, time-consuming, and difficult to scale for large datasets, thereby limiting the scope of qualitative research.

Recent advancements in Natural Language Processing (NLP) have enabled the automation of these processes through a combination of traditional methods and advanced models. Foundational approaches like Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW) provide simple but effective representations of text. Meanwhile, sentence encodings from pretrained transformer models, such as BERT (Bidirectional Encoder Representations from Transformers), RoBERTa (Robustly Optimized BERT), and DeBERTa (Decoding-enhanced BERT with disentangled attention), allow for a deeper understanding of contextual and semantic nuances. These sentence embeddings, paired with neural networks, can be fine-tuned for tasks such as hierarchical text classification, making them particularly well-suited for complex qualitative coding.

This project proposes an end-to-end framework to automate qualitative coding by integrating multiple NLP techniques, including traditional encodings, sentence embeddings, and neural network architectures, for two prediction tasks:

1. Classifying **NLP codes** directly from preprocessed interview text using a combination of base models (TF-IDF, BoW) and other encoding techniques.
2. Predicting **Conceptual Codes** from the intermediate NLP code representations using a secondary classifier built on neural networks and transformers.

PROBLEM STATEMENT

Manually coding interview transcripts into both fine-grained and high-level categories is impractical for large datasets. The key challenges addressed in this study are:

1. **Scalability:** Enabling researchers to process large volumes of qualitative data without the constraints of manual effort and scheduling.
2. **Efficiency**: Reducing the time required for coding through automation.
3. **Accuracy**: Ensuring consistent and reliable coding performance.
4. **Hierarchy**: Integrating NLP codes and conceptual codes into a unified pipeline.

By leveraging transformer-based models with custom classification layers, this project develops a scalable, accurate, and interpretable solution to automate hierarchical coding of interview data. The primary objective is to develop a transformer-based model architecture that can predict NLP codes from textual data and further map these NLP codes to corresponding conceptual codes using a secondary classification layer. This approach integrates hierarchical predictions, where NLP codes serve as intermediate outputs and conceptual codes act as the final outputs, mirroring real-world research workflows. By evaluating the accuracy, performance, and reliability of this framework, we aim to provide an efficient solution for automating qualitative coding tasks. The proposed method represents a novel approach that can be adapted to various domains, such as healthcare, education, and social sciences, where qualitative coding plays a critical role in data analysis.

2. DATA

OVERVIEW

The dataset used in this project consists of 130 interview transcripts, providing detailed qualitative insights into various aspects of scientific careers, global collaborations, and researchers' perspectives. Each transcript was annotated with NLP Codes and 11 others with Conceptual Codes to capture both fine-grained and higher-level thematic information.

NLP Codes: A total of 102 codes, categorized into the following thematic areas: Background, The Researcher, Research Impacts, Outlets, State of the Field, Career Mobility and Time Abroad, Structure of the Field, The Global Science, International Collaboration and Competition, Biases and Prejudices, Scientific Internationalism and Scientific Cosmopolitan, Science and Society. These codes often intersect or form subsets of the broader Conceptual Codes.

Conceptual Codes: A total of 7 high-level themes: Scholarly Positioning and Motivation; National or International Context; Discipline and Knowledge Production; Personal Turning Points; Politics, Ethics, and Morals; Research Impact on the Real World; Research Impact on Academia

LIMITATIONS

1. **Variability in Text Length and Encoding Challenges**

The transcripts vary widely in sentence length, creating challenges during encoding. Short sentences linked to specific codes often require significant padding, which can dilute the embeddings with irrelevant noise. On the other hand, long sentences may be truncated, leading to a loss of critical information. This variability complicates the normalization process and impacts the model’s ability to handle text uniformly.

2. **Class Imbalance and Rare Codes**

   The dataset includes 102 NLP Codes, but their distribution is uneven. Many codes are underrepresented, making it difficult for the model to learn meaningful patterns. Class imbalance leads to biased predictions and limits the model's ability to generalize effectively across all codes, particularly for those with sparse representation.

3. **Overlapping and Ambiguous Code Hierarchies**

   The NLP Codes often intersect or act as subsets of the broader Conceptual Codes, resulting in ambiguous mappings. For instance, the same text segment may be relevant to multiple NLP and Conceptual Codes, introducing noise into the model's training process. This overlap challenges the model’s ability to disentangle nuanced relationships and accurately classify hierarchical themes.

4. **Context Loss due to Segmentation**

   To facilitate encoding, transcripts are segmented into smaller text chunks. However, this segmentation can strip away contextual relationships between sentences, which are often critical for qualitative coding. This loss of context particularly impacts codes requiring an understanding of narrative flow or dependencies across sentences.

5. **Subjectivity in Annotations**

Manual coding by researchers, while detailed, is inherently subjective. Differences in interpretation can introduce inconsistency in the labels, which the model may inadvertently learn. This annotation bias affects both the reliability of the training data and the model's ability to generalize to new datasets.

3. DATA PREPROCESSING AND EDA

To understand the dataset and identify patterns for NLP-based conceptual coding, a thorough Exploratory Data Analysis (EDA) was conducted, focusing on the following:

1. **Code Distribution Analysis:** Evaluated the frequency distribution of NLP and Conceptual Codes across the dataset to identify class imbalances and underrepresented categories.
2. **Text Characteristics:** Analyzed sentence lengths and token counts to assess the variability and suitability for encoding techniques. Identified outliers, such as excessively short or long sentences, that could affect model performance.
3. **Group-based Aggregation:** Grouping analysis revealed patterns and trends within the dataset by categorizing responses based on metadata such as roles, theaters, locations, and gender. These groupings not only provided valuable insights into the dataset but also suggested practical use cases for qualitative research and broader applications.
    - Understanding Demographic Trends: Grouping by role (e.g., Postdocs vs. Professors) and theater (e.g., AI vs. Genetics) helps researchers identify who contributes the most to specific topics. For instance, the AI theater has high participation from Assistant Professors, which may highlight the engagement of early-career researchers in this domain.

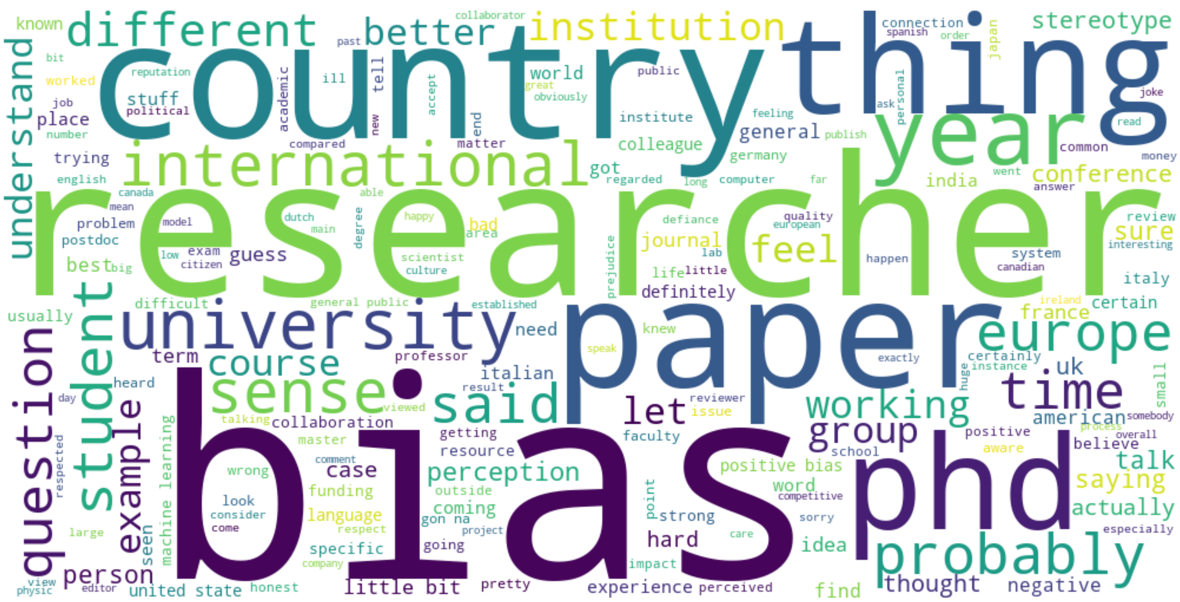
- Geographic Insights: Grouping by location reveals trends in global collaboration. For example, the predominance of the United States in AI and Space indicates hotspots of research activity, which can inform funding decisions or collaboration strategies.
- Gender Representation: Grouping by gender helps identify disparities. The low female representation in the AI theater, compared to a more balanced ratio in Genetics, points to areas where diversity efforts might be needed.
- Domain-Specific Analysis: Grouping by theater provides a lens into domain-specific trends. For instance, the Genetics theater's smaller dataset with a more balanced gender ratio contrasts with AI's larger dataset but skewed demographics, suggesting different challenges and opportunities for qualitative researchers in each domain.
- Sentiment-Based Insights: By grouping responses by sentiment, researchers can tailor recommendations for themes that elicit positive or neutral tones. For instance, slightly positive sentiment in Questions 2, 3, 4, 6, and 7 can highlight areas of success, while neutral sentiment in others may indicate a need for further exploration or intervention.
- Research Policy and Strategy: Identifying patterns across roles, theaters, and geographies can guide research policy, funding allocation, and collaboration strategies. For example, higher Postdoc participation in Space and AI can inform mentorship and support programs for early-career researchers.

4. **Sentiment Analysis:** Computed sentiment polarity using TextBlob for grouped responses to understand the emotional tone within different NLP Codes. Questions 2, 3, 4, 6, and 7 exhibited slightly positive sentiment, reflecting optimistic perspectives on themes like collaboration and scientific impact. Other questions yielded predominantly neutral sentiment, focusing on factual or descriptive content rather than emotional tones.
5. **Key Phrase Extraction:** To gain insights into recurring themes and concepts within the dataset, three methods of key phrase extraction were applied -
    a. TF-IDF: Using the TfidfVectorizer, key phrases were extracted by identifying terms with the highest TF-IDF scores for each question and response. This method highlighted terms that were significant within individual transcripts but less frequent across the entire dataset, making it effective for identifying localized themes. However, TF-IDF tends to favor unigrams and may miss multi-word phrases critical for qualitative analysis.
    b. RAKE (Rapid Automatic Keyword Extraction): RAKE was employed to extract domain-independent keywords by analyzing word co-occurrence patterns and excluding stopwords. This method was particularly effective in identifying multi-word phrases and contextually significant terms, such as "scientific collaboration" and "global impact." RAKE provided enriched representations of text that were useful for subsequent hierarchical coding tasks.
    c. NER (Named Entity Recognition): Named Entity Recognition (NER) was applied using spaCy to extract proper nouns and entities such as locations, organizations, and individuals. This method proved useful for identifying metadata-like information embedded within transcripts, including countries of collaboration and

institutional affiliations. However, NER had limitations in capturing thematic keywords that are not proper nouns.

6. **Topic Modeling:** Topic modeling was conducted using LDA (Latent Dirichlet Allocation) and NMF (Non-negative Matrix Factorization) to uncover latent themes in the dataset.
    a. LDA: LDA identified recurring topics by treating each transcript as a mixture of topics. Despite attempts to fine-tune the number of topics and coherence score, LDA struggled to generate clearly interpretable topics for certain subsets of the data. Key terms in the identified topics often overlapped, suggesting limited separation of themes.
    b. NMF: NMF, leveraging TF-IDF input, provided more coherent and interpretable topics compared to LDA. For instance, topics related to "international collaboration" and "career mobility" were clearly delineated. However, the interpretability of results depended heavily on the quality of the preprocessing and the choice of hyperparameters like the number of topics.
7. **Visualization:** Generated word clouds and other visual representations for frequently occurring terms and themes to validate and refine code assignments.

Example: Biases and Prejudices



Preprocessing:

A robust preprocessing pipeline was implemented to prepare the dataset for NLP modeling, ensuring clean and consistent input for classification and hierarchical predictions:

1. **Text Normalization:**
    a. Lowercasing: Converted all text to lowercase to ensure uniformity, as case-sensitive words could otherwise result in redundant tokens.
    b. Removing Punctuation and Numbers: Removed punctuation using the regular expression re.sub(r"[^\w\s]", "", text) and numbers using re.sub(r"\d+", "", text), focusing solely on alphabetic content.
2. **Stopword Removal:** A combination of standard and custom stopwords was used to eliminate filler words and other uninformative terms. The custom stopwords were tailored for interview transcripts, including conversational fillers, conjunctions, prepositions, pronouns, auxiliary verbs, determiners and other functional words. These stopwords were merged with the NLTK.

3. **Tokenization and Lemmatization:** Tokenized text into individual words for vectorization. Applied WordNet lemmatizer to reduce words to their base forms, minimizing redundancy in feature representation.
4. **Encoding Techniques:** Experimented with foundational models such as TF-IDF and Bag of Words (BoW) to create sparse representations of text for baseline analysis. Utilized advanced transformer-based sentence encodings (e.g., RoBERTa, DeBERTa) to capture contextual and semantic information critical for accurate coding.
5. **Metadata Integration:** Mapped demographic information (e.g., role, theater, location) to corresponding text segments to enable group-level analysis and enhance model interpretability.
6. **Keyword Extraction Using RAKE:** The RAKE (Rapid Automatic Keyword Extraction) algorithm was applied to extract key phrases from preprocessed text. RAKE excludes stopwords and ranks candidate keywords based on co-occurrence patterns, identifying the most significant phrases.
    - The preprocessed text was passed through the RAKE model initialized with the custom stopwords.
    - RAKE ranks candidate keywords based on their co-occurrence scores, identifying the most significant phrases.
    - The top-ranked keywords were combined into a single string, which served as the enriched version of the original text.

Example: To illustrate, the following example demonstrates how the preprocessing and RAKE keyword extraction transform the raw text:

Original Text	Preprocessed Text
"Um, the researcher collaborated internationally to solve healthcare challenges."	"Researcher collaborated internationally healthcare challenges"

In the above example, conversational fillers (*"Um"*) and stop words were removed, and the significant phrases (*"researcher collaborated internationally healthcare challenges"*) were extracted.

4. PREDICTIVE MODELS

1. BASIC MACHINE LEARNING MODELS

1.1 TF-IDF

TF-IDF is a statistical approach that measures the importance of words in a document relative to their frequency across the entire corpus. Using the TfidfVectorizer, terms were assigned weights based on their term frequency (TF) and scaled inversely by their document frequency (IDF). This approach emphasizes rare but meaningful words, reducing the impact of frequently occurring, less informative terms.

Training and Evaluation

The dataset was split into an 80:20 ratio for training and testing, and target labels were numerically encoded. The TF-IDF feature matrix served as input to a Random Forest Classifier, which was trained to predict NLP Codes. The model's performance on the test set was evaluated using metrics like accuracy, precision, recall, and F1-score. The results showed strong performance for frequently occurring NLP Codes but struggled with underrepresented ones, highlighting challenges associated with sparse feature representations.

Performance

The TF-IDF model achieved reasonable accuracy and classification performance for dominant codes. However, it struggled to classify rare codes effectively due to the inherent sparsity of the data. While providing a solid baseline, the model's inability to capture relationships between terms limited its effectiveness for nuanced predictions.

Limitations

The TF-IDF approach has several drawbacks. Its feature representations are sparse and high-dimensional, which increases computational costs and memory usage for large datasets. Furthermore, TF-IDF does not account for word order or semantic relationships, resulting in a lack of contextual understanding. These limitations restrict its applicability to complex qualitative coding tasks involving hierarchical relationships.

1.2 BOW

Bag of Words (BoW) represents text by creating a sparse matrix of unique words in the corpus, ignoring grammar and word order. The CountVectorizer was used to construct a vocabulary of the top 5,000 most frequent terms, and each document was transformed into a vector indicating the presence or absence of these terms. This method provides a simple yet interpretable representation of textual data.

Training and Evaluation

The same 80:20 data split and label encoding approach used for TF-IDF was applied here. The BoW feature matrix was input to a Random Forest Classifier for training. Performance metrics such as accuracy and classification reports highlighted the model's ability to classify dominant codes effectively, though it struggled with less frequent ones. BoW provided results comparable to TF-IDF but with similar limitations.

Performance

The BoW model achieved competitive accuracy and demonstrated reliability for well-represented codes. However, like TF-IDF, it showed diminished performance for rare codes due to the sparse nature of the data. The lack of contextual understanding hindered its ability to handle more nuanced classification scenarios.

Limitations

BoW has significant limitations. Its sparse, high-dimensional feature representations are computationally expensive and do not scale well with increasing vocabulary size or dataset volume. Additionally, the model ignores word order and semantic nuances, treating phrases with similar meanings as unrelated. These shortcomings make BoW unsuitable for tasks requiring deeper semantic understanding or hierarchical predictions.

2. TRANSFORMER BASED MODELS

We then implemented transformer-based models as the next step in advancing our predictive framework. Building upon the foundational techniques of TF-IDF and Bag of Words (BoW), the aim was to leverage state-of-the-art transformer architectures such as RoBERTa and DeBERTa. These models are pre-trained on massive text corpora and are renowned for their ability to generate contextual embeddings, making them suitable for complex natural language processing tasks.

Model Architecture

The model architecture is straightforward, leveraging the pre-trained RoBERTa and DeBERTa backbones for feature extraction. The embeddings generated by these transformers are passed through a single linear classifier to predict the NLP codes. This approach was designed as an initial experiment to evaluate the feasibility of applying transformer-based models to qualitative coding tasks.

The model architecture included:

- Transformer Backbone: RoBERTa and DeBERTa served as the backbone models, which were fine-tuned to suit the specific needs of this project.
- Single-Head Classifier: The embeddings generated by the transformer were passed through a fully connected linear layer, which served as a classifier for NLP codes.

Training and Evaluation

- Input Data: Preprocessed interview text data was tokenized using the respective tokenizers for RoBERTa and DeBERTa.
- NLP Code Prediction: The embeddings were fed into a fully connected layer, which acted as the classifier to predict the NLP codes.

- Loss Calculation and Optimization: CrossEntropy Loss was used to measure the performance of the predictions against the true labels. The model weights were updated using the Adam optimizer during training.

Performance

The transformer-based models achieved an accuracy of 52% for predicting NLP codes. This outcome indicated that while the approach effectively utilized the contextual capabilities of the transformer architecture, there was still room for improvement in handling the hierarchical nature of the coding tasks.

Limitations

1. Focusing on NLP Codes Only: The current implementation was restricted to predicting NLP codes, overlooking the hierarchical relationship between NLP codes and conceptual codes. This omission limited the scope of the model in addressing the broader context of the coding process.
2. Sequential Context Dependence: Despite their contextual embedding capabilities, transformer models could still misinterpret sequential dependencies within a transcript, especially in dialogues where meaning builds progressively across multiple turns.

3. TRANSFER LEARNING

To address the limitations of the initial transformer-based models, we implemented a transfer learning approach(see *fig 3.1)* by customizing RoBERTa and DeBERTa based architectures to predict both NLP and conceptual codes from textual interview data. This hierarchical prediction framework was designed to align with the two-tier coding process required for the analysis.

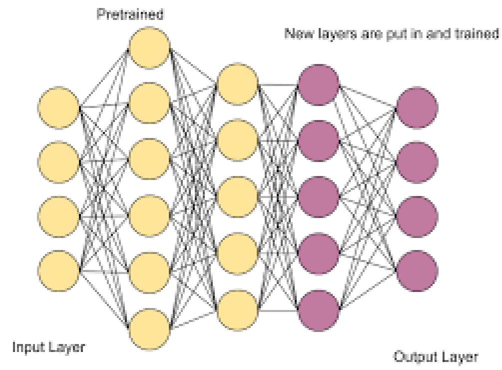


Fig 3.1 Transfer learning architecture

Model Architecture

The model architecture is designed to function in two hierarchical stages:

1. Stage 1: Predicting NLP codes from the input text.
2. Stage 2: Using the predicted NLP codes as intermediate outputs to further classify them into their corresponding conceptual codes.

Model Components

The model comprises three main components:

1. Pre - trained transformer:

   The RoBERTa or DeBERTa model is used for feature extraction. These models are pretrained on massive corpora, providing rich contextual embeddings of input text data.

2. Dual-Head Classifier:

   To perform the hierarchical predictions:

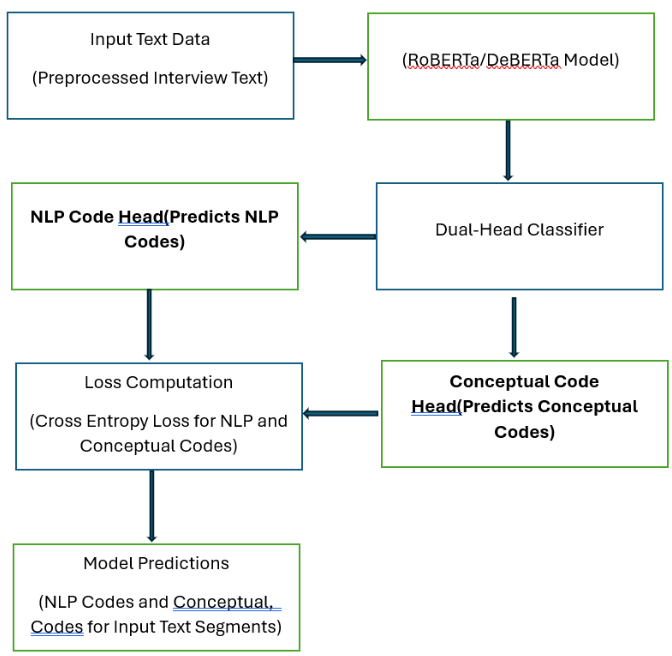
   - *NLP Code Classifier*: A linear layer maps the embeddings produced by the transformer backbone to NLP code predictions.

- *Conceptual Code Classifier*: Another linear layer maps the embeddings to conceptual code predictions, treating the predicted NLP codes as an intermediate representation.

3. Loss Function:

   The loss is computed as the sum of the CrossEntropy Loss for the NLP codes and conceptual codes. This ensures that both predictions are optimized during training.

Model Workflow



Training and Evaluation Process

The model is trained using the following process:

1. *Input*: Preprocessed text data is tokenized and passed through the transformer backbone (RoBERTa/DeBERTa), which generates contextual embeddings.

2. *NLP Code Prediction*: The embeddings are passed through the first classifier (NLP head), which predicts the NLP codes.
3. *Conceptual Code Prediction*: Simultaneously, the embeddings are passed through the second classifier (Conceptual head), which predicts the conceptual codes.
4. *Loss Calculation*: The loss is calculated as the sum of the CrossEntropy loss for both predictions:

   Loss=LossNLP+LossConceptualLoss = Loss_{NLP} + Loss_{Conceptual}Loss=LossNLP+LossConceptual

   This ensures the model optimizes both hierarchical outputs.
5. *Optimization*: The optimizer updates the model weights to minimize the combined loss.

Limitations

While the transfer learning approach brought significant advancements, it also faced certain challenges:

1. Low Volume of Conceptual Code Data: The limited availability of labeled conceptual code data reduced the model's ability to generalize effectively. This data scarcity led to lower prediction accuracy for conceptual codes.
2. Increased Model Complexity: Adding the conceptual code prediction layer increased the model's complexity, leading to a higher risk of overfitting, especially given the small dataset size.

5. RESULTS AND DISCUSSION

Predictive Model	Accuracy	Precision	F1

TF-Idf	0.333	0.34	0.32
BOW	0.309	0.32	0.30
Transformer Models(RoBerta)	0.5125	0.4905	0.469
Transfer learning(RoBerta (Conceptual)	0.897	0.839	0.867
Transfer learning(RoBerta (NLP)	0.475	0.490	0.469

The results of this project reflect both the strengths and limitations of different modeling approaches in the context of conceptual coding. The TF-IDF and Bag of Words models provided a baseline for text representation but lacked the sophistication to capture contextual nuances, resulting in limited performance for predicting NLP codes. Transformer-based models like RoBERTa and DeBERTa offered significant improvements by generating contextual embeddings, achieving a moderate accuracy of 51.25%. However, these models were limited to NLP codes and did not include Conceptual codes, missing the hierarchical structure inherent in the data. To address this, the Transfer Learning approach introduced a dual-classification framework, predicting both NLP and Conceptual codes. While this approach expanded the scope, it faced challenges due to the limited and imbalanced dataset for Conceptual codes, leading to overfitting and reduced accuracy . Additionally, the inherent ambiguity in transcript data, where overlapping themes and lengthy sentences are common, further complicated the labeling process..

6. FUTURE WORK

This project provides a foundational framework for automating qualitative coding, yet it highlights several opportunities for improvement and future exploration. Expanding the dataset is critical to address the overfitting observed in the transfer learning model, particularly for Conceptual codes. Future work should also focus on enhanced preprocessing techniques, such as splitting long sentences into smaller segments and assigning corresponding codes, to reduce ambiguity in the transcript data. Additionally, incorporating more advanced transformer architectures, such as GPT-based models or other fine-tuned BERT variants, could improve the accuracy and robustness of predictions. Introducing regularization techniques, data augmentation, and hierarchical prediction refinements, such as multi-stage models with separate layers for refining NLP and Conceptual codes, may further enhance performance. Moreover, integrating semantic relationships between NLP and Conceptual codes could improve contextual understanding and prediction coherence. Ultimately, this project sets the stage for developing scalable, domain-specific solutions for qualitative coding that can be extended to other fields, including healthcare, education, and social sciences.

7. REFERENCES

Leeson, W., Resnick, A., Alexander, D., & Rovers, J. (2019). Natural Language Processing (NLP) in Qualitative Public Health Research: A Proof of Concept Study. International Journal of Qualitative Methods, 18. https://doi.org/10.1177/1609406919887021

Nelson, L. K., Burk, D., Knudsen, M., & McCall, L. (2021). The Future of Coding: A Comparison of Hand-Coding and Three Types of Computer-Assisted Text Analysis Methods. Sociological Methods & Research, 50(1), 202-237. https://doi.org/10.1177/0049124118769114

Guetterman TC, Chang T, DeJonckheere M, Basu T, Scruggs E, Vydiswaran VGV. Augmenting Qualitative Text Analysis with Natural Language Processing: Methodological Study. J Med Internet Res. 2018 Jun 29;20(6):e231. doi: 10.2196/jmir.9702. PMID: 29959110; PMCID: PMC6045788.

Abram, M. D., Mancini, K. T., & Parker, R. D. (2020). Methods to Integrate Natural Language Processing Into Qualitative Research. International Journal of Qualitative Methods, 19. https://doi.org/10.1177/1609406920984608

Towler L, Bondaronek P, Papakonstantinou T, Amlôt R, Chadborn T, Ainsworth B, Yardley L. Applying machine-learning to rapidly analyze large qualitative text datasets to inform the COVID-19 pandemic response: comparing human and machine-assisted topic analysis techniques. Front Public Health. 2023 Oct 31;11:1268223. doi: 10.3389/fpubh.2023.1268223. PMID: 38026376; PMCID: PMC10644111.

Crowston, K., Allen, E. E., & Heckman, R. (2011). Using natural language processing technology for qualitative data analysis. *International Journal of Social Research Methodology*, *15*(6), 523–543. https://doi.org/10.1080/13645579.2011.625764