Leveraging Transformer Models and Elasticsearch to Help Prevent and Manage Diabetes through EFT Cues

Aditya Ashishkumar Shah

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science and Applications

Edward A. Fox, Chair

Ismini Lourentzou

Dawei Zhou

May 4, 2023

Blacksburg, Virginia

Keywords: Natural Language Processing, Deep Learning, Elasticsearch, Language models, Diabetes.

Leveraging Transformer Models and Elasticsearch to Help Prevent and Manage Diabetes through EFT Cues

Aditya Ashishkumar Shah

ABSTRACT

Diabetes in humans is a long-term (chronic) illness that affects how our body converts food into energy. Approximately one in ten individuals residing in the United States is affected with diabetes and more than 90% of those have type 2 diabetes (T2D). Human bodies fail to produce insulin in type 1 diabetes, causing you to take insulin for survival. However, with type 2 diabetes, the body can't use insulin well. A proven way to manage diabetes is through a positive mindset and a healthy lifestyle. Several studies have been conducted at Virginia Tech and the University of Buffalo on discovering different helpful characteristics in a person's day-to-day life, which relate to important events. They consider Episodic Future Thinking (EFT), where participants identify several events/actions that might occur at multiple future time frames (1 month to 10 years) in text-based descriptions (cues). This research aims to detect content characteristics from these EFT cues. However, class imbalance often presents a challenging issue when dealing with such domain-specific data. To mitigate this issue, this research employs Elasticsearch to address data imbalance and enhance the machine learning (ML) pipeline for improved accuracy of predictions. By leveraging Elasticsearch and transformer models, this study constructs classifiers and regression models, which can be utilized to identify various content characteristics from the cues. To the best of our knowledge, this work represents the first such attempt to employ natural language processing (NLP) techniques to analyze EFT cues and establish a correlation between those characteristics and their impacts on decision-making and health outcomes.

Leveraging Transformer Models and Elasticsearch to Help Prevent and Manage Diabetes through EFT Cues

Aditya Ashishkumar Shah

GENERAL AUDIENCE ABSTRACT

Diabetes is a serious and long-term illness that impacts how the body converts food into energy. It affects around one in ten individuals residing in the United States, and over 90% of these individuals have type 2 diabetes (T2D). While a positive attitude and healthy lifestyle can help with management of diabetes, it is unclear exactly which mental attitudes most affect health outcomes. To gain a better understanding of this relationship, researchers from Virginia Tech and the University of Buffalo conducted multiple studies on Episodic Future Thinking (EFT), where participants identify several events or actions that could take place in the future. This research uses natural language processing (NLP) to analyze the descriptions of these events (cues) and identify different characteristics that relate to a person's day-to-day life. With the help of Elasticsearch and transformer models, this work handles the data imbalance and improves the model predictions for different categories within cues. Overall, this research has the potential to provide valuable insights that can impact their diabetes risk, potentially leading to better management and prevention strategies and treatments.

Dedicated to my beloved parents Ashish and Jyoti Shah, my late grandparents Kanaiyalal and Padma Shah, and my darling sister Yashvi Ashish Shah.

Acknowledgments

I would like to express my profound gratitude to my advisor and mentor, Dr. Edward Fox, for his unwavering guidance, support, and tireless efforts. His wise words and timely suggestions have been instrumental in shaping my research and paving the way for the successful completion of my Master's degree. I would also like to thank Dr. Lourentzou and Dr. Zhou for their valuable suggestions that helped in the completion of this study. I would like to acknowledge Sareh Ahmadi for her invaluable suggestions and assistance throughout this research.

Furthermore, I would like to acknowledge the Computer Science Department at Virginia Tech for providing me with the opportunity to pursue and fund my Master's degree. I am especially thankful to Trey Mayo and Sharon for their outstanding administrative support, which has made the entire process seamless. I am grateful to Dr. Jeffrey S. Stein and his team for their invaluable suggestions and assistance throughout this research. Lastly, I would like to express my gratitude to all the members of the Digital Library Research Laboratory for their support and insightful suggestions. Their contributions have been invaluable in shaping my research and making it a success.

I am grateful for the support of NIH NIDDK 3R01DK129567-02S1 that funded my research assistantship on this project.

Contents

List of Figures

List of Tables

List of Abbreviations

DD Delay Discounting

EFT Episodic Future Thinking

mturk Amazon Mechanical Turk

NLG Natural Language Generation

NLP Natural Language Processing

T2D Type 2 Diabetes

Chapter 1

Introduction

1.1 Motivation

Diabetes is a long-term (chronic) illness that affects the body and its ability to convert food into energy. As per the CDC [1], about 1 in 10 people in the US have diabetes, and more than 90% of those have type 2 diabetes (T2D), which cannot be simply cured with insulin. A proven way to prevent or manage diabetes is through a positive mindset, and learning how to make long-lasting healthy lifestyle changes. Finding out how to manage stress, staying motivated, and having a lifestyle coach are some manageable steps recommended by the CDC [2].

Some research has been done [3, 4, 5, 6] on using machine learning and natural language processing (NLP) techniques to prevent and improve diabetes care using medical data. However, not much work has applied these methods to studying texts about a person's lifestyle and identifying different characteristics and events that might help with managing diabetes. If we can leverage artificial intelligence (AI) techniques and utilize their linguistic understanding for getting insights from personal narratives, that might provide crucial information for medical experts. This work conducts research on building intelligent models which can identify different characteristics in texts about a person's day-to-day life and life events. Our participants identify several events / actions that might occur at multiple future time frames (1 month to 5 years) in the form of text-based descriptions (cues). We can then use different

state-of-the-art NLP and machine learning (ML) models to discover major categories within these cues which will be helpful for medical experts to understand behaviors relevant to T2D, and accelerate discovery in T2D management and prevention.

1.2 Problem Statement

The primary problem in this research is how to build classifiers that can identify categories that correlate to diabetes management, and help medical experts understand their effect on decision-making and health behaviors.

Researchers from Virginia Tech and the University of Buffalo have collected data where participants generate vivid episodic textual descriptions of these events or cues by completing an experimenter-guided interview or self-administered survey task. These cues show broad heterogeneity in content characteristics, which range across categories, including for recreational events, personal milestones, goal orientation, narrative connectivity, event vividness, and episodicity. Ultimately, about 30% of these cues will be labeled by students or annotators from Amazon Mechanical Turk by using a rubric and a slider bar (between 0 to 100) to rate the probability that each text corresponds to each of the categories. Part of our research is to help construct a good sample of the full dataset that will lead to good classifiers.

Once the labeled set of cues is obtained from annotators, the goal is to build high-quality classifiers that can recognize content characteristics. This work includes experimentation to determine the best suitable pretrained models [7] and further fine-tune them (i.e., domain adaption) using our corpus. This will allow us to make efficient use of training data and address some challenges like:

- imbalanced / skewed probability distribution in training data,

- identification of varied content characteristics within EFT cues, across binary and continuous settings, and

- assessment of linguistic characteristics of both individual and sets of cues.

1.3 Hypotheses

The primary hypothesis in this research is that content characteristics of cues play an important role in the management of diabetes. State-of-the-art NLP models can help classify cue texts, reducing the costs of human annotation. These advanced models will help to identify challenging categories from the cues like "better", future orientation, recreation, health, personal, etc. Another challenge that is often faced with domain-specific tasks is data imbalance and scarcity. This work aims to leverage Elasticsearch [8] to retrieve sets of cue texts likely to be rated positive when reviewed by annotators.

The hypotheses of this work are as follows.

H1: Using Elasticsearch applied to our semi-automatically constructed queries devised for the topical categories will identify sets of cue texts that annotators will confirm have at least .8 precision.

H2: A greedy ensemble approach using Elasticsearch and semantic search is more reliable in recommending a good set of cues to be annotated, as compared to simply relying on a single search method.

H3: Resolving class imbalance through the above-discussed method can result in greater F1 scores for challenging categories within EFT cues.

H4: Our transformer models are able to learn from our EFT cue training data and generate high F1 scores for our categories despite the data imbalance.

H5: Our trained high-quality classifiers will yield F1 scores of at least .9 for our categories, based on a human assessment of a sample of the predictions on remaining cues, thereby saving data annotation cost and time.

1.4 Research Questions

The broader research question for this work is: Can we create robust AI models that can understand important characteristics from personal narratives, and identify different topics or categories from these cues? This can be further decomposed into the following research questions.

- How can we efficiently handle imbalance among different categories within the cues when the data is not mutually exclusive?
- Can the result from search methods help to improve classifier performance for challenging categories by reducing class imbalance?
- Can a transformer model identify categories within EFT cues in the presence of class imbalance?
- How accurately can we design this model so that, once it's trained, it can be effectively used to label new cues from many different participants?

1.5 Thesis Outline

- Chapter 1 discusses the motivation, problem statement, hypotheses, and major research questions.
- Chapter 2 summarizes relevant work with a literature review in the area of classification, as well as NLP use in diabetes and personal narratives.
- Chapter 3 gives a background on how content characteristics are defined for binary and continuous categories.
- Chapter 4 introduces EFT data and explains the data collection and annotation process.
- Chapter 5 explains how Elasticsearch can be used to handle class imbalance for EFT cues.
- Chapter 6 and Chapter 7 introduce modeling strategies for identifying content characteristics from cues. The approach, experiment details, and results are presented in these chapters.
- Chapter 8 summarizes the results and important insights from this research. It also discusses some potential ideas for future work.

Chapter 2

Literature Review

2.1 Classification Techniques

Text classification and document categorization are essential as they help readers to understand the broader domain or the context which a collection of words represent. This section briefly discusses various classification techniques.

In recent times, we have seen rapid improvement in analyzing texts and documents. This includes various machine learning and deep learning techniques to classify texts. Unlike numerical data, text data first needs to be preprocessed while taking into account various linguistic characteristics before we can perform any sort of classification. Traditional methods focused on using various techniques to preprocess the text before using any machine learning classifier. In recent deep learning-based techniques, a neural network model makes classification predictions after it extracts and learns important textual features.

2.1.1 Traditional Methods

Traditional methods focused on extracting numeric features from raw texts, which are then fed into a machine learning or other probabilistic modeling algorithm to perform the prediction. One of the first attempts to classify text was made using Naive Bayes (NB) [9], where the authors used a probabilistic model for automatic indexing. Later, various ma-

chine learning-based approaches were proposed like KNN [10], where the authors used the nearest neighbor decision rule to assign a label to an unclassified sample point, or SVM [11], where the authors used Support Vector Machines [12] for learning text classifiers.

Given a piece of raw text, often the first step in a traditional classification technique is to preprocess that raw text [13, 14, 15]. This includes various steps like splitting a text into a set of tokens, cleaning these tokens to remove any stop words, and then using numbers to represent these tokens in a space [16, 17, 18]. One type of numeric representation employs embeddings [19] that aim to capture the meaning of different tokens via a fixed-dimension vector [20].

Bag of words (BOW) [21] is one of the initial methods used to generate features from text. Given a sentence, the BOW model creates a dictionary mapping array where each key is a word and the value of the key represents the count of occurrences of (or other function of) the word in the sentence. The sentence is thus represented in a vector form where each vector element value represents a respective word. However, the BOW model does not take into account the order of words in the sentence and thus misses out on the context. This drawback was tackled in the Word2Vec model [22], where authors used a local contextual window and a neural network model to obtain word representations. They show that fixed-length word vectors can be obtained using two models — CBOW where the current word is predicted using the surrounding words, and Skip-gram which predicts the surrounding word using the current word. Another popular approach is GloVe [23], which uses a local contextual window and constructs a global co-occurrence matrix to obtain word representations. However, the methods discussed so far were not able to handle out-of-vocabulary (OOV) issues, i.e., being able to learn a meaningful representation for a new word that was not observed in training data. Fast-text [24] overcomes this by using a skip-gram model for sub-words and obtaining the word embeddings through a combination of words and characters.

These models helped to convert a piece of text into some meaningful embedding in a latent space which can then be fed to any machine learning model for prediction. Recent traditional methods focused on obtaining a meaningful representation of text in the form of word embeddings and then passing this to a suitable ML model for classification. As the field of machine learning advanced, various ML models were proposed like XG boost [25], which uses tree-boosting through a sparsity-aware algorithm; and LightBGM [26], which used a greedy approach to speed up the training process for gradient boosting, thus making it computationally more efficient. These methods achieved state-of-the-art results on machine learning and predictive modeling tasks [27, 28]. However, the general pipeline for text classification remained the same, where the word embedding models were required to be trained separately to extract text features. This changed when the use of neural networks started gaining popularity for textual data [29, 30].

2.1.2 Deep Learning Based Methods

Deep learning methods use artificial neural networks which try to simulate human brain neurons and learn high-level data features to perform downstream tasks [31, 32, 33]. In the past, various deep learning architectures have been proposed to perform text classification across different domains like sentiment analysis [34], topic modeling [35], relation classification [36], etc.

A Multi-Layer Perceptron (MLP) [37, 38], also known as a vanilla Neural Network [33, 39], was one of the initial methods explored by researchers for classification tasks [40, 41]. The key idea was to extract the embeddings obtained from a model like Word2Vec and pass these to the neural network for training. The main challenge with this approach was that it could not handle larger sentences and long-range dependencies between different words [42, 43].

This was first tackled in Recurrent Neural Networks (RNNs) [44], where the authors proposed a new mechanism to handle sequential modeling through a recursive processing unit. The general idea is that at any given time stamp $\mathbf{t}$, the input to the model will depend on the hidden state which represents the past knowledge up to time stamp $\mathbf{t-1}$. Thus the hidden state up to the previous time step and current input are used to calculate the new hidden state, which will be fed to a future time step $\mathbf{t+1}$. This recursive unit helps the model to handle any variable size of input, which is widely useful in sequence modeling tasks [45]. Over time, many variants of RNNs have been proposed for text classification tasks. Long Short-term Memory (LSTM) [46], one of the most popular RNN architectures, can better handle long-range dependencies by using three memory gates – input, output, and forget gate. LSTM helps to solve the vanishing gradient problem [47] which vanilla RNNs face when the input sequence is quite long. An alternative to LSTM is Gated Recurrent Unit (GRU) [48], which also solves the vanishing gradient problem and handles long-range sequences using just two memory units – update and reset gate – thus being more efficient than LSTM.

Some researchers have also tried convolution neural networks (CNNs) for text classification (TextCNN) [49], where the idea is to train a CNN network on top of embeddings obtained from Word2Vec. C-LSTM [50] is another popular architecture that combines CNN and LSTM for sentiment and question classification tasks. Another similar approach, SeqTextRCNN [51], combined CNN and RNN models for sequential prediction. Both C-LSTM and TextCNN use a fixed pooling window to process the texts, which might not be the best approach for variable-size input. The Dynamic CNN (DCNN) [52] model handles this by using a dynamic k-max pooling operator for subsampling, which helps to explicitly capture short and long-range relations within a sentence. Most of the classification-based methods operate on word-level semantics. However, there are many cases where words at test time do not

appear in training data, leading to the OOV issue. One way to tackle this is to build models which use character-level semantics. Thus, given any sentence, the features are obtained for characters instead of words. CharCNN [53] involved an empirical study on character-level CNN for text classification, where experiments across different benchmarks showed that CharCNN can be an effective method. Kim et al. [54] used a character-aware network where inputs are characters, but predictions are made at the word level. There are other interesting CNN-based approaches like tree-based CNN architecture [55], text matching as recognition [56], and using CNN for medical text classification [57, 58].

RNNs and LSTMs process texts sequentially over time, while CNNs process texts over a spatial neighborhood. Although they perform well for text classification, they do not consider any contextual information from all the surrounding words in a sentence. If we have a long sentence with a few words of a sentence having important contextual information, then a model like LSTM or CNN might not be able to generate accurate representation as it would not know which words are most important in a sentence. That's where the attention mechanism helps; this was originally proposed by Bahdanau et al. [59] for neural machine translation. The authors showed that in the encoder-decoder architecture, if the decoder can focus on the appropriate words during computation, then it can generate more accurate translations. Attention was first used for classification by Yang et al. [60], where they applied two levels of attention mechanism through a hierarchical attention network. The model showed promising results for document classification along with visualizations to interpret the model output. Inspired by this, various models were proposed which used some form of attention for classification. Direction self-attention (DiSAN) [61] uses a novel directional attention mechanism for sentence encoding without the need for any CNN or RNN architecture. Liu et al. [62] proposed an inner attention model on top of BiLSTM for inference tasks. Wang et al. [63] introduced a joint label-text embedding model for classification tasks

where they embed the text and labels in the same latent space. Other attention-based approaches have performed well for classification tasks like self-attentive sentence embedding [64], which uses interpretable sentence embeddings; multi-scale feature attention [65], which uses a dense CNN model with multi-scale attention; and neural attentive bag-of-entities [66], which performs text classification via attention obtained from the entities. Attention-based methods have proven to show state-of-the-art results on various classification benchmarks [67, 68].

Most of these attention-based approaches were used on either LSTM, CNN, or some form of base encoder-decoder model. Vaswani et al. [69] introduced a new Transformer architecture that was based solely on the attention mechanism without the need for any recurrence or convolution units. Since the inputs in the Transformer model were not processed sequentially, it achieved a high level of parallelism with significantly lower training time. Inspired by this, Devlin et al. [70] proposed BERT, a deep bidirectional transformer-based encoder model for language representation. They introduced a new unsupervised strategy called masked language modeling (MLM) for training the model on the massive open-source text, and achieved state-of-the-art results on several natural language processing tasks. BERT showed that since transformers are computationally cost-effective, it is possible to build very large models and train them on massive data sets with GPUs. Since then we have seen numerous Transformer-based Pre-trained Language Models aka Large Language Models (LLMs). The RoBERTa [71] model is a more optimized version of BERT which is trained on larger datasets and for more epochs. Lan et al. [72] proposed ALBERT, a lighter version of BERT which uses less memory and training resources. DistilBERT [73] uses knowledge distillation to reduce the model size while still being as competitive as the BERT model. XLNET [74] uses an autoregressive pretraining method and outperforms BERT on different NLP benchmarks. Raffel et al. [75] proposed a T5 model which considers every NLP problem as a text-to-text

transformation, and achieves state-of-the-art results on several benchmarks, including text classification.

The majority of these transformer-based models are largely open-sourced[1]. The general framework for a classification pipeline is to use a pretrained transformer model and fine-tune it on the required downstream task. While these models have shown state-of-the-art results on classification benchmarks, they haven't been explored for personal narratives which often contain challenging topics/categories. This work aims to leverage the knowledge of these models and build classifiers to identify different topics in personal narratives obtained from patients. Based on the extracted topics, medical experts will be able to consider focusing on characteristics that appear to correlate with improved health outcomes.

2.2 NLP in Healthcare

AI and NLP have been widely used in medical applications to detect and prevent diseases [76, 77, 78, 79]. These methods use some form of biomedical data like CT scan images [80, 81], electronic health records [82, 83], clinical trials [84], health surveys [85], etc. However, some diseases like type 2 diabetes (T2D) require monitoring that goes beyond the medical data, and prevention/management of such diseases depends on day-to-day activities, lifestyle changes, etc.

2.2.1 NLP for Diabetes

Several works have been proposed that use machine learning to detect and prevent diabetes. Zheng et al. [86] use an NLP based system to identify diabetes mellitus in patients from

[1] https://huggingface.co/models

electronic medical records. Using unsupervised feature extraction, they were able to discover important clinical variables that correspond to diabetes in patients. Jin et al. [87] developed RNN and CNN-based deep learning models to detect hypoglycemic events from the health records of diabetic patients. Mishra et al. [88] used discharge summaries of patients to extract relevant topics which are correlated with diabetes. Their algorithms also helped to identify high-risk factors and assess protocol compliance from the patient records. In [89], the authors used a ResNet model to identify the presence of type 2 diabetes from retinal fundus images. There are many other studies on using machine learning and deep learning for type 2 diabetes. Fregoso-Aparicio1 et al. [4] conduct an extensive review of different predictive models and strategies used for the prevention of type 2 diabetes. Based on the findings, the authors claim that deep learning models were most efficient for preventing T2D when they are used on some form of lifestyle data. Wu et al. [90] did a study to understand several reasons to discontinue insulin in T2D patients, and had similar findings wrt the effect of lifestyle on T2D.

2.2.2 Personal Narratives for Healthcare

Lifestyle behaviors, mental health, and diet can have a significant impact on illness. Few works aim to capture important insights from personal narratives which describe participants' lifestyles and day-to-day activities. In [91], the authors use NLP to identify mental health disorders from narratives written by female patients. The study showed that AI has the potential to identify important features which may correlate with PTSD in women after childbirth. Vandenbussche et al. [92] did a similar work where they use ML and NLP-based classifiers to obtain relevant topics from narratives by migraine patients. They show that narratives often contain important context for any illness like migraine, and that ML systems can achieve good performance in identifying such information. Genugten et al. [93] proposed

an approach to automatically score autobiographical interview responses with natural language processing to reduce scoring burden and enable larger studies. In [94], Fischer et al. showed that NLP can be effective to identify depression among diabetic patients using free-form text like office notes, which often contain narratives about a person's day. Koleck et al. [95] used NLP techniques to extract a wide range of symptom information from free-text narratives of health records. This symptom information can be more useful and speed up the manual verification process of symptoms for disease detection. Work done by Sultana et al. [96] shows the importance of narrative datasets and how NLP methods can be applied to human stories for extracting important knowledge. While much work has been done on using personal narratives for different illnesses, there's more to explore, particularly for conditions that cannot be cured by medications, like T2D. One of the relevant works in this direction is by Vidyadharan et al. [97] where they used NLP and deep learning techniques to conduct an evidence-based study of diabetes prevention. The paper used a data corpus with a question-answer format, on different topics like food habits, diet, risk factors, etc. Then they use LSTM and CNN models to identify relevant concepts associated with diabetes.

Most of the research on diabetes uses AI techniques on some form of medical or electronic health records. However, as studies suggest, it is important to understand a person's lifestyle habits, stress, anxiety, etc. for diseases like T2D. There has been limited research specifically focused on the emerging topic of Emotion-Focused Therapy (also called EFT) that utilizes text-based event descriptions and personal narratives to identify relevant topics that may correlate with T2D. To the best of our knowledge, this work represents the first such attempt to employ NLP techniques to characterize Episodic Future Thinking cues and investigate correlations with decision-making and health outcomes.

Chapter 3

Categorization and Content Characteristics

3.1 Background

Episodic Future Thinking (EFT) can be thought of as a scalable intervention to reduce Delay Discounting (DD) and improve health-related behaviors [98, 99]. Here, participants identify several events that may occur at multiple future time frames (e.g., 1 month to 10 years) and generate text-based event descriptions or cues that can prompt the EFT. They generate vivid episodic descriptions of these events or cues, by completing an experimenter-guided interview or self-administered survey task. In this generalized form of EFTs, cues show heterogeneity in content characteristics. Broad categories in these cues can include:

- *topics* ranging – from everyday recreational activities (e.g., watching a favorite movie) to personal milestones in work, health, etc.,
- *structure* – featuring varying degrees of goal orientation and narrative connectivity across different time frames (e.g., initiating a weight loss program in one month), and
- *imagery* – featuring a variation in event vividness, episodicity, and emotional valence.

This work considers a total of 15 different content categories, some of which have a continuous

rating while others possess a binary rating. Table 3.1 gives a summary of the different categories that can be used to describe the cues.

Table 3.1: EFT Content Characteristics

Content	**Categories**	**Type**
Topics	Alone	Binary
	Better	Binary
	Celebration	Binary
	Family	Binary
	Food	Binary
	Friends	Binary
	Health	Binary
	Partner	Binary
	Pet	Binary
	Recreation	Binary
Structure	Future	Binary
	Narrative	Continuous
Imagery	Emotion	Continuous
	Episodic	Continuous
	Vividness	Continuous

Since these cues are *not* mutually exclusive, a given cue can belong to more than one category. Consider the following example:

```
In about 5 years, I am having a child. Or rather, my fiancee is. My sister
waits with me in the hospital. It's my first child. We're nervous, but
is excited to welcome a new, healthy life into the world.
```

The above cue would belong to the following categories: *health* because it mentions a healthy life being brought to this world; *family* because it mentions about child, sister, and fiancee; *future* because it discusses an event that will take place in the future; and *better* because the event describes a positive feeling that makes the participant's mood better, and is about positive changes in life. Next, let's discuss a brief overview of the definition and corresponding

annotation label for each category.

3.2 Binary Content Characteristics

These content categories have a binary annotation. For a given category, the cue will either be highly related to the category or have no relation at all. Table 3.2 shows all the binary categories that are taken into consideration, and their associated definitions. These definitions are shared with annotators to assist them to label every cue relative to its respective category, as accurately as possible. All of the categories discussed in Table 3.2 possess discreetness in their semantic meaning, i.e., either a topic can be related to health, or it is not related; either an event mentions a family member or it has no mention of any family members. Thus there are very slim chances of these categories exhibiting some form of ambiguity. However, there are certain categories where ambiguity cannot be ignored.

3.3 Continuous Content Characteristics

Imagery categories often consist of far richer semantic information compared to the ones that are discussed before. For a category like vividness, a given cue can be anywhere between extremely detailed with highly articulate context, to bland and simple with no details or strong adjectives or adverbs. Thus, for such categories, a continuous label is preferred where the annotators are given a slider that ranges between 0 - 100, so they can select the appropriate number, based on how informative the text is. Table 3.3 lists all of the continuous categories, with their definitions. As we can see from the definitions in the table, these categories exhibit a high degree of semantic information. For example, if a cue has a high degree of positive emotion, then the annotators would assign it a relatively high score

anywhere between 75 to 100.

A complex category like *narrative* uses a set of cues, where the degree of narrativeness is defined based on the relatedness between the cues within the set. Annotators are provided with 3 distinct labels, namely: no relation, moderately related, and highly related. This helps annotators to understand the degree of difference between cues, and assign them an appropriate score.

Table 3.2: Binary Content Characteristics

Categories	Label	Definitions
Alone	No	Contains no references or mentions of any activities which portray that they are alone or by themselves.
	Yes	Contains an obvious, specific reference to events and activities which shows that they are being done alone or by themselves.
Better	No	Contains no references to "a better me", personal development, or self-improvement.
	Yes	Contains obvious references to "a better me", including personal development, self-improvement, making positive changes in life, etc.
Celebration	No	Contains no references to a celebration or a celebratory event.
	Yes	Contains an obvious, specific reference to a celebration or a celebratory event.
Family	No	Contains no references to any family members.
	Yes	Contains an obvious, specific reference to a family member like mother, father, son, daughter, etc.
Food	No	Contains no references to food, eating, or cooking.
	Yes	Contains obvious or specific references to food, eating, cooking, or a meal. Eating or food is a major and essential component of the text.
Friends	No	Contains no references or mentions of any friends.
	Yes	Contains an obvious, specific reference to friends discussing some event or activities.
Future	No	The writer is imagining themselves in a past event.
	Yes	The writer is completely imagining themselves in a future event. Only future events and activities are mentioned.
Health	No	Contains no references to health; does not discuss physical state, mental health, or intentional changes in behavior and health outcomes.
	Yes	Contains an obvious reference to physical or mental health. Examples may describe mental and physical health, changes in behaviors, etc.
Partner	No	Contains no references or mentions of any romantic partner, e.g., wife, fiancee, or boyfriend.
	Yes	Contains an obvious, specific reference or mentions of romantic partner, e.g., spouse, wife, husband, fiancee, girlfriend, or boyfriend.
Pet	No	Contains no references or mentions of any pets like cats or dogs.
	Yes	Contains an obvious, specific reference to any pet animals that the participant either owns or relates to.
Recreation	No	Contains no references to engaging in an activity for leisure or fun while not working at one's job.
	Yes	Contains obvious or specific references to engaging in an activity for leisure or fun while not working at one's job.

Table 3.3: Continuous Content Characteristics

Categories	**Label**	**Definitions**
Narrative	No Relation	No cues in the set are obviously or specifically related thematically or otherwise connected. Connected texts may tell a story or describe related events.
	Moderately Related	Some, but not all cues in the set, are obviously or specifically related thematically or otherwise connected. Connected texts may tell a story or describe related events.
	Highly Related	The majority or all cues in the set are obviously or specifically related thematically or otherwise connected. Connected texts may tell a story or describe related events.
Vividness	No Relation	The text contains no details about the event. It is difficult to imagine the event. No context has been given regarding the event.
	Moderately Related	The text contains only a few details or mostly non-specific details, making it somewhat hard to imagine the event. More details could have been provided describing the event.
	Highly Related	The text contains sufficient and specific details so that the event described is readily and easily imaginable. A considerable amount of context has been given regarding the event.
Episodicity	No Relation	The writer primarily describes general knowledge of events or occurrences. The event is described as if the writer is not present or personally experiencing the event.
	Moderately Related	The writer describes both personal experiences, events, and actions in addition to general facts or ideas. The writer is somewhat in the moment but also adds in a few facts or ideas.
	Highly Related	The writer primarily describes personal experiences, events, and actions, not general facts or ideas. The writer is describing events as if they are currently experiencing them "at the moment".
Emotion	Negative Emotion	Primarily contains references to negative emotions or behaviors, including sadness, crying, or anger.
	Neutral Emotion	Contains references to both positive (e.g., laughing, smiling) and negative emotions or behaviors (e.g., crying, sadness); or contains weak or ambiguous references to positive or negative behaviors.
	Positive Emotion	Primarily contains references to positive emotions or behaviors including laughing, smiling, and happiness.

Chapter 4

Data

4.1 Background

Successful prevention of diabetes may result when present behavior is guided by future outcomes. Delay Discounting (DD) is a common term used to refer to the tendency of individuals to value immediate rewards more than delayed rewards. In the context of diabetes, delay discounting can play a significant role in how people manage their condition. For example, if a person with diabetes is presented with the option of eating a piece of cake now or eating a healthy meal that will help regulate their blood sugar levels in the long term, they may be more likely to choose the cake due to the immediate pleasure it provides. This can lead to poor management of the condition, as individuals may prioritize short-term gratification over the long-term benefits of managing their blood sugar levels. Research has shown that interventions aimed at reducing delay discounting can improve diabetes self-management behaviors, such as medication adherence, following exercise regimes, and healthy eating habits. This suggests that addressing delay discounting may be an important component of diabetes care.

4.2 Data Collection

Participants in one related trial are mostly 120 urban and underserved patients who have poor control over T2D. This allowed design of a behavioral T2D treatment that bridges the gap between geographic and other sociodemographic health disparities. Most studies on EFT have examined generalized content which reflects a broad range of personally meaningful positive future events.

Our data was originally compiled from 18 studies conducted by the medical research teams at Virginia Tech or the University of Buffalo, that experimentally examined the effects of EFT on T2D and other relevant health behavior outcomes. Our dataset includes a total of 1691 participants who have T2D or are at risk of developing T2D. Participants generate a vivid episodic description of events and related cues by completing an experimenter-guided interview or self-administered survey task. These data sources comprise lab, online, and clinical studies. The lab studies comprised single or multiple sessions of interviews or survey tasks. The online studies comprised a single session with survey panels, while the clinical studies were typically longer, lasting anywhere between a few weeks to months.

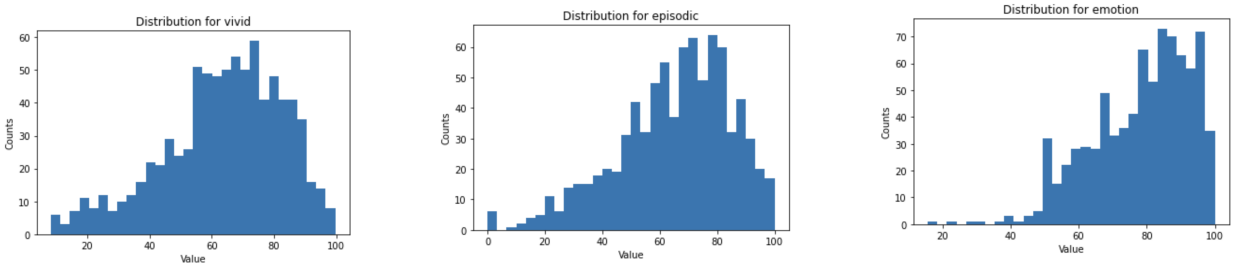
Overall, our dataset comprises an estimated 11,000 cues, each having an approximate length of a few sentences. Out of these, about 8000 cues have been collected so far from different participants and studies. Table 4.1 shows a summary of some of our data sources. The largest sets of cues are obtained from online studies which contain the most participants, followed by lab studies.

Table 4.1: EFT data sources comprising of different studies and participants

Studies	Population	Participants	Cues	Primary outcomes
Lab Studies	Overweight / Obesity	283	1649	DD, dietary intake, food purchasing
	Prediabetes	145	747	DD, dietary intake, food purchasing
Online Studies	Overweight / Obesity	656	2656	DD, food purchasing
	T2D	396	1848	DD
Clinical Studies	Overweight / Obesity	129	485	DD, food purchasing, weight
	Prediabetes	64	448	DD, weight, glycemic control, physical activity
	T2D	18	126	DD, weight, glycemic control, physical activity
	Total	1691	7959	

4.3 Data Annotation

Manual annotation is used to label a training set selected from these cues. The annotators are from Amazon Mechanical Turk (mturk) [100], a crowdsourcing platform where humans can complete intelligent tasks in exchange for monetary compensation. All of the workers involved in this annotation have a Master's qualification, indicating they have demonstrated a high degree of accuracy and success in various intelligent tasks. Initially, assessors will be assigned a subset of these cues for manual annotation, to construct a suitable training set, numbering roughly 3000. They will use a rubric sheet to guide their assessment. A binary rating is required for 11 of the categories listed in Table 3.2. For the first pilot study, we had a total of 10 binary categories, while for the second pilot study a new category "partner" was added. Regarding the 4 continuous categories described in Table 3.3, they will use a slider bar which will range from 0 - 100 to rate the probability that a given cue text belongs to a particular category. Once every cue is rated three times, the final ground truth is calculated



(a) Distribution for Vivid (b) Distribution for Episodic (c) Distribution for Emotion

Figure 4.1: Data distribution of continuous categories from first pilot study

by taking the average from all three ratings. For continuous categories, an average of the three scores is used as the final label. For binary categories, majority voting is performed between the three different ratings for the final label. This will help us to build a gold standard corpus of EFT-based cues.

The data annotation is conducted in sequential rounds where a batch of data is passed at a given time to annotators. Once this data is annotated, the next set of data is passed and this process continues until collectively we have all of the required labeled data, i.e., 3000 cues in our case.

4.3.1 First Pilot Study

In the first pilot study, annotations for about 800 examples are obtained from mturk. Each of these examples is rated at least three times. Table 4.2 shows the data distribution across labels for different binary categories.

As we can see from the table, the annotations possess imbalance, with most categories having more negative than positive samples.

For continuous categories, a single real-valued score is used which ranges between 0 - 100 for all 800 examples. Figure 4.1 shows the data distribution for continuous categories.

Table 4.2: Data distribution of binary categories from first pilot study

Categories	Negative Samples	Positive Samples	Percent Positive
Alone	631	108	13.1
Better	511	228	27.8
Celebration	556	183	22.3
Family	468	271	33.1
Friends	668	71	8.6
Food	503	236	28.8
Future	180	559	68.2
Health	527	212	25.8
Pet	790	29	3.5
Recreation	304	435	53.1

The visualization in Figure 4.1 shows that for each of the categories, a significant number of cues are rated as highly positive or highly related.

4.3.2 Second Pilot Study

In the second pilot study, around 1600 annotated examples had been obtained from mturk. The annotators proceeded as before, but also assessed an eleventh binary category, "partner". Each of the examples is rated at least three times. Table 4.3 shows the data distribution across labels for the different binary categories.

For continuous categories, a single real-valued score is used which ranges between 0 - 100 for all 1600 examples. Figure 4.2 shows the data distribution for continuous categories.

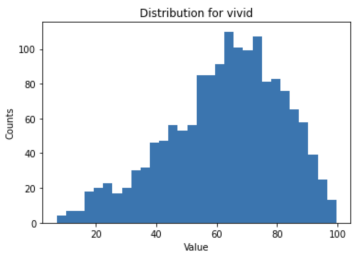
The histogram plot shows that for the each of the categories, a significant number of cues

Table 4.3: Data distribution of binary categories from second pilot study

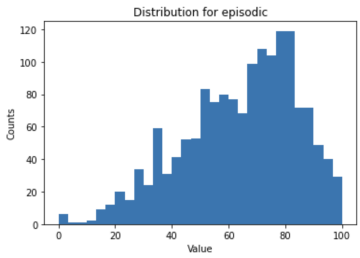
Categories	**Negative Samples**	**Positive Samples**	**Percent Positive**
Alone	1274	280	18.1
Better	1076	478	30.7
Celebration	1161	393	25.2
Family	971	583	37.5
Friends	1312	242	15.6
Food	1045	509	32.7
Future	485	1069	70.5
Health	1126	428	27.5
Partner	1163	391	25.1
Pet	1490	64	4.1
Recreation	629	925	59.5

have been rated as highly positive or highly related.

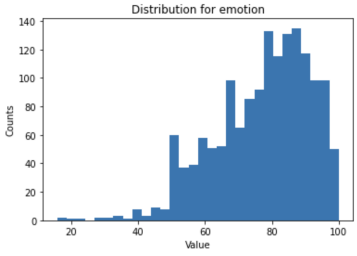
For some binary categories like “Pet”, the data is highly imbalanced. This can be because a very small percentage of participants owned a pet. This makes it less likely for the participants to talk about pets in the cues in the data collection. To help handle the imbalance, this work includes an Elasticsearch-based method which is discussed in the following chapter.



(a) Distribution for Vivid



(b) Distribution for Episodic



(c) Distribution for Emotion

Figure 4.2: Data distribution of continuous categories from second pilot study

Chapter 5

Elasticsearch for Annotation Enhancement

5.1 Background

Elasticsearch [8] is a popular search and analytics engine used for indexing, searching, and analyzing large volumes of data. It provides powerful features for text-based search and retrieval, such as stemming, approximate matching, and relevance ranking. Elasticsearch can be combined with machine learning algorithms to improve search accuracy and relevance. Several frameworks use Elasticsearch to index and retrieve data, and machine learning algorithms to analyze user queries and provide personalized search results [101, 102, 103].

One of the interesting applications of Elasticsearch is for enhancing data annotation. It can be used to enhance supervised machine learning by providing a powerful platform for text indexing, search, and retrieval. When there is less data available for training machine learning models, Elasticsearch can be used to enrich the available data by providing additional context and insights. This can help to monitor the quality of annotations and make sure that they are consistent, for different categories. In the case of a large unlabelled real-world dataset, often we can face class imbalance issues where some common categories are more likely to be recorded by the participants. In the EFT data, there are a few categories with severe class imbalance. So this work aims to use Elasticsearch to identify cues that belong to the

rare categories from the pool of unlabelled cues. These retrieved cues from Elasticsearch can then be passed on to the annotators in mturk. By doing this, we increase the probability of obtaining positively labeled cues for specific categories, thereby resulting in a more balanced training dataset.

As was explained in the previous chapter, and we can see from Tables 4.2 and 4.3, the data samples are imbalanced, particularly regarding positives. To lead to more balanced samples, Elasticsearch is used to identify cues which have a higher probability of belonging to a required category. The retrieved cues can then be given to the mturk annotators to increase the likelihood of instances of the required label being found in training data.

5.2 Queries for Different Categories

To identify potentially useful cues from the pool of unlabelled data, tailored queries are used for every category. These queries are usually keyword-based, and Elasticsearch uses them to find cues that approximately match with these keywords.

To create these queries, we leverage the classifiers trained on the initial and second pilot study data.[1] Specifically, we utilize SHAPley Additive exPlanations (SHAP) [104], a popular method for studying model predictions. SHAP values enable us to visualize and understand the important parts of the input data that the model focuses on while predicting the output label. SHAP values provide both global and local explainability, which helps us understand how the features of the input data are related to the outputs.

By using SHAP on the trained classifiers, we identify the set of keywords that the model focuses on while predicting the cue as positive. For example, Figure 5.1 shows the top words

[1]Classifiers, that will be described in subsequent publications, were built through the research of our project team, led by Dr. Fox and GRA Sareh Ahmadi. See more about the classifiers in Chapters 6 and 7.

generated by SHAP for the "health" category.

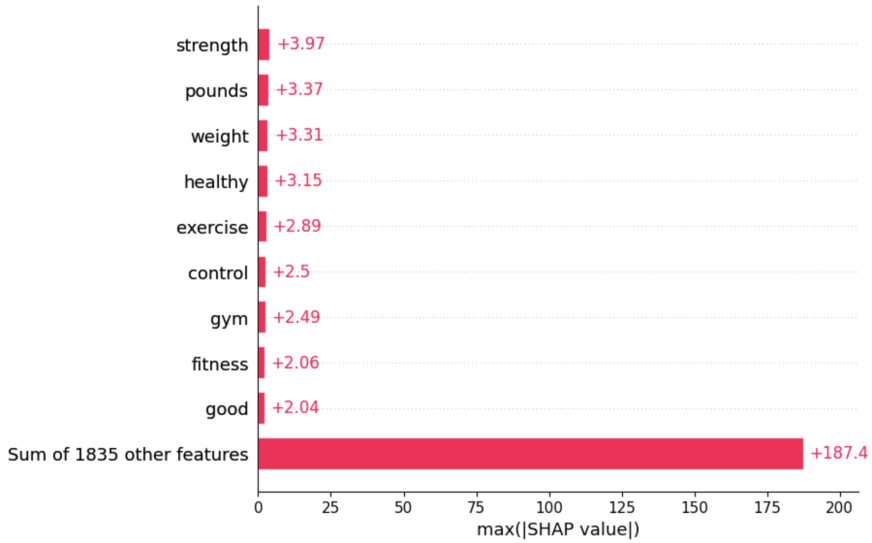


Figure 5.1: Top words obtained using SHAP for classifier trained on "Health" category

We manually verify these words, selecting from them to form a high quality set of keywords, which can be used as a query to retrieve helpful cue sets. We follow this procedure for each of the six most imbalanced categories to obtain a set of cues that can be used to reduce the imbalance in the training dataset, and thus improve the classifiers' performance.

Table 5.1 shows different queries that should be helpful to retrieve cues for the positive class of a category. For example, for a cue to be positive or highly related to the *celebration* category, it generally will have some mention of words that indicate celebration, like "birthday", "party", "celebrate", etc. Elasticsearch thus can help to aid the annotation process to generate a more balanced training dataset.

5.3 Retrieval using Elasticsearch

The queries mentioned in Table 5.1 are used on the entire pool of unlabelled data. These queries help to retrieve example cues for every required category, which could help us con-

Table 5.1: Elasticsearch queries for categories

Categories	Queries
Alone	`alone myself`
Better	`better-me accomplish`
Celebration	`celebrate birthday christmas party`
Family	`family`
Food	`food meal eat cook`
Friends	`friend`
Future	`ago past`
Health	`health weight gym exercise`
Partner	`partner wife husband boyfriend girlfriend`
Pet	`dog cat puppy pet`
Recreation	`hike movie art game trip`

struct a hopefully optimal set of cues for training high quality classifiers. Thus, to get more positive examples for "pet", Elasticsearch can be used to retrieve cues that mention keywords like "dog", "cat", "pet", etc. This set is then passed on for annotation, which should help achieve a better balance in training datasets by identifying cues likely to be rated positive when reviewed by annotators.

When the cues are searched using a query, Elasticsearch will retrieve the top ranked cues, along with a similarity score assigned to each of them. This score estimates how similar the cue is to the given query. Figure 5.2 shows an example of how cues are ranked based on scores when searched using a query.

Categories to be searched are selected based on their imbalance ratio, i.e., the ratio between negative and positive examples in the data. This imbalance ratio is obtained using the data

Figure 5.2: Example of results obtained using the query for the "Partner" category. Names and personal identifiers have been redacted.

distribution (Table 4.3) from the most recent pilot study. Table 5.2 shows the number of examples retrieved using the queries for the top six most imbalanced categories. Due to the highly imbalanced nature of the "Pet" category, we have excluded it from our search retrieval.

Table 5.2: Results from Elasticsearch

Categories	**Imbalance Ratio**	**Cues Retrieved**
Pet	0.042	160
Friends	0.184	500
Alone	0.219	830
Partner	0.336	1300
Celebration	0.338	900
Health	0.380	320

The search is performed on 7000 cues of unlabeled data, i.e., which are yet to be annotated. For every category, the retrieved cues are sorted based on their similarity score and ranked

accordingly.

To increase the likelihood of the cues being rated positive when reviewed by annotators, this work combines the results obtained from Elasticsearch with the research done by Sareh Ahmadi, a Ph.D. student also working with Dr. Fox on this project. She worked on obtaining relevant cues using semantic search with the same queries as discussed in Table 5.1. Semantic search retrieves the top documents by trying to understand the context of the query and then estimating similarity between the contextual representations of the query and documents.

To form the best set of cues for annotators, we combine cues obtained from both the keyword-based method (Elasticsearch) and contextual method (semantic search). We perform a set intersection of the cues retrieved from both search methods and select the top 25 cues from each of the 5 imbalanced categories (as shown in Table 5.2). The resulting hopefully optimal set of 125 cues is passed on to the annotators. By utilizing both search methods, we can retrieve cues that are more likely to be classified as positive by annotators, improving the accuracy of the annotation process.

5.4 Retrieval Enhancement using Combined Search

This work addresses the challenge of using retrieval methods to select a limited set of texts for annotation in text classification, where constraints on human resources exist. An additional challenge addressed is dealing with binary categories that have a small number of positive instances, reflecting severe class imbalance. In our situation, where annotation occurs over a long time period, the selection of texts to be annotated can be made in batches, with previous annotations guiding the choice of the next set.

For this experiment, Elasticsearch is performed on the data obtained from the second pilot

study (Table 4.3), focusing specifically on three categories: "Alone," "Friends", and "Health". The same set of queries are used for performing the search as was given in Table 5.1. This work is combined with the results obtained from semantic search by Sareh Ahmadi. We retrieved data for each category, considering the top 20, 40, 60, and 80 cues as positive examples (label 1) based on their similarity score. To analyse the performance of both of the search methods, we measure the precision obtained using the retrieved data, i.e., P@K, given that we have the true label for each retrieved example. The results are shown in Table 5.3.

Table 5.3: Precision of top K samples for search methods

Category	**Data Samples**	**Elasticsearch - P@K**	**Semantic search - P@K**
Alone	Top 20	**0.9**	0.8
	Top 40	**0.86**	0.6
	Top 60	**0.85**	0.51
	Top 80	**0.85**	0.47
Friends	Top 20	0.85	**1.0**
	Top 40	0.86	**0.92**
	Top 60	**0.92**	0.8
	Top 80	**0.89**	0.7
Health	Top 20	1.0	**1.0**
	Top 40	0.98	**1.0**
	Top 60	0.97	**1.0**
	Top 80	0.98	**1.0**

For the semantic search, the best performance comes from the health category with a precision of 1 for all the top k cues, i.e., given the query, everything it returns belongs to the health category (true label). The performance for the friend category for the top 60 cues is 80%; for the alone category the top 40 cues have precision 60%. For Elasticsearch, similar

to semantic search, the health category has the best search results. But the performance of semantic search is better for this category. However, the search result for the alone category in Elasticsearch is much better compared with semantic search, with a precision of 85% for the top 80 search results. For the friend category, semantic search surpasses Elasticsearch for the top 40 retrieved data, but after that, Elasticsearch has better performance.

In order to gain a more comprehensive understanding of the performance of Elasticsearch, we calculated precision and recall scores using ground truth data provided by annotators. The precision and recall scores were computed across all of the data, and the results are summarized in Table 5.4. This analysis provides a more detailed assessment of the effectiveness of Elasticsearch across the entire dataset.

Table 5.4: Precision and Recall for Elasticsearch

Category	**Data Samples**	**Elasticsearch**	
		Precision	**Recall**
Alone	Top 20	0.82	0.19
	Top 40	0.76	0.31
	Top 60	0.74	0.55
	Top 80	0.74	0.72
Friends	Top 20	0.84	0.17
	Top 40	0.82	0.34
	Top 60	0.81	0.56
	Top 80	0.82	0.77
Health	Top 20	0.95	0.19
	Top 40	0.93	0.37
	Top 60	0.92	0.56
	Top 80	0.92	0.78

Table 5.3 shows that Elasticsearch and semantic search complement each other; thus using

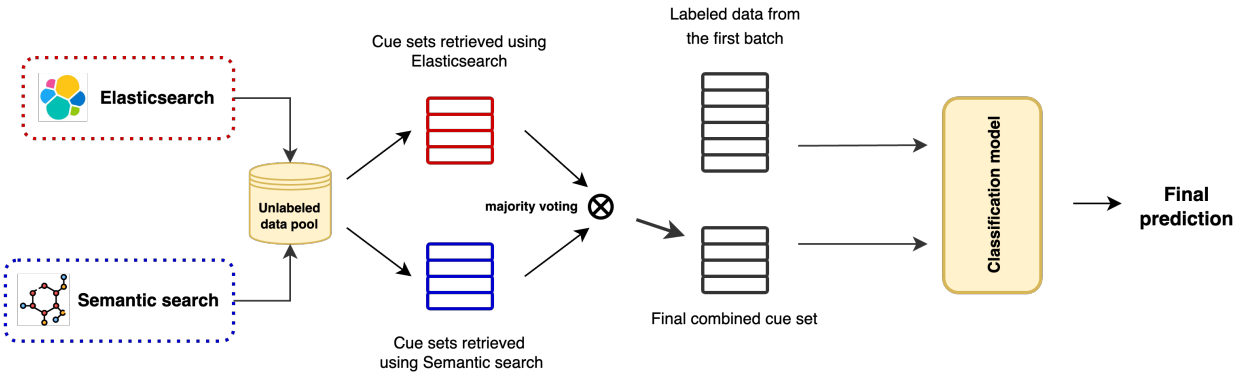


Figure 5.3: Overview of the combined search framework. The resulting cue sets obtained from both methods are combined with the previously labeled data and used for modeling.

the retrieved data combined from both the methods is likely to yield better results than the data from either. For this reason, we consider the majority vote for each retrieval model to ensure that the retrieved data is present in both models. The result is likely to be a positive sample for that specific category. An overview of the combined search method is depicted in Figure 5.3.

This will help us to construct a set of cues for training high-quality classifiers. This set is then passed on for annotation, increasing the likelihood of them being labeled as positive when reviewed by annotators and thereby achieving a better balance in training datasets. After receiving the new batch of the data with newly identified positive examples, we trained the classifiers on the new data and measured the F1-score (discussed in Section 6.4.3).

Chapter 6

Modelling for Binary Content Characteristics

6.1 Approach

Our dataset consists of cues which are textual descriptions of events or plans. This work aims to leverage knowledge from transformer models by fine-tuning them on the EFT dataset to identify various categories within a cue. For binary content characteristics, the model is trained to predict whether a given cue belongs to a respective category.

Consider an input text cue, $X = [x_1, ..., x_n]$, where x_i is the $i-$th token in the text and n is the length of the sequence. This sequence of tokens is passed as an input and uses the same tokenizer library as that used by the respective transformer model [69, 70]. Since this is a classification task, we obtain the $[CLS]$ (classification head) output from the transformer model. Let $C \in R^d$ be the output from the classification head with hidden dimension size d. This output is passed to feed-forward networks followed by a softmax layer for final prediction [38]. Fig. 6.1 shows the architecture of our pipeline. Separate classifiers are trained for each of our binary categories.

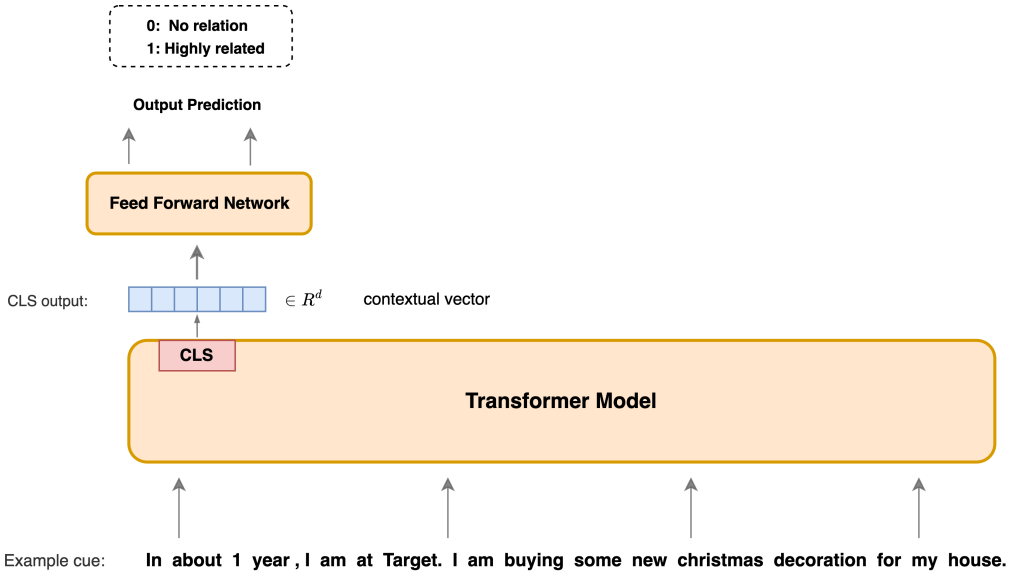


Figure 6.1: Pipeline for Binary Content Characteristics

6.2 Experiment Details

This research involves experiments with three transformer models, namely BERT, RoBERTa, and XLNet, as they have shown to generate state-of-the-art performance on various text classification benchmarks [105]. The versions used are *bert-base-cased*, *roberta-base*, and *xlnet-base-cased*, respectively, from Hugging Face [106]. For the RoBERTa and BERT models, we have $d = 768$, while for the XLNet model, $d = 1024$. The model is trained for 12 epochs with a learning rate of $2e-5$ and the Adam optimizer with $\epsilon = 1e-8$. The batch size is 16 for all of the experiments. The model is fine-tuned on training data and evaluated on a validation set. All of the experiments are performed on NVIDIA's A100 GPUs.

During training, after every epoch, the F1 score is calculated on training and validation data. If the current F1 score obtained on validation data is better (higher) than before, then that better model checkpoint is saved. All of the experiments are conducted for 3 trials, and the average result is reported. During testing, the checkpoint for the best model is loaded and evaluated on test data.

6.3 Evaluation Metrics

For the classification task, F1 score, precision, recall, and accuracy are reported on our datasets [107].

Consider TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

Accuracy

This is defined as the fraction of the model predictions that are correct. It is not particularly useful when the classes are imbalanced. For binary classification, we can calculate accuracy as:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

Precision

Precision shows, among all of the positive predictions made by the model, what fraction of those predictions were correct (truly positive). We can calculate precision as:

$$Prec = \frac{TP}{TP + FP} \tag{6.2}$$

Recall

Recall shows, among all of the positives, what fraction of those were predicted as positive. We can calculate recall as:

$$Rec = \frac{TP}{TP + FN} \tag{6.3}$$

F1 Score

F1 score is often considered a more robust metric as compared to accuracy. It is defined as the harmonic mean of precision and recall. We can calculate the F1 score as:

$$F1 = \frac{2 * Prec * Rec}{Prec + Rec} \tag{6.4}$$

6.4 Results

6.4.1 First Pilot Study

We begin to train the classifiers on the dataset obtained from the first pilot study as shown in Table 4.2. For the first pilot study, since the dataset was small, the validation set is used as the testing set. The experiments follow a *train : val* split of 90% : 10%. All of the experiments are conducted three times and the average result is reported in Table 6.1.

From the table, we can see that the transformer model struggles with categories like Better, Recreation, and Alone, where the performance is poor. These categories have rich semantics, which can sometimes be difficult for humans to comprehend. Recreation often includes texts on fun and leisure activities. Sometimes, these text descriptions can be subtle enough to be misunderstood as a non-recreational event, though it might very well align with a positive label. Since the current dataset volume is low, further experiments are conducted to see if a larger dataset with better balanced annotations can help to address such challenges.

Table 6.1: Results on First Pilot Study for Binary Content Characteristics. **bold** indicates the best performance observed for each category.

Category	Model	F1 Score		Precision		Recall		Accuracy	
		Train	Test	Train	Test	Train	Test	Train	Test
Alone	BERT	91.2	79.9	92	82.1	91.1	79.1	91.2	79.1
	RoBERTa	94.5	**83.7**	94.6	**83.9**	94.5	**83.8**	**94.5**	**83.8**
	XLNet	**97.4**	82.5	**97.5**	83.1	**97.4**	83	83	88.1
Better	BERT	92.1	80.3	92.2	80.3	92.2	80.3	92.2	80.2
	RoBERTa	95.3	81.6	95.3	81.8	95.2	81.4	95.1	81.5
	XLNet	**96.4**	**84.9**	**96.8**	**85**	**96.8**	**84.9**	**96.8**	**84.9**
Celebration	BERT	96.5	92.7	96.5	92.4	96.5	92.7	96.6	92.7
	RoBERTa	96.4	93.9	96.2	94.4	96.4	94	96.4	94
	XLNet	**97**	**94.2**	**97**	**94.5**	**97**	**94.2**	**97**	**94.2**
Family	BERT	95.2	92.5	95.3	92.6	95.3	92.5	95.3	92.6
	RoBERTa	96.9	94.3	96.9	94.3	**96.9**	94.3	96.9	94.3
	XLNet	**97.9**	**95.3**	**97.8**	**95.8**	96.5	**94.6**	**97.9**	**95.8**
Food	BERT	94.5	91.2	94.6	91.1	94.5	91.2	94.5	91.1
	RoBERTa	94.3	92.7	94.4	92.7	94.3	92.7	94.3	92.7
	XLNet	**97.8**	**93.1**	**97.8**	**93.1**	**97.8**	**93.2**	**97.8**	**93.1**
Friends	BERT	95.5	85.7	95.6	87	95.4	85.8	95.4	85.8
	RoBERTa	96.5	86.9	96.6	**88.5**	96.6	87	96.6	87
	XLNet	**98.6**	**88**	**98.7**	88.2	**98.6**	**88**	**98.6**	**88.1**
Future	BERT	98.8	97.5	98.8	97.6	98.8	97.3	98.7	97.5
	RoBERTa	99.5	97.6	99.5	97.6	99.3	97.5	99.5	97.5
	XLNet	**99.8**	**98**	**99.7**	**98.2**	**99.8**	**98**	**99.8**	**98**
Health	BERT	97.4	90.5	97.6	90.6	97.5	90.6	97.5	90.5
	RoBERTa	95.7	92.6	95.9	**93.8**	92.6	92.6	95.8	92.6
	XLNet	**99.8**	**93.3**	**99.8**	93.3	**99.7**	**93.4**	**99.7**	**93.4**
Pet	BERT	97.8	31.2	97.9	25.2	97.9	50	98	50
	RoBERTa	98.6	32.1	98.9	26.2	98.8	50	98.9	50
	XLNet	**99.2**	**34.5**	**99.1**	**27.6**	**99.1**	**52.2**	**99.1**	**52.2**
Recreation	BERT	86.3	81.1	86.3	82.2	86.3	81.3	86.3	81.3
	RoBERTa	87.5	84.4	87.6	**84.8**	87.6	84.5	87.6	84.5
	XLNet	**87.8**	**84.6**	**87.8**	84.6	**87.8**	**84.7**	**87.8**	**84.6**

6.4.2 Second Pilot Study

Next, we train the classifiers on the dataset from the second pilot study. The data distribution from the second pilot study is shown in Table 4.3. For this experiment, a separate testing set is maintained for the model evaluation. The experiments follow a *train : val : test* split of 85% : 5% : 10 %. The model checkpoint is saved based on the best F1 score obtained from the validation set. This checkpoint is later used to evaluate the model on the test set. All of the experiments are conducted three times and the average result is reported in Table 6.2.

Since the data in the second pilot study is twice the size of the data in the first, minor improvements are observed in the F1 score and other metrics for a few categories. From Table 4.3 we can see that the Pet category is highly skewed with the majority of samples belonging to the negative class. Due to this, we observe that the transformer model is unable to learn how to properly classify cues for the Pet category.

6.4.3 Retrieval-enhanced data

In this experiment, the classifiers are trained on the retrieval enhanced data obtained using the framework discussed in Section 5.4. Experiments are performed using SVM and three transformer models: BERT, its distilled version DistillBERT, and XLNet. The used versions are bert-base-cased, distilbert-base-cased, and xlnet-base-cased, respectively, all from Hugging Face [106]. Table 6.3 shows the performance of the classifiers for the three categories.

The first row for each category is the performance observed on the data from the second pilot study (Table 4.3). It is seen that the F1-score on the minority class is less than 90% for all of the models. The second row is after adding a small sample of assumed positive cues for each category. For this study, the top 100 examples retrieved from both the search

Table 6.2: Results on Second Pilot Study Binary Content Characteristics. **bold** indicates the best performance observed for each category.

Category	Model	F1 Score		Precision		Recall		Accuracy	
		Train	Test	Train	Test	Train	Test	Train	Test
Alone	BERT	92.4	83.2	92.8	84.4	92.5	83.4	93	83.4
	RoBERTa	93.6	83.4	93.7	84.5	93.8	83.6	93.6	83.7
	XLNet	**97.2**	**86.7**	**97**	**86.9**	**96.9**	**87**	**97**	**86.8**
Better	BERT	93.2	82.8	93	83.4	93.2	82.8	93.2	82.8
	RoBERTa	94.2	83.7	94.1	84	94.2	83.7	94	84.8
	XLNet	**96.5**	**86.3**	**96.5**	**86.3**	**96.6**	**86.3**	**96.5**	**86.3**
Celebration	BERT	92.3	78.6	92.4	78.7	92.4	78.6	92.4	78.6
	RoBERTa	95.5	86.6	95.5	86.8	95.6	86.7	95.6	86.7
	XLNet	**97.8**	**91.2**	**97.9**	**91.2**	**97.9**	**91.2**	**97.9**	**91.2**
Family	BERT	95.6	94.2	95.7	94.2	95.5	94.2	95.5	94.2
	RoBERTa	96.9	**96.6**	97	**96.4**	97.1	**96.6**	97	**96.6**
	XLNet	**97.8**	96.3	**97.9**	96.3	**97.9**	96.3	**97.8**	96.3
Food	BERT	95.6	93.1	95.7	93	95.7	93	95.7	93
	RoBERTa	95.8	93.7	95.8	93.8	95.8	93.8	95.8	93.8
	XLNet	**97.8**	**95.3**	**97.8**	**95.3**	**97.8**	**95.3**	**97.8**	**95.3**
Friends	BERT	96.4	**87.5**	96.1	87.3	97	87.4	96.3	87.4
	RoBERTa	96.8	87.3	97	**89.3**	96.6	**87.5**	96.5	**87.5**
	XLNet	**97.9**	85	**97.9**	87.6	**98.2**	85	**98.2**	85.1
Future	BERT	98.9	96.5	99	96.8	99	96.5	99	96.6
	RoBERTa	99.5	96.7	99.5	96.9	99.3	96.8	99.5	96.8
	XLNet	**99.8**	**98.4**	**99.7**	**98.4**	**99.8**	**98.4**	**99.8**	**98.4**
Health	BERT	94.2	91.6	94.3	92.1	94.3	91.6	94.3	91.6
	RoBERTa	95.2	92.5	95.2	93	95.2	92.5	95.2	92.5
	XLNet	**98.7**	**94.1**	**98.8**	**94.2**	**98.8**	**94.1**	**98.8**	**94.1**
Partner	BERT	94.7	91.2	95	91.4	95	91.1	95.1	91.1
	RoBERTa	95.7	92.1	95.9	92	95.7	92.2	95.7	92.2
	XLNet	**97.1**	**93.7**	**97**	**93.9**	**97.2**	**93.7**	**97.2**	**93.7**
Pet	BERT	97.8	33.2	97.9	26.2	97.9	50	98	50
	RoBERTa	98.6	34.1	98.9	27.2	98.8	50	98.9	50
	XLNet	**99.2**	**40.5**	**99.1**	**31.6**	**99.1**	**56.2**	**99.1**	**56.2**
Recreation	BERT	87.6	80.8	87.6	81.4	88.1	80.5	87.7	80.5
	RoBERTa	**88.5**	80	**88.7**	80.1	87.6	79.7	**88.6**	79.7
	XLNet	88.4	**85.1**	88.5	**85.2**	**88.5**	**85.2**	88.5	**85.2**

methods were considered to obtain the common cues. Specifically, we obtained 25 (presumed) positives for the alone category and the friend category, and 35 for health. For this scenario, the performance of BERT and XLNET has increased 1% for the alone category, from 87% to 88% and 76% to 77%, for transformer BERT and XLNET, respectively. For the friend category, the BERT model has improved the F1-score to up to 85%. For the health category, transformers BERT and XLNET increase by 1%, and distilBERT increases from 88% to 90%.

Given that the first round of search improved the F1-score, we performed another round of search to identified more examples. In this study, we utilized all of the examples retrieved by both search methods, rather than just the top 100, and applied a majority voting approach to identify the intersection. As a result, we identified 52 (presumed) positive examples (leading to a total of 35+52=87) for the "health" category, 44 (presumed) positive examples (leading to a total of 25+44=69) for the "alone" category, and 33 (presumed) positive examples (leading to a total of 25+33=58) for the "friends" category. We ran the experiments on this newly labeled data. The results are the third row for each category. It is shown that all transformer models have gained more F1-score for alone, i.e., 2% more.

The best transformer is BERT, with macro F1-score of 96%. For friend, the distilBERT F1-score is increased form 85% to 91%. BERT has increased from 85% to 89% and XLNET from 84% 87%. Overall the improvement for all the transformer models is significant for this category. For health, the improvement for distillBERT is 2%, from 88 to 90, and BERT and XlNET increase 1% to 90% and 93%, respectively. The improvement for the SVM classifier is significant in all three categories, ranging from 4% for the alone category, 2% for friends, and 8% for the health category for the minority class.

The results show that using the retrieved cues as potential positive examples increased the F1-score for the minority class leading to more accurate classifiers. For the transformer models, the more data that is identified, the better performance can be gained. For the

Table 6.3: Performance on the retrieval enhanced datasets. * indicates the cue sets obtained for the minority class by considering all the retrieved examples from both the search methods. **bold** indicates the best results obtained.

Category	Data Samples	SVM		DistillBERT		BERT		XLNet	
		Minority F1 score	Macro F1 score	Minority F1 score	Macro F1 score	Minority F1 score	Macro F1 score	Minority F1 score	Macro F1 score
Alone	1600	0.66	0.79	**0.87**	**0.92**	0.87	0.92	0.76	0.86
	1600 + 25	0.65	0.79	0.87	0.92	0.88	0.93	0.77	0.86
	1600 + 69*	**0.70**	**0.82**	0.89	0.93	**0.90**	**0.96**	**0.79**	**0.88**
Friends	1600	0.74	0.85	0.80	0.89	0.82	0.90	0.84	0.90
	1600 + 25	0.73	0.84	0.85	0.91	0.85	0.91	0.84	0.91
	1600 + 58*	**0.76**	**0.86**	**0.91**	**0.95**	**0.89**	**0.94**	**0.87**	**0.92**
Health	1600	0.77	0.85	0.88	0.91	0.88	0.91	0.91	0.94
	1600 + 35	0.83	0.87	0.88	0.91	0.89	0.93	0.92	0.94
	1600 + 87*	**0.85**	**0.88**	**0.90**	**0.93**	**0.90**	**0.93**	**0.93**	**0.95**

SVM algorithm the improvement is more significant, suggesting a very effective method for non-neural machine learning algorithms when they do not require a lot of data examples. Based on these findings, it appears that transformer models can improve their performance even with limited data when dealing with class imbalance. In fact, the proposed retrieval-based framework presented in this work demonstrates that an efficient technique can enhance the model's performance on the minority class by adding only a minimal number of positive samples.

Chapter 7

Modelling for Continuous Content Characteristics

7.1 Approach

Our dataset consists of cues which are textual descriptions of events and plans. This research aims to leverage the knowledge from transformer models by fine-tuning them on the EFT dataset to identify various categories within a cue. For continuous content characteristics, the models are trained to predict a score that indicates how likely it is that the cue belongs to a category. This can be formulated as a regression problem.

Consider an input text cue, $X = [x_1, ..., x_n]$, where x_i is the $i-$th token in the text and n is the length of the sequence. The sequence of tokens is passed as input. The tokenizer library chosen is that used by the transformer model. For continuous categories, the labels are real-valued and scored ranged between 0 - 100, to indicate how likely it is that the cue belongs to the category. For numerical stability, the output labels are scaled between 0 - 10 during training and evaluation. This helps the model to converge better and prevents any gradient explosion. To tackle this regression problem, we obtain the $[CLS]$ (classification head) output from the transformer model [69, 70]. Let $C \in R^d$ be the contextualized output vector obtained from the classification head which has a hidden dimension size d. This vector is then passed as an input to the linear regression model which generates the final score as

a real-valued number [108]. Fig. 7.1 shows the architecture of our pipeline for continuous categories. Separate regression models are trained for each of the continuous categories.

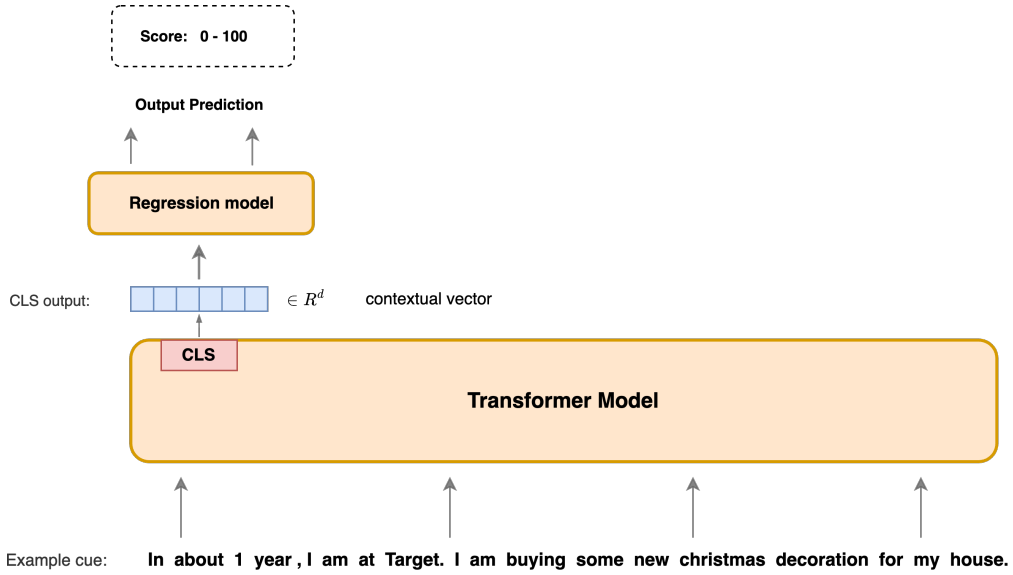


Figure 7.1: Pipeline for Continuous Content Characteristics

7.2 Experiment Details

This research involves experiments with three transformer models, namely BERT, RoBERTa, and XLNet. The versions used are *bert-base-cased*, *roberta-base*, and *xlnet-base-cased*, respectively, from Hugging Face [106]. For the RoBERTa and BERT models, we have $d = 768$, while for the XLNet model, and $d = 1024$. The model is trained for 12 epochs with a learning rate of $2e-5$ for the base transformer model and a learning rate of $2e-3$ for the regression model attached on top. Since this is a regression problem, Mean Squared Error (MSE) loss and the Adam optimizer with $\epsilon = 1e-8$ are used. The batch size is 4 for all the experiments. The model is fine-tuned on training data and evaluated on the validation set. All of the experiments are performed on NVIDIA's A100 GPUs. The best model checkpoint is saved by measuring the Mean Absolute Error (MAE) score on validation data. That best

checkpoint is later used for testing. All of the experiments are conducted for 3 trials and the average result is reported.

7.3 Evaluation Metrics

For the regression task, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are reported [109].

Root Mean Squared Error:

RMSE is a frequently used metric to measure differences in values. It is calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}} \tag{7.1}$$

Mean Absolute Error:

MAE measures the absolute difference between two values. It is calculated as:

$$MAE = \frac{\sum_{i=1}^{N}|\hat{y}_i - y_i|}{N} \tag{7.2}$$

For regression tasks, a Mean Absolute Error (MAE) value within a 10% margin of the range of the dependent variable is often considered a good fit. In this example, the scores are scaled to range between 0 and 10 for numerical stability. In the given scenario, if the actual score is 8.7 and the predicted score is 7.7, then the difference between them is 1.0. Therefore, the mean difference of 1.0 falls within the 10% margin, which is considered a good fit for our regression tasks.

7.4 Results

7.4.1 First Pilot Study

We train the classifiers on the dataset obtained from the first pilot. For continuous categories, the output label is a real-valued score between 0 - 100 for all 800 cues. Since the dataset was small, the validation set is used as the testing set. The experiments follow a *train : val* split of 90% : 10%. All of the experiments are conducted three times and the average result is reported in Table 7.1.

Table 7.1: Results on First Pilot Study for Continuous Content Characteristics. **bold** indicates the best performance observed for each category.

Category	Model	RMSE		MAE	
		Train	**Test**	**Train**	**Test**
Emotion	BERT	1.2	1.3	**0.9**	1.1
	RoBERTa	1.6	1.4	1.3	1.1
	XLNet	**1.1**	**1.1**	0.92	**0.95**
Vivid	BERT	1.5	1.6	1.4	1.4
	RoBERTa	1.7	1.8	1.4	1.5
	XLNet	**1.6**	**1.6**	**1.2**	**1.2**
Episodic	BERT	1.5	1.6	1.3	1.3
	RoBERTa	1.6	1.6	1.3	1.3
	XLNet	**1.5**	**1.5**	**1.2**	**1.2**

7.4.2 Second Pilot Study

Next, we train the regression model on the dataset from the second pilot study. For the second pilot study, there are 1600 examples where each cue has rated a score between 0 -

100. For this experiment, a separate testing set is maintained for the model evaluation. The experiments follow a *train : val : test* split of 85% : 5% : 10 %. The model checkpoint is saved based on the best MAE score obtained from the validation set. This checkpoint is later used to evaluate the model on the test set. All of the experiments are conducted three times and the average result is reported in Table 7.2.

Table 7.2: Results on Second Pilot Study for Continuous Content Characteristics. **bold** indicates the best performance observed for each category.

Category	**Model**	**RMSE**		**MAE**	
		Train	**Test**	**Train**	**Test**
Emotion	BERT	1.0	1.1	**0.78**	0.89
	RoBERTa	1.5	1.3	1.2	1.07
	XLNet	**1.0**	**1.0**	0.85	**0.84**
Vivid	BERT	1.4	1.5	1.2	1.2
	RoBERTa	1.6	1.6	1.3	1.2
	XLNet	**1.5**	**1.5**	**1.1**	**1.2**
Episodic	BERT	1.4	1.5	1.2	1.2
	RoBERTa	1.5	1.5	1.2	1.2
	XLNet	**1.4**	**1.4**	**1.1**	**1.2**

The results of the second pilot study indicate that the "emotion" category shows a 10% improvement in MAE scores, with XLNet outperforming the other models. This suggests that having a larger amount of data can improve performance in this category. In contrast, the performance for the "vivid" and "episodic" categories remains consistent across both pilot studies. This may be due to the fact that these categories have more complex semantics, making it more difficult for AI models to accurately predict continuous scores.

Chapter 8

Conclusion and Future work

8.1 Conclusion

This study employs Elasticsearch and transformer models to analyze content characteristics from Episodic Future Thinking (EFT) cues, a promising approach to understanding decision-making and health outcomes. The use of domain-specific data presents a challenge for accurate modeling due to class imbalance, but the research successfully addresses this by designing suitable queries and using Elasticsearch to try to retrieve relevant cues for imbalanced categories. By doing so, the study could achieve a better balance in training datasets by improving the likelihood of positive labeling by annotators. The results obtained from Table 5.3 and Table 5.4 support our proposed hypotheses *H1* and *H2*.

Further, the research involves experiments on data obtained from pilot studies to build classifiers for predicting binary content characteristics and regression models for predicting continuous content characteristics. As proposed in hypothesis *H4*, the trained transformer models are able to identify content characteristics from the EFT dataset. This identification of different content characteristics may have important implications for medical experts seeking to prevent and manage type 2 diabetes. To address data imbalance and enhance the performance of the model on the minority class, a novel retrieval-based framework was proposed in this study. The results, as depicted in Table 6.3, indicate that the AI models were able to achieve up to a 10% increase in F1 scores by utilizing data obtained from the

proposed framework. This finding confirms the hypothesis *H3* proposed in this study.

The AI models developed in this work can be further improved through the incorporation of additional data and advanced large language models. Overall, this research should contribute to the development of effective interventions for the prevention and management of type 2 diabetes through natural language processing.

8.2 Future Work

Current classifiers and regression models are trained on limited and imbalanced data. As more annotated data is made available in future studies, that data can be considered to improve the performance of the AI models in identifying content characteristics from EFT cues.

Incorporating additional data sources, such as medical records and patient histories, might provide more context and further improve the predictive power of the models. Exploring alternative approaches to address the class imbalance, such as data augmentation, cost-sensitive learning, and active learning, might further enhance the performance of the models.

As a future direction for ablation studies, one potential experiment could involve sending the retrieved examples for annotation and subsequently comparing the accuracy of our framework. For example, if our search framework retrieved 50 examples for the "health" category and after annotation, 45 out of 50 were labeled as positive by the annotators, we could determine that our proposed framework for the "health" category is 90% accurate. However, due to budget constraints, obtaining additional annotations after our resources are used up may present a challenge. Nonetheless, this experiment would provide valuable insight into the effectiveness of our framework and could guide further improvements to our methodology.

The findings of this study can be applied to other chronic illnesses beyond type 2 diabetes, providing opportunities for the development of tailored interventions based on individual content characteristics. Overall, these avenues for future work have the potential to advance the understanding of decision-making and health outcomes in chronic illnesses and contribute to the development of effective interventions for their prevention and management.

Bibliography

[1] Centers for Disease Control and Prevention, "Type 2 diabetes." https://www.cdc.gov/diabetes/basics/type2.html, 2023. Accessed: April 19, 2023.

[2] Centers for Disease Control and Prevention, "Prevent type 2 diabetes." https://www.cdc.gov/diabetes/prevent-type-2/index.html, 2023. Accessed: April 19, 2023.

[3] A. D. Misra-Hebert, A. Milinovich, A. Zajichek, X. Ji, T. D. Hobbs, W. Weng, P. Petraro, S. X. Kong, M. Mocarski, R. Ganguly, J. M. Bauman, K. M. Pantalone, R. S. Zimmerman, and M. W. Kattan, "Natural Language Processing Improves Detection of Nonsevere Hypoglycemia in Medical Records Versus Coding Alone in Patients With Type 2 Diabetes but Does Not Improve Prediction of Severe Hypoglycemia Events: An Analysis Using the Electronic Medical Record in a Large Health System," *Diabetes Care*, vol. 43, pp. 1937–1940, 05 2020.

[4] L. Fregoso-Aparicio, J. Noguez, L. Montesinos, and J. A. García-García, "Machine learning and deep learning predictive models for type 2 diabetes: a systematic review," *Diabetology & Metabolic Syndrome*, vol. 13, p. 148, Dec 2021.

[5] A. Turchin and L. F. Florez Builes, "Using Natural Language Processing to Measure and Improve Quality of Diabetes Care: A Systematic Review," *J Diabetes Sci Technol*, vol. 15, pp. 553–560, Mar. 2021.

[6] Y. Zheng, V. V. Dickson, S. Blecker, J. M. Ng, B. C. Rice, G. D. Melkus, L. Shenkar, M. C. R. Mortejo, and S. B. Johnson, "Identifying patients with hypoglycemia using natural language processing: Systematic literature review," *JMIR Diabetes*, vol. 7, p. e34681, May 2022.

[7] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Science China Technological Sciences*, vol. 63, pp. 1872–1897, Oct 2020.

[8] Elasticsearch, "Elasticsearch." https://www.elastic.co/, 2010. Accessed: April 6, 2023.

[9] M. E. Maron, "Automatic indexing: An experimental inquiry," *J. ACM*, vol. 8, p. 404–417, July 1961.

[10] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[11] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Machine Learning: ECML-98* (C. Nédellec and C. Rouveirol, eds.), (Berlin, Heidelberg), pp. 137–142, Springer Berlin Heidelberg, 1998.

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[13] M. Razno, "Machine learning text classification model with NLP approach," *Computational Linguistics and Intelligent Systems*, vol. 2, pp. 71–73, 2019.

[14] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[15] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "A survey on Text Classification Algorithms: From Text to Predictions," *Information*, vol. 13, no. 2, p. 83, 2022.

[16] J. J. Webster and C. Kit, "Tokenization as the initial phase in NLP," in *COLING 1992 volume 4: The 14th international conference on computational linguistics*, 1992.

[17] W. J. Wilbur and K. Sirotkin, "The automatic identification of stop words," *Journal of information science*, vol. 18, no. 1, pp. 45–55, 1992.

[18] E. Wallace, Y. Wang, S. Li, S. Singh, and M. Gardner, "Do NLP models know numbers? Probing numeracy in embeddings," *arXiv preprint arXiv:1909.07940*, 2019.

[19] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of Arabic word embedding models for use in Arabic NLP," *Procedia Computer Science*, vol. 117, pp. 256–265, 2017.

[20] F. Almeida and G. Xexéo, "Word embeddings: A survey," *arXiv preprint arXiv:1901.09069*, 2019.

[21] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, pp. 43–52, Dec 2010.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv, 2013. https://arxiv.org/abs/1301.3781.

[23] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[24] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[25] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data*

Mining, KDD '16, (New York, NY, USA), p. 785–794, Association for Computing Machinery, 2016.

[26] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 3149–3157, Curran Associates Inc., 2017.

[27] R. A. Stein, P. A. Jaques, and J. F. Valiati, "An analysis of Hierarchical Text Classification using Word Embeddings," *Information Sciences*, vol. 471, pp. 216–232, 2019.

[28] A. Bansal and S. Kaur, "Extreme gradient boosting based tuning for classification in intrusion detection systems," in *Advances in Computing and Data Sciences: Second International Conference, ICACDS 2018, Dehradun, India, April 20-21, 2018, Revised Selected Papers, Part I 2*, pp. 372–380, Springer, 2018.

[29] J. Nam, J. Kim, E. Loza Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale Multi-label Text Classification—Revisiting Neural Networks," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*, pp. 437–452, Springer, 2014.

[30] J. Camacho-Collados and M. T. Pilehvar, "On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis," *arXiv preprint arXiv:1707.01780*, 2017.

[31] A. Krogh, "What are artificial neural networks?," *Nature biotechnology*, vol. 26, no. 2, pp. 195–197, 2008.

[32] J. Zou, Y. Han, and S.-S. So, "Overview of Artificial Neural Networks," *Artificial Neural Networks: Methods and Applications*, pp. 14–22, 2009.

[33] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[34] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[35] R. Churchill and L. Singh, "The Evolution of Topic Modeling," *ACM Comput. Surv.*, vol. 54, Nov 2022. https://doi.org/10.1145/3507900.

[36] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation Classification via Convolutional Deep Neural Network," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, (Dublin, Ireland), pp. 2335–2344, Dublin City University and Association for Computational Linguistics, Aug. 2014.

[37] M. Gardner and S. Dorling, "Artificial Neural Networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, no. 14, pp. 2627–2636, 1998.

[38] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 809–821, 2015.

[39] Hecht-Nielsen, "Theory of the backpropagation neural network," in *International 1989 Joint Conference on Neural Networks*, pp. 593–605 vol.1, 1989. DOI 10.1109/IJCNN.1989.118638.

[40] S. K. Pal and S. Mitra, "Multilayer Perceptron, Fuzzy sets, and Classification," *IEEE Transactions on neural networks*, vol. 3, no. 5, pp. 683–697, 1992.

[41] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5-6, pp. 183–197, 1991.

[42] F. Belletti, M. Chen, and E. H. Chi, "Quantifying long range dependence in language and user behavior to improve RNNs," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1317–1327, 2019.

[43] G. Di Gennaro, A. Buonanno, and F. A. Palmieri, "Considerations about learning Word2Vec," *The Journal of Supercomputing*, pp. 1–16, 2021.

[44] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation*, p. 318–362. Cambridge, MA, USA: MIT Press, 1986.

[45] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[47] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[48] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, (Doha, Qatar), pp. 103–111, Association for Computational Linguistics, Oct. 2014.

[49] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1746–1751, Association for Computational Linguistics, Oct. 2014.

[50] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM Neural Network for Text Classification," arXiv, 2015. https://arxiv.org/abs/1511.08630.

[51] J. Y. Lee and F. Dernoncourt, "Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, California), pp. 515–520, Association for Computational Linguistics, June 2016.

[52] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Baltimore, Maryland), pp. 655–665, Association for Computational Linguistics, June 2014.

[53] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, (Cambridge, MA, USA), p. 649–657, MIT Press, 2015.

[54] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-Aware Neural Language Models," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 2741–2749, AAAI Press, 2016.

[55] L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin, "Natural Language Inference by Tree-Based Convolution and Heuristic Matching," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Berlin, Germany), pp. 130–136, Association for Computational Linguistics, Aug. 2016.

[56] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text Matching as Image Recognition," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 2793–2799, AAAI Press, 2016.

[57] S. Peng, R. You, H. Wang, C. Zhai, H. Mamitsuka, and S. Zhu, "DeepMeSH: deep semantic representation for improving large-scale MeSH indexing," *Bioinformatics*, vol. 32, pp. i70–i79, June 2016.

[58] M. Hughes, I. Li, S. Kotoulas, and T. Suzumura, "Medical Text Classification Using Convolutional Neural Networks," *Stud Health Technol Inform*, vol. 235, pp. 246–250, 2017.

[59] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," 2016. https://arxiv.org/abs/1409.0473.

[60] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical Attention Networks for Document Classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, California), pp. 1480–1489, Association for Computational Linguistics, June 2016.

[61] T. Shen, J. Jiang, T. Zhou, S. Pan, G. Long, and C. Zhang, "DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, AAAI Press, 2018.

[62] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning Natural Language Inference using Bidirectional LSTM model and Inner-Attention," 2016. https://arxiv.org/abs/1605.09090.

[63] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint Embedding of Words and Labels for Text Classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 2321–2331, Association for Computational Linguistics, July 2018.

[64] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," arXiv, 2017. https://arxiv.org/abs/1703.03130.

[65] S. Wang, M. Huang, and Z. Deng, "Densely Connected CNN with Multi-Scale Feature Attention for Text Classification," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, p. 4468–4474, AAAI Press, 2018.

[66] I. Yamada and H. Shindo, "Neural Attentive Bag-of-Entities Model for Text Classification," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, (Hong Kong, China), pp. 563–573, Association for Computational Linguistics, Nov. 2019.

[67] C. Du and L. Huang, "Text classification research with attention-based recurrent neural networks," *International Journal of Computers Communications & Control*, vol. 13, no. 1, pp. 50–61, 2018.

[68] R. Jing, "A self-attention based LSTM network for text classification," in *Journal of Physics: Conference Series*, vol. 1207, p. 012008, IOP Publishing, 2019.

[69] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[70] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[71] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv, 2019. https://arxiv.org/abs/1907.11692.

[72] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," arXiv, 2019. https://arxiv.org/abs/1909.11942.

[73] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 2019. https://arxiv.org/abs/1910.01108.

[74] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[75] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *J. Mach. Learn. Res.*, vol. 21, jan 2020.

[76] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced LSTM for natural language inference," *arXiv preprint arXiv:1609.06038*, 2016.

[77] A. Ševčík and M. Rusko, "A Systematic Review of Alzheimer's disease detection based on speech and natural language processing," in *2022 32nd International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 01–05, IEEE, 2022.

[78] R. K. Karunanayake, W. M. Dananjaya, M. Y. Peiris, B. Gunatileka, S. Lokuliyana, and A. Kuruppu, "CURETO: skin diseases detection using image processing and CNN," in *2020 14th international conference on Innovations in Information Technology (IIT)*, pp. 1–6, IEEE, 2020.

[79] S. Adhikari, S. Thapa, U. Naseem, P. Singh, H. Huo, G. Bharathy, and M. Prasad, "Exploiting linguistic information from Nepali transcripts for early detection of Alzheimer's disease using natural language processing and machine learning techniques," *International Journal of Human-Computer Studies*, vol. 160, p. 102761, 2022.

[80] X. Geets, J.-F. Daisne, S. Arcangeli, E. Coche, M. De Poel, T. Duprez, G. Nardella, and V. Grégoire, "Inter-observer variability in the delineation of pharyngo-laryngeal tumor, parotid glands and cervical spinal cord: comparison between CT-scan and MRI," *Radiotherapy and oncology*, vol. 77, no. 1, pp. 25–31, 2005.

[81] D. Chiumello, M. Busana, S. Coppola, F. Romitti, P. Formenti, M. Bonifazi, T. Pozzi, M. M. Palumbo, M. Cressoni, P. Herrmann, *et al.*, "Physiological and quantitative CT-scan characterization of COVID-19 and typical ARDS: a matched cohort study," *Intensive care medicine*, vol. 46, pp. 2187–2196, 2020.

[82] T. A. Koleck, C. Dreisbach, P. E. Bourne, and S. Bakken, "Natural language processing of symptoms documented in free-text narratives of electronic health records: a systematic review," *Journal of the American Medical Informatics Association*, vol. 26, no. 4, pp. 364–379, 2019.

[83] L. Ohno-Machado, "Realizing the full potential of electronic health records: the role of natural language processing," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 539–539, 2011.

[84] L. Li, H. S. Chase, C. O. Patel, C. Friedman, and C. Weng, "Comparing ICD9-encoded diagnoses and NLP-processed discharge summaries for clinical trials pre-screening: a case study," in *AMIA Annual Symposium Proceedings*, vol. 2008, p. 404, American Medical Informatics Association, 2008.

[85] A. Névéol, P. Zweigenbaum, *et al.*, "Making sense of big textual data for health care: findings from the section on clinical natural language processing," *Yearbook of medical informatics*, vol. 26, no. 01, pp. 228–234, 2017.

[86] L. Zheng, Y. Wang, S. Hao, A. Y. Shin, B. Jin, A. D. Ngo, M. S. Jackson-Browne, D. J. Feller, T. Fu, K. Zhang, X. Zhou, C. Zhu, D. Dai, Y. Yu, G. Zheng, Y.-M. Li, D. B. McElhinney, D. S. Culver, S. T. Alfreds, F. Stearns, K. G. Sylvester, E. Widen, and X. B. Ling, "Web-based real-time case finding for the population health management of patients with diabetes mellitus: A prospective validation of the natural language processing–based algorithm with statewide electronic medical records," *JMIR Med Inform*, vol. 4, p. e37, Nov 2016.

[87] Y. Jin, F. Li, V. G. Vimalananda, and H. Yu, "Automatic Detection of Hypoglycemic Events From the Electronic Health Record Notes of Diabetes Patients: Empirical Study," *JMIR Med Inform*, vol. 7, p. e14340, Nov 2019.

[88] N. K. Mishra, R. Y. Son, and J. J. Arnzen, "Towards automatic diabetes case detection and ABCS protocol compliance assessment," *Clinical Medicine & Research*, vol. 10, no. 3, pp. 106–121, 2012.

[89] K. Zhang, X. Liu, J. Xu, J. Yuan, W. Cai, T. Chen, K. Wang, Y. Gao, S. Nie, X. Xu, X. Qin, Y. Su, W. Xu, A. Olvera, K. Xue, Z. Li, M. Zhang, X. Zeng, C. L. Zhang, O. Li, E. E. Zhang, J. Zhu, Y. Xu, D. Kermany, K. Zhou, Y. Pan, S. Li, I. F. Lai, Y. Chi, C. Wang, M. Pei, G. Zang, Q. Zhang, J. Lau, D. Lam, X. Zou, A. Wumaier, J. Wang, Y. Shen, F. F. Hou, P. Zhang, T. Xu, Y. Zhou, and G. Wang, "Deep-learning models for the detection and incidence prediction of chronic kidney disease and type 2 diabetes from retinal fundus images," *Nature Biomedical Engineering*, vol. 5, pp. 533–545, Jun 2021.

[90] J. Wu, F. Morrison, Z. Zhao, G. Haynes, X. He, A. K. Ali, M. Shubina, S. Malmasi, W. Ge, X. Peng, and A. Turchin, "Reasons for discontinuing insulin and factors associated with insulin discontinuation in patients with type 2 diabetes mellitus: a real-world evidence study," *Clinical Diabetes and Endocrinology*, vol. 7, p. 1, Jan 2021.

[91] A. Bartal, K. M. Jagodnik, S. J. Chan, M. S. Babu, and S. Dekel, "Identifying women with postdelivery posttraumatic stress disorder using natural language processing of personal childbirth narratives," *American Journal of Obstetrics Gynecology MFM*, vol. 5, no. 3, p. 100834, 2023.

[92] N. Vandenbussche, C. Van Hee, V. Hoste, and K. Paemeleire, "Using natural language processing to automatically classify written self-reported narratives by patients with migraine or cluster headache," *The Journal of Headache and Pain*, vol. 23, p. 129, Sep 2022.

[93] R. van Genugten and D. L. Schacter, "Automated scoring of the autobiographical interview with natural language processing," 2022. PsyArXiv, https://-doi.org/10.31234/osf.io/nyurm.

[94] L. R. Fischer, W. A. Rush, J. C. Kluznik, P. J. O'Connor, and A. M. Hanson, "Ab-

stract C-C1-06: identifying depression among diabetes patients using natural language processing of office notes," *Clinical Medicine & Research*, vol. 6, no. 3-4, pp. 125–126, 2008.

[95] T. A. Koleck, C. Dreisbach, P. E. Bourne, and S. Bakken, "Natural language processing of symptoms documented in free-text narratives of electronic health records: a systematic review," *Journal of the American Medical Informatics Association*, vol. 26, pp. 364–379, 02 2019.

[96] S. Sultana, R. Zhang, H. Lim, and M. Antoniak, "Narrative Datasets through the Lenses of NLP and HCI," in *Proceedings of the Second Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, (Seattle, Washington), pp. 47–54, Association for Computational Linguistics, July 2022.

[97] V. Vidyadharan, M. Hamdan, and A. M. S. Zalzala, "An evidence-based study of diabetes prevention and management with nlp and deep learning," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, 2021.

[98] J. S. Stein, A. G. Wilson, M. N. Koffarnus, T. O. Daniel, L. H. Epstein, and W. K. Bickel, "Unstuck in time: episodic future thinking reduces delay discounting and cigarette smoking," *Psychopharmacology*, vol. 233, pp. 3771–3778, 2016.

[99] J. S. Stein, A. N. Tegge, J. K. Turner, and W. K. Bickel, "Episodic future thinking reduces delay discounting and cigarette demand: an investigation of the good-subject effect," *Journal of behavioral medicine*, vol. 41, pp. 269–276, 2018.

[100] Amazon Mechanical Turk, "Amazon Mechanical Turk." https://www.mturk.com/, 2005. Accessed: April 6, 2023.

[101] Y. Dong and M. Oyamada, "Table enrichment system for machine learning," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3267–3271, 2022.

[102] V.-A. Zamfir, M. Carabas, C. Carabas, and N. Tapus, "Systems monitoring and big data analysis using the Elasticsearch system," in *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, pp. 188–193, IEEE, 2019.

[103] Z. Liu, J. Feng, Z. Yang, and L. Wang, "Document retrieval for precision medicine using a deep learning ensemble method," *JMIR Medical Informatics*, vol. 9, no. 6, p. e28272, 2021.

[104] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 4768–4777, Curran Associates Inc., 2017.

[105] R. Stojnic, R. Taylor, M. Kardas, and Elvis, "Text classification." https://paperswithcode.com/task/text-classification. Accessed: April 6, 2023.

[106] T. Wolf, V. Sanh, J. Chaumond, and C. Delangue, "Hugging face: State-of-the-art natural language processing." https://huggingface.co/, 2019. Accessed: April 6, 2023.

[107] G. Forman *et al.*, "An extensive empirical study of feature selection metrics for text classification.," *J. Mach. Learn. Res.*, vol. 3, no. Mar, pp. 1289–1305, 2003.

[108] J. Xin, R. Tang, Y. Yu, and J. Lin, "BERxiT: Early exiting for BERT with better fine-tuning and extension to regression," in *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: Main Volume*, pp. 91–104, 2021.

[109] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "On mean absolute error for deep neural network based vector-to-vector regression," *IEEE Signal Processing Letters*, vol. 27, pp. 1485–1489, 2020.