DETECTING NO-OPINION RESPONSES IN THE CANADIAN LONGITUDINAL STUDY ON AGING (CLSA) DATASET USING UNSUPERVISED METHODS AND ACTIVE LEARNING

by

Disha Malik

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
February 2022

To those who - lovingly and selflessly - made this possible.
My parents and sister.

Contents

List of Tables

List of Figures

Abstract

In open-ended surveys, participant answers that do not give any legitimate answer or opinion to the question being asked are called no-opinion responses. We consider the problem of detection of no-opinion answers in the CLSA dataset using a Machine Learning approach. The CLSA dataset contains verbatim responses from over 51,000 participants to the question of what promotes healthy aging. Our foremost goal is to clean the CLSA dataset to help foster the healthy aging study and pave a healthier way forward for the future generations. This thesis investigates the performance of existing state-of-the art approaches, using distance measures coupled with embeddings and Active Learning to cluster and classify no-opinion responses. Among the unsupervised techniques we obtained the best performance using the BERT embeddings with Euclidean Distance. We also show that the Active Learning approach is viable approach to identify no-opinion responses in a large survey, and in our experiments the SVM base classifier had the best performance of 0.97 in the AUC score of PR curve. Using this approach we identified 1157 instances of no-opinion responses in the CLSA dataset.

List of Abbreviations Used

AL - Active Learning
AMI - Adjusted Mutual Information
AUC - Area Under the Curve
BERT - Bidirectional Encoder Representations from Transformers
BOW - Bag of Words
CIHR - Canadian Institute of Health Research
CLSA - Canadian Longitudinal Study on Aging
CS - Cosine Similarity
CSHA - Canadian Study on Health and Aging
DCS - Data Collection Site
FMI - Fowlkes Mallows Index
MI - Mutual Information
ML - Machine Learning
MLM - Masked Language Model
NB - Naïve Bayes
NER - Named Entity Recognition
NLP - Natural Language Processing
NN - Neural Network
PR - Precision-Recall
RF - Random Forest
ROC - Receiver Operating Characteristics
SCS - Soft Cosine Similarity
SVM - Support Vector Machine
TF-IDF - Term Frequency–Inverse Document Frequency
VM - V- Measure
WMD - Word Mover's Distance

Acknowledgements

This thesis marks a significant period of my exciting and challenging trajectory. Completing a Master's course abroad is a journey on its own, and I would like to thank all the people that supported me in this process.

I want to express my gratitude to my supervisor, Dr. Vlado Keselj, for his guidance and support throughout my time at Dalhousie. His feedback and insights helped me take the right direction in my next step forward. In addition, I want to extend my gratitude to my co-supervisor, Dr. Dijana Kosmajac, for her time and for giving critical inputs, which helped shape this thesis. I appreciate their invaluable help, ideas and mentoring very much. Furthermore, I am indebted to them for their continued support and patience throughout my Master's and for giving me this opportunity. I also want to thank the other members of my thesis committee, Dr. Evangelos Milios and Dr. Srinivas Sampalli, for taking their time and effort to review my thesis work.

I want to extend my indebtedness to the people who give sense to my life and work, my family, Harendra, Pavitra and Shreya. Without their love, support, guidance and sacrifices, I would not have accomplished this milestone in my career. Without them, nothing would have mattered: there would have been no love, nothing to care for, to smile at, or to be proud of.

I'm also blessed with beautiful friends; in many ways, my successes are theirs, too. If we were to live it all over again, I could not think of anything they didn't already do for me. I want to thank my friends back home, thank you for making me feel like I never left and for being so close despite being so far away. I also want to thank all the fantastic people I had an opportunity to meet during my time at Dalhousie. Thank you for your feedback, support and friendship. Without you, something would have been missing from this journey, and I would not be standing where I am today.

Without the support of anyone mentioned above, I would not have been able to complete this journey.

This research was made possible using the data collected by the Canadian Longitudinal Study on Aging (CLSA). Funding for the CLSA is provided by the Government of Canada through the Canadian Institutes of Health Research (CIHR) under grant reference: LSA 9447 and the Canada Foundation for Innovation. This research has been conducted using the CLSA dataset [Baseline Tracking Dataset version 3.2, Comprehensive Dataset version 3.1], under Application Number [170305]. The CLSA is led by Drs. Parminder Raina, Christina Wolfson and Susan Kirkland.

Chapter 1

Introduction

This chapter explains the thesis topic and why this thesis was built around a particular subject of concern. This chapter acquaints readers with the motivation of this thesis, survey data and concludes with the research objectives of this study.

Surveys facilitate the collection of a wide variety of information. Researchers, businesses and governments conduct surveys to understand and improve products and to collect demographic data. Data collected through surveys are enormous and often in a raw format, unstructured and unlabeled, making it difficult to process the data. It is also tedious to manually skim through the data and remove noise.

Text similarity measures play an increasingly significant role in text-related research and facilitate information retrieval, text classification, topic detection, question answering, machine translation and text summarization [25]. These techniques can help decipher various patterns or commonalities among the opinions given in the survey.

In the age when amounts of data are growing faster, the demand for using Machine Learning ML techniques for processing and analyzing the data is also increasing. Since not all data is valuable and cannot be easily combined, the demand for solving this classification problem has become significant [35]. The main categories of machine learning methods that can be applied are: supervised, semi-supervised, or unsupervised.

The textual format is one of the most common types of data formats in real-world industrial settings. Various prediction tasks have been applied to texts, such as sentiment analysis, summarization and question answering. However, semantic similarity measure is one of the most practical prediction tasks defined as measuring and recognizing semantic relations between two texts. Semantic similarity between sentences in NLP is a complex task as the meaning of words changes significantly with the change in context. Semantic similarity has a wide range of applications in Natural

Language Processing (NLP). For example, it is used to estimate the relatedness between search engine queries and generated keywords for web advertising, it is used in the biomedical field for analyzing gene clustering and gene expansion; and it is also beneficial in information retrieval, text summarization and categorization.

Various methodologies can be used to calculate semantic similarity across multiple domains. Since the concept of calculating semantic similarities has a common underlying conceptual foundation regardless of the domain, a methodology with a robust algorithm that can accurately estimate semantic similarity while incorporating a variety of domain-specific predefined standard language measures is desirable. To improve the existing algorithms that determine the closeness of implications of the objects under comparison, it is clear that a domain-specific predefined standard measure that readily describes the relatedness of the meanings in context is necessary. If we use natural language to compare the natural language sentences, it would be a recursive problem with no stopping condition. Hence, it is essential to have some predefined measures.

1.1 Survey Data

Survey research is used to answer questions, solve problems posed or observed, assess needs and set goals, establish baselines for comparisons, analyze trends across time, describe what exists, in what amount and context. Kraemer [36] talks about three distinctive characteristics of survey research:

- It is used to explain specific aspects of a given population quantitatively.
- The data required for research is subjective as it is collected from people.
- It uses a portion of the population and later generalizes the findings to the population.

A survey is a simple data collection tool for carrying out survey research. Pinsonneault and Kraemer [71] described a survey as a "means for collecting information about the attributes, actions, or ideas of a large group of people." Surveys can also be used to evaluate needs and demands and examine the consequence [81].

Since surveys can collect information from large samples of the population, they are well suited for collecting demographic data that can explain the sample's composition [56]. However, surveys are not suitable when it comes to an understanding of the historical context of events. Surveys can also extract information about complex attitudes using observational techniques. Nevertheless, biases might occur, either from lack of response or from the nature and accuracy of the received responses. However, it is essential to note that surveys only provide estimates for the actual population, not exact measurements [81].

1.1.1 Types of Survey Questions

Open-ended survey questions can help collect many possible responses, and the participants can give their thoughts, opinions or concerns about the question. Open-ended questions make it possible to gather a varied range of informal answers from the respondents. Open-ended questions also allow the researcher to explore ideas that would not otherwise be aired and are helpful in seeking additional insights. They are also helpful when researchers are unfamiliar with the subject area and cannot provide specific answer options. Open-ended questions require more reflection and thinking from the interviewee, so more time is needed to answer them. The results obtained from open-ended questions are also more challenging to analyze. Finally, it is more difficult to determine a single course of action from the broad responses received to open-ended questions.

In contrast, *closed-ended survey questions* require the respondent to examine the given choices and choose from a given set of responses. The choices form a continuum of 2–7 responses, for example, Likert scales [49] and numerical ranges. These types of questions are the easiest to answer and analyze. The other closed-ended questions are:

- closed-ended question with ordered choices
- closed-ended questions with unordered choices
- partially closed-ended questions.

1.1.2 Data Collection and Storage

Traditionally, data was collected using paper-pencil or face-to-face. Advancement and innovations of technology have opened up new possibilities for improving and expanding survey capabilities and collection methods [10]. From face-to-face surveys to telephonic surveys to online and email surveys, the world of survey data collection has changed with time.

Face-to-face interviews are one of the oldest and widely used methods. It also achieves the highest response rate (70%) since the participant is more committed to participating [13]. They allow for in-depth data collection and thorough understanding. Along with responses, participant's body language and facial expressions are more clearly identified and understood, which is an added advantage. However, face-to-face interviews can be expensive as they include time and travel. To overcome this, organizations have started to collect data through telephones.

Since everyone has a telephone or cell/mobile phone, telephone interviews, also known as CATI or Computer-Assisted Telephonic Interviews, can be used to reach samples over a wide geographic area. One major disadvantage of this mode is that the interviewee has the control to answer the call or not. Moreover, unlike a face-to-face interview, interviewers cannot see body language. Furthermore, given the relatively widespread aversion to telemarketers, participants may sometimes perceive legitimate research interviews as sales calls and refuse to participate.

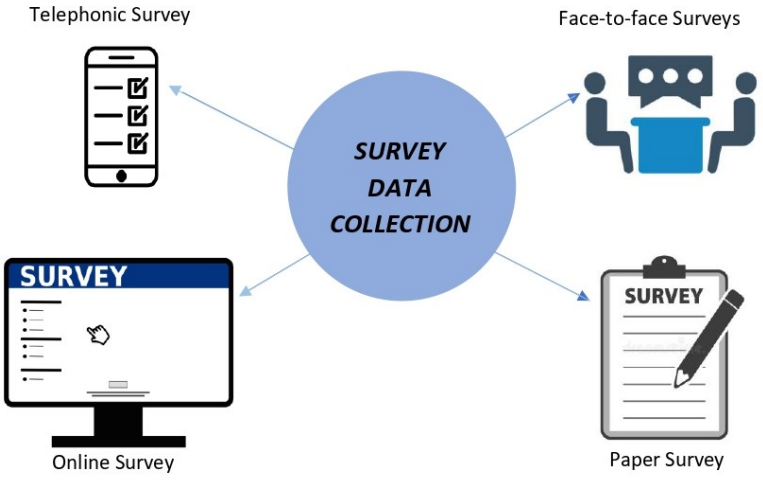

Figure 1.1: Survey data collection methods

Web-based surveys have several benefits over conventional paper or face-to-face methods. They are the most popular, accessible and inclusive form of the survey. They allow a further reach, potentially including a global audience, but there is no guarantee that the survey will be taken or completed, making it prone to errors and leading to a low response rate. However, the data is captured in electronic format, which facilitates faster and cheaper analysis.

There are various reasons for collecting data. A vast collection of data helps researchers ensure and maintain the integrity of the research question while also reducing the likelihood of errors in the results. Data collection is essential for researchers to make informed decisions and support new ideas, changes or innovations.

The data collected in surveys fall under two categories: primary data and secondary data. Primary data is the raw data and is further divided into two segments: qualitative and quantitative data. Quantitative data is any quantifiable information, therefore numbers or categories. It is used to answer "How many?" or "How much?" questions. Moreover, it is usually used for mathematical calculations or statistical analysis, which helps obtain helpful information represented in graphs or charts. In contrast, qualitative data is non-quantifiable, descriptive, and regards phenomena that can be observed but not measured, such as languages, feelings or emotions.

On the other hand, Secondary Data is the second-hand data that already exist; for example, already published books, journals or portals. The use of primary data and secondary data depends on nature, scope and area of research.

1.2 No-Opinion Responses

Surveys are used in a wide range of disciplines. In general, they are directed at collecting participant's opinions on a given issue. The primary goal of public opinion research is the description of opinions held by a population. While surveys offer an efficient and effective means of gathering large amounts of data in a relatively short period, they come with some disadvantages. Surveys can be characterized by low response rates, which may have a biasing effect on information gathered. From the perspective of conversational norms, the mere fact that a person is asked a question presupposes that they can answer it. Thus, responding that one has no opinion is an illegitimate answer to an opinion question [22]. The problem of no-opinion response

arises when participants do not provide an answer for a question in the survey. When the surveyor asks participants about their opinions, they usually presume that their answers reflect some information. However, when participants respond that they have no knowledge or no-opinion or don't know how they feel about it, we call such responses as no-opinions or no-responses since we do not have an actually usable response from the participant.

No-opinion responses lead to missing data which produces less valid data and estimates. While also making it difficult to use standard statistical techniques conceived to deal with complete data and bias due to systematic differences. For automated tasks such as clustering and toping modelling, no-opinion responses are a source of noise. However, it is challenging, time-consuming, error-prone and subjective to remove no-opinion responses manually for automated methods. It would be beneficial to use machine learning methods to overcome these drawbacks and challenges. Moreover, the CLSA dataset consists of over 40,000 responses, making it difficult to clean the data manually. Furthermore, in the previous analysis, a number of no-opinion responses were observed in the dataset.

1.3 Research Objectives

Understanding the factors involved in healthy aging is essential to pave a healthier way forward for our future generations. The CLSA dataset contains potential ways to age healthier and needs future processing to identify the various themes present. The dataset is substantial enough to make manual processing to remove no-opinion responses unfeasible.

Determining the similarity between sentences is one of the crucial tasks in NLP and has a broad impact in many text-related research fields. A similarity measure assigns a ranking score between a query and texts in a corpus in information retrieval, and this is one of a few examples of sentence semantic similarity application [66].

Computing sentence similarity is not a trivial task due to the variability of expression in natural language. Techniques for detecting similarity between long texts (documents) focus on analyzing shared words, but word co-occurrence may be rare or even null in short texts. That is why sentence semantic similarities incorporate the syntactic and semantic information extracted at the sentence level. However,

techniques for detecting similarity between long texts must still be considered because their adaptation can be used to compute sentence similarity as same principles and techniques, or modified versions can solve the problem with short sentences.

The primary goals for this thesis are outlined below:

- Given the problem of open-ended surveys and sentence similarity, we would like to find a solution to detect no-opinion responses in the CLSA dataset.
- Various Machine Learning approaches can be used to solve the problem of no-opinion responses. In this thesis, existing state-of-the-art approaches of sentence similarity have been evaluated and compared.

The ultimate goal of this research is to provide the CLSA with a clean dataset that can help foster text study; thus, some of the experiments presented in this thesis will be applied to the real-life dataset where the genuine issue of no-opinion responses was tried to be resolved.

1.4 Thesis Overview

All the research exposed in this thesis is just a small set of the literature's methods, architectures, and models. All of them offer different features and can be used in specific ways in the NLP context.

With this amount of techniques for the similarity between texts, an analysis and a comparison are needed, and a study of the strengths and weaknesses of the different approaches to extract a set of conclusions to guide the method selected in projects that work with text similarity.

This thesis is organized in 5 chapters, where this is the first and the remainder of this thesis is organized as follows:

- Chapter 2 describes the algorithms and models used and the related research work.
- Chapter 3 provides an overview of CLSA and its dataset. It explains the data pre-processing step. Further, it discusses the methodology used to conduct the experiments in our research.

- Chapter 4 discusses the experimental setup used to compare the performance of each model. The results are profoundly studied to distinguish which model yields the best result.
- Chapter 5 concludes by summarizing the thesis and proposing directions for potential future work.

Chapter 2

Background and Related Work

This chapter discusses the background and related work in the field of Ageing, No-Opinion Responses, Machine Learning, and Active Learning.

2.1 Previous Work Related to the CLSA Project

Intensive research and studies have been done on baseline data provided by CLSA to find factors that can improve our understanding of healthy ageing. Similar research was also conducted in the past on the CSHA dataset, which focused on people having dementia and how it affects the caregivers. It also included Alzheimer's disease and other health topics.

Definition of "Healthy Aging" keeps changing from person to person. Factors such as sex, household income, ethnicity and education, affect a person's understanding of healthy ageing. Data collected by CLSA is analyzed to find how older Canadian adults from different ethnic groups define ageing by Shooshtari et al. [88]. After analyzing the responses of 21,241 Canadians aged between 45–85, it was found that the most common themes in all ethnocultural groups were related to "lifestyle", "physical activity", and "attitude". Impact of exercise on the older Canadian population have also been investigated in a study [29]. The study examines 6,297 community-dwelling elderly Canadians. Findings from this study suggested that people who exercised daily were younger and fitter when compared to people who did little or no exercise. It was also found that a high level of physical activity reduced the risk of death and improved health status in the elderly population. Finally, it can be concluded that lifestyle is related to themes like smoking habit and alcohol intake [80].

2.2 Previous Work on Detecting No-Opinion Responses

In survey-based research, careless, no-answer, or no-opinion responses are a concern. Researchers need to go through the data to screen out such responses. Such responses can reduce internal consistency, reliability and potentially result in erroneous results. A clean dataset is highly desirable in survey-based researche, and data is commonly screened to remove inappropriate responses.

In literature, "no-opinion" responses have been studied for a long time [38]. To include no-option in a response scale has been a debatable issue [70, 48, 5, 72]. A persistent option to deal with such responses has been to ignore them or omit subjects from the study, i.e., listwise and pairwise deletion. Implementing a prompt response to no-opinion responses has also proved to be beneficial in reducing non-meaningful responses. Methodological strategies to minimize non-meaningful responses (e.g. probing) are also recommended, and a few have been tested for their impact [39, 12]. Alternatively, algorithms and data augmentation techniques have also been used. Imputation methods have been used to handle missing data to fill in such responses to complete the dataset for further analysis so that information from the value elicitation question is not lost [78, 50].

A probabilistic framework for treating "no-opinion" responses was proposed by Manisera and Zuccolotto [53]. They considered them as a valid response and defined a framework to exploit the information contained in those responses. The proposed model replaces a few no-opinion responses with a substantive response by firstly deleting all the no-opinion responses and the themes including them and then adjust the uncertainty parameters' estimates.

Another approach compares a methodological strategy to three standard analytic practices to prevent no-opinion responses, proposed by Denman et al. [14]. The methodological strategy gives participants who respond with no-opinion a prompt. The analytic practices excluded no-opinion responses from analyses, considered no-opinion as neutral values on the Likert scale, and replaced no-opinions with computed item level mean. The results suggested that prompt, reassuring participants that their opinion is solicited and giving them a second chance to consider their response is a superior method for managing no-opinion responses than analytical treatments.

Most of the study has been conducted on detecting no-opinion responses in close-ended surveys, even though open-ended surveys suffer from a higher rate of no-opinion response [59]. They also have a high cost in analyzing responses. Regardless, it would be easier to design automated surveys to mitigate the problem of no-opinion detection in open-ended surveys. However, in open-ended surveys, brief responses are typically sparse and respondents produce different responses and generate frequent or infrequent mentions of topics that can have different importance to the respondents, thus making automatic detection of no-opinion responses difficult.

As discussed in Section 1.2, no-opinion responses make the data and estimates less valid. Though substantial work has been done to detect no-opinion responses, it primarily deals with close-ended surveys. In literature, Machine Learning algorithms such as Expectation-Maximization have been used to filter out and reduce the number of no-opinion responses in close-ended surveys. However, the use of Machine Learning algorithms and techniques in open-ended surveys is limited to analyzing and identifying the context and themes in the responses collected. To the best of our knowledge, no prior work utilizing ML algorithms has been done to detect no-opinion responses in open-ended surveys and this is the first of its kind research.

2.3 Machine Learning Methods for Text Similarity

Machine Learning is a broad subfield of artificial intelligence and has been around since the 1950s. It involves developing algorithms and techniques which allow computers to learn. In earlier days, symbolic data was used, and algorithm design was based on logic [94]. Artificial neural network learning came after 1986 when authors proposed the non-linear back-propagation algorithm [55]. The capacity of ML to learn from experience, analytical observation, and other means result in a system continuously improving itself, increasing efficiency.

Machine learning has become one of the backbones of information technology in the past two decades, thus an essential but hidden part of our lives. However, individuals and corporations' increasing amount of data generated (and stored) daily demands an intelligent analysis. It is here where machine learning comes to the stage as a necessary ingredient for technological progress [92]. It has a wide range of applications such as medical diagnosis [46], bio-informatics [19], detecting credit

card fraud, classifying DNA sequences [23], speech and handwriting recognition [4], object recognition in computer vision [17], and robot locomotion.

Some data is provided to the learner, which is divided into training and test sets. The training set consists of a set of input points in multi-dimensional space. The goal is to map from the input points to the labels corresponding to some interest categories in some domain. The testing set is a set of examples that are used to assess the performance of the learner. Learning is about generalization. There are two types of learning called *Inductive Learning* and *Transductive Learning* [105]. In Inductive Learning, the task is to build a good classifier on the training set to generalize any unseen data. The test set is unknown at the time of training. However, in Transductive Learning, the learner knows the test set at the time of training and therefore only needs to build a good classifier that generalizes to this known test set.

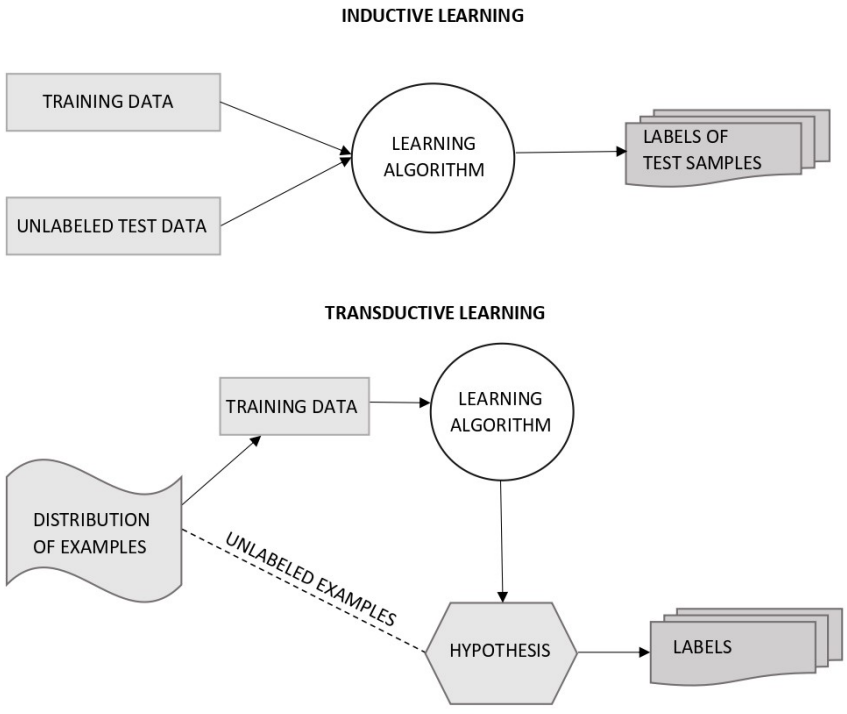

Figure 2.1: Inductive and Transductive learning

There are three techniques in machine learning for learning a concept from data. They are:

- Supervised Learning

- Unsupervised Learning
- Semi-Supervised Learning

In *Supervised Learning*, the training set consists of only labeled data. The goal is to learn a function that can generalize well on unseen data. The approach is called supervised because desired labels are given to the learner. Whereas in *Unsupervised Learning*, examples are presented to the system as observations without any label. There is no prior output. It uses methods that try to find the patterns of the data. The learner discovers patterns by itself. All the training data is labeled in supervised learning, and in unsupervised learning, none is labeled. However, in *Semi-Supervised Learning*, both labeled and unlabeled data are used. In contrast to labeled data, often expensive and time-consuming to gather, unlabeled data is usually easier to collect. So, it is common in semi-supervised learning to have a small amount of labeled data with a large amount of unlabeled data. Therefore, semi-supervised learning tries to find a better classifier from both labeled and unlabeled data.

Traditionally, in ML, it is assumed that the data is available. However, it is time-consuming and costly to gather the necessary data for training a classifier to a reasonable level of performance. In actuality, it is often the case that a small amount of labeled data is available and that more unlabeled data could be labeled on demand at a cost. If a process obtains the labeled data outside of the learner's control, then the learner is passive, which is called passive learning. If the learner picks the data to be labeled, then this becomes Active Learning (AL). AL has the advantage of picking data to gain specific information to speed up the learning process.

2.3.1 Embeddings

The order of the characters, size or the word root, is used to measure the similarity between words. However, when two texts are being compared, a specific representation of them is required to adequate tasks for a computer and not explode in terms of complexity.

A numerical-vector representation of texts is a good idea and is justified because:

- Computers work with the binary system at the bottom, but it can be said that a computer works with numbers by abstraction.

- The memory of computers can be seen as the primary vector.

An embedding is a low-dimensional continuous vector representation of a discrete variable, usually at a high-dimensional level. It acts as a mapper that represents variables in a transformed space. One of the most powerful features of embeddings is that they can be learned once and reused in other contexts, working similarly to a language model.

TYPE	BASE	NAME	DESCRIPTION
Non-Context Sensitive	Feed-Forward Neural Network	word2vec	Outputs one vector for each word.
		doc2vec	Extends word2vec to construct embeddings of variable-length texts.
		fastText	Extends word2vec to take into account subword information.
	Log-bilinear Model	GloVe	Combines the concepts of context window and text statistics in order to create word embeddings.
Context Sensitive	Long Short-Term Memory	Context2vec	Represents sentential context of target words as low dimensional continuous vectors. It uses Bidirectional LSTM.
		CoVe	Uses an LSTM encoder model trained for machine translation to contextualize word vector.
		ELMo	Uses Bidirectional LSTM and creates embeddings through the concatenation of the hidden states.
	Transformers	BERT	Based on Bidirectional Transformer and built using encoder blocks, it constitutes a language model that can be used in different tasks by adding a final layer. Different versions depend on the number of parameters, and various simplifications have been carried out, such as ALBERT and RoBERTa.
		CPT	Based on Bidirectional Transformer and built using decoder blocks. Different versions depend on the number of parameters and three possible versions.
		XLNet	Based on Transformer-XL, an evolution of Transformer can work with variable length context.

Figure 2.2: List of some Embeddings

With the publication of word2vec [57] in 2013, embeddings, and more concretely the neural network embeddings, suffered a significant advance due to the state-of-the-art performance of the vectors introduced. Later, other neural network embeddings were also introduced, creating a set of architectures, models and techniques that efficiently represent texts. These representations are fascinating in terms of text

comparison tasks. Some examples are GloVe, by Pennington et al. [69] in 2014 and fastText by Bojanowski et al. [6] in 2016. In 2014 doc2vec was introduced by Le and Mikolov [44], a natural extension of word2vec in order to use the same architecture, but for entire document instead of producing embeddings just for words. These models produce a vector for each word, representing vector (an embedding), the semantic and syntactic cached by the model. The efficiency reached by these embeddings in training time with large datasets, and the obtained quality described by the authors, motivate the emergence of new similarity measures.

The embeddings available in the literature can be grouped, as general as possible as follows: Machine Learning (ML) Embeddings, which use ML techniques to produce vectors; and non-Machine Learning Embeddings, which collect a large set of approaches. For non-ML Embeddings, the most popular and easy-to-understand model is Bag of Words (BOW), which provides vectors that count the apparitions of words in a text.

The ML Embeddings group is divided into non-context sensitive and context-sensitive techniques, depending on the possibilities that these provide to understand the context when learning vectors representing texts.

Other machine learning approaches like encoders have been intensively explored, for example, InferSent [9] that embed sentences using Bi-direction LSTMs, or the Universal Sentence Encoder [8] that runs on a Deep Averaging Network. Such options and other possibilities have been discarded because they focus on small-size texts or exceed this project's scope.

Embedding word2vec

The embedding word2vec (w2v), introduced by Mikolov et al. [58] from Google, is the name for two different techniques that create word embeddings. Before introducing w2v in 2013, the two mainly used approaches to create word embeddings were the Feedforward Neural Network (FNN) and the Recurrent Neural Network (RNN), which are hard to train in terms of complexity. The authors analyzed both and concluded that most of the complexity is caused by the non-linear hidden layer.

The new approaches follow the idea introduced by Mikolov et al., which consists of training network language models in two steps:

- Learning word vectors using a simple model.
- Training N-gram Neural Network Language Models (NNLM) on the top of the representation of words.

As this is the most attractive aspect of neural networks, the authors propose simpler models that cannot represent data like FNNs and RNNs but can be trained on much more efficiently. The log-linear model architectures introduced are the Continuous Bad-Of-Words Model (CBOW) and the Continuous Skip-gram Model (CSG); both models follow the same idea of predicting words based on other words.

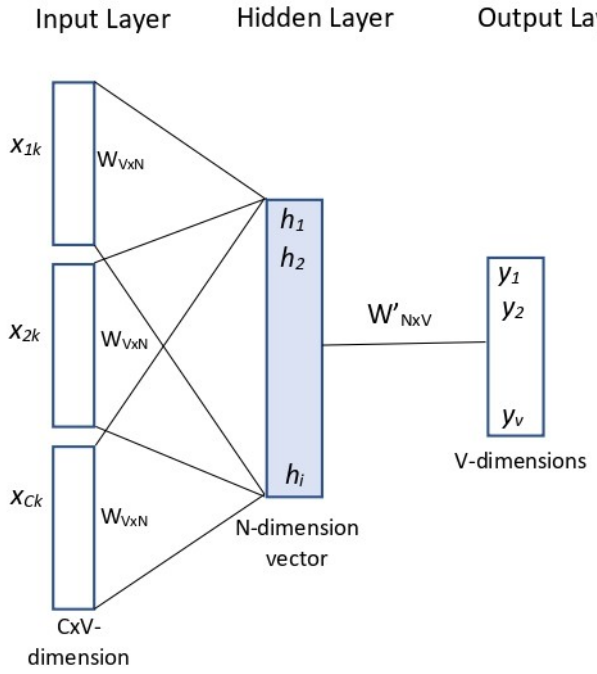

Figure 2.3: CBOW word2vec Model

The CBOW model predicts a current word based on the context (a given window of words around the one to be predicted). All the input words share a projection layer W, shown in the Figure 2.3, being projected in the same position, making an average of their vectors. As all the vectors are averaged, the position of the context words does not influence them. For the output, a softmax is applied.

For this case, the authors found that the best context window seems to be four history words and four future words, i.e., four previous and four following words from

the word to be predicted.

In contrast, the CSG model predicts the context for a current word, i.e., the architecture tries to predict a window of words around a known word. In Figure 2.4, it can be seen that the architecture consists of an input layer, one-hot encoder, a hidden layer that represents the input word vector using the input weights, and the output context vectors. Each output vector has softmax applied.

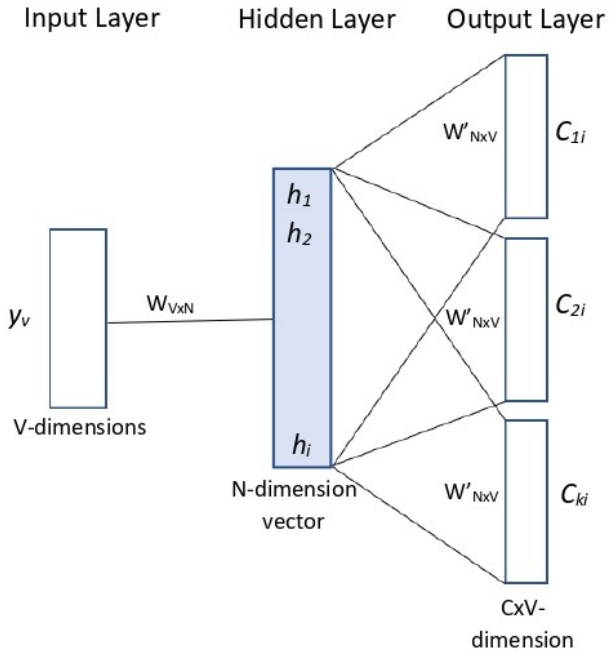

Figure 2.4: CSG word2vec Model

The authors found that for the CSG model, the window size increases, the quality of the resulting vectors increases too, but so does the complexity. Since the relation of the words inside the window tends to be lesser when the distance from the middle word increase, the model weights all the words accordingly during the training.

The w2v methods established a new approach in the word vectors creation. On the original paper, the authors conclude that both models outperform in many cases the previous stat-of-the-art models, being syntactically more advanced the CBOW model, and semantically more advanced the CSG model.

The authors found that less-dimension vectors, 300 for example, trained in more

data, can catch more accuracy than high-dimension vectors trained in fewer data. They also found that the training on twice the data during one epoch achieves better results than iterating over the same data during three epochs, making these models interesting to be trained on large amounts of data.

As an example of the power of w2v models, Mikolov et al. show that w2v vectors learn recoverable relationships. An example, performing a subtraction of two vectors and adding the result to another word vector, is:

$$\text{vec}(\text{“New Delhi”}) - \text{vec}(\text{“India”}) + \text{vec}(\text{“Canada”}) = \text{vec}(\text{“Ottawa”})$$

Embedding doc2vec

The doc2vec(d2v), embeddings were introduced by Le and Mikolov from Google [44]. It is also known as Paragraph Vector, and it extends w2v models to use the same base algorithm to learn document vectors and make the predictions more accurate using their information.

As in w2v models, the proposed d2v models map every word to a unique vector, a column in a matrix W, but they also map every sentence or large text to another unique vector, a column in a matrix D. In d2v, the hidden layer is computed by averaging the matrices W and D instead of only the matrix W, as in w2v.

To trace and extend the w2v models, the authors present two d2v models: the Distributed Memory Model of Paragraph Vectors (PV-DM), which is based on CBOW, and the Distributed Bag of Words Paragraph Vector (PV-DBOW), which is based on Continuous Skip-gram Model.

Using the proposed models in Sentiment Analysis and Information Retrieval tasks, the authors found that both outperformed in error rate, all the baselines compared. PV-DBOW performs worse than PV-DM due to its inherited complexities, but the authors suggest combining both, which obtains better results than in a separated usage. A range from 5 to 12 is proposed for the window size.

The complexity of doc2vec can be expected to be the same as for word2vec as it is all reduced to vectors, and the work process is the same as for word2vec but with an extra vector. Moreover, the training might be slightly worse due to the non-fixed-size text processing.

2.3.2 Cosine Similarity

Cosine Similarity (CS) measures the similarity between two vectors. It is the cosine of the angle between two non-zero vectors in a multi-dimensional space and determines whether two vectors are similar or not. If the cosine angle is small, the cosine similarity is higher and vice versa. It is used to measure the similarity between documents or texts, irrespective of their size, for analysis. Cosine similarity can be beneficial even if the two documents or texts are far apart by Euclidean Distance. A sentence can be represented in vectors by recording the frequency of words or phrases. These are referred to as "term-frequency vectors." Term-frequency vectors are long and sparse, containing many 0 values. These are used for information retrieval, gene feature mapping and biological taxonomy.

Cosine measure lies between the ranges of -1 and 1. -1 means that vectors are exactly opposite, 1 means that the vectors are similar and 0 indicates that the vectors are orthogonal. Other possible values within this range indicate similarity or dissimilarity. However, in text comparison, all components of the vectors are usually positive, i.e., they are in the first quadrant, and the cosine values are between 0 and 1, 0 meaning orthogonal or very dissimilar, and 1 and close to 1 very similar.

CS has a low-complexity of $O(n)$, where n is the vector dimensionality. In sparse vector space, only non-zero dimensions are considered to speed up the computation.

Given two non-zero vectors, A and B, the cosine between them can be derived from the Euclidean dot product:

$$A \cdot B = ||A|| \cdot ||B|| \cdot \cos\theta \tag{2.1}$$

that is,

$$\cos\theta = \frac{A \cdot B}{||A|| \cdot ||B||}. \tag{2.2}$$

Therefore,

$$\cos\theta = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.3}$$

where A_i and B_i are components of vectors A and B. Thus, the similarity between two non-zero vectors A and B is the cosine of the angle between them:

$$\text{sim}(A, B) = \cos\theta_{A,B} \tag{2.4}$$

Cosine similarity measure has been popularly used in text classification, speaker verification and information retrieval. The performance of the cosine similarity measure has been investigated for the task of Arabic text classification by Al-Anzi et al. [2]. Further, the experiments showed that the performance of the cosine similarity measure is comparable to that of SVM and KNN. Cosine Similarity measure along with tf-idf has also been used to evaluate the similarity in documents by Lahitani et al. [41]. The experiments were performed to determine the similarity level in Indonesian essay assessments. Results were sorted based on similarity levels of the documents to that of an expert's document, and it was established that the simplicity of the cosine similarity measure accelerates essay category correction.

Due to the popularity and simplicity of cosine similarity, it is also used along with Neural Networks [102]. Using Cosine measure with neural network structure for information retrieval and ranking the documents. Further, the results pointed to the improved retrieval performance of NN when coupled with cosine similarity.

Cosine similarity is also used for the task of speaker verification. Shum et al. [89] utilized the speed and convenience of the cosine similarity metric to develop an unsupervised algorithm. The algorithm takes advantage of the simplicity of cosine similarity scoring and achieves state-of-the-art results, and tackles the problem of unsupervised speaker adaptation.

Cosine Similarity has been also extended as Soft Cosine Similarity (SCS). SCS was introduced by Sidorov et al. [90] in 2014. The objects are represented as vector values of features in the Vector Space Model (VSM), and each feature corresponds to a dimension in the VSM. In traditional CS, there is no relation between the features. SCS is a generalized form of the (Hard) Cosine Similarity.

2.3.3 Word Mover's Distance

The Word Mover's Distance (WMD) is a distance measure derived from the Earth Mover's Distance, also known as the Wasserstein metric [40]. WMD was introduced by Kusner et al. in 2015. It is a distance function between text documents and it works with any feature vectors. Moreover, for the specific use in NLP, WMD tries to consider the relation or distance between individual words when comparing text documents. This measure uses the property of word embeddings to preserve

the semantic relations in vector operations. WMD represents text documents as weighted point clouds of embedded words. It measures the distance between two documents as the minimum cumulative distance that words from document A need to travel to match document B. Specifically, WMD reduces the problem of distance to the problem of transportation.

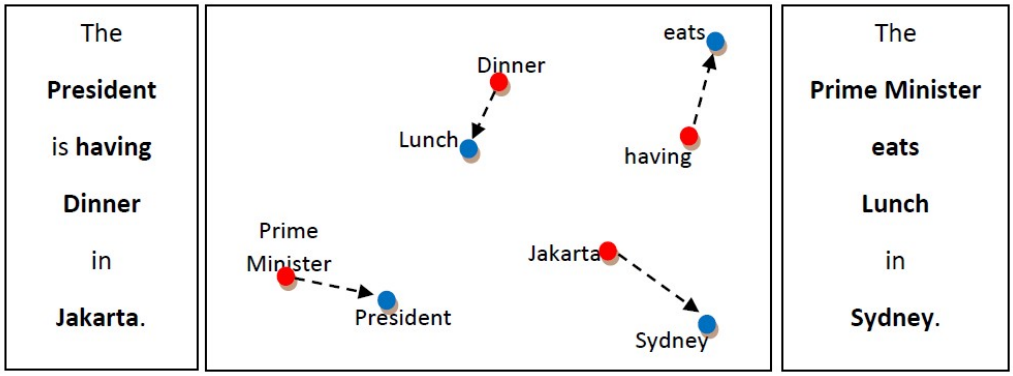

Figure 2.5: Illustration of the WMD for the sentences "The President is having dinner in Jakarta" and "The Prime Minister eats lunch in Sydney".

To measure the distance between two text documents, it is required:

- A word embedding matrix $X \in R^{d \times n}$ for a finite size vocabulary of n words.
- Texts represented as Normalized Bag of Words *(nBOW)* vectors $d \in R^n$.

The transportation problem for documents d and d' is formulated as follow:

$$\min_{T \geq 0} \sum_{i,j=1}^{n} T_{ij} c(i,j) \tag{2.5}$$

subject to,

$$\sum_{j=1}^{n} T_{ij} = d_i \quad \forall_i \in \{1, \ldots, n\}$$

and,

$$\sum_{i=1}^{n} T_{ij} = d_j \quad \forall_j \in \{1, \ldots, n\}.$$

In the above formula, $c(i,j)$ factor is the similarity between the words, i.e, similarity between the word embeddings of the words i and j. That similarity is measured

with the Euclidean distance and represents the cost associated with traveling from one word to another:

$$c(i, j) = \|x_i - x_j\|^2 \tag{2.6}$$

The WMD calculates the cost of transporting the words from one document to another. The flow matrix $T \in R(n \times n)$ represents the cost of travelling from a word to the rest to calculate the distance.

The two restrictions in WMD ensure that all the flow is equal to d_i, i.e., the word count in the document and that the flow is bidirectional.

Though WMD is powerful, it has a high time complexity. The authors proposed a relaxed version of the Word Mover's Distance (RWMD) apart from WMD. The difference from WMD lies in the elimination of one of the two constraints [40].

WMD has been used for document similarity, topic modelling and knowledge retrieval. Tashu and Tomas [95] proposed a method for Automatic Essay Evaluation (AEE) using WMD. The proposed method measures the distance between individual words from the reference solution and a student's answer. It uses the skip-gram model of word2vec to obtain word embedding.

Nasir et al. [63] proposed a model that uses WMD to capture different features of linguistic coordination in dynamic conversations potentially. The proposed method is also helpful in capturing interpersonal behavioural information.

Various methods have been proposed that extend WMD. A method that extends WMD to character 3-gram embedding, proposed by Oguni et al. [65]. This method enables character 3-gram Mover's Distance to be utilized. WMD is unsupervised, and its uses are not limited to any of the tasks.

Deudon [15] proposed an algorithm that incorporated supervision into the WMD and hence, can be used in tasks such as news article classification based on sentiments or topics. The Variational Siamese Network extends WMD for continuous representation of sentences. The proposed extended model performed strongly on the Quora question pair dataset and proved effective on question-retrieval in the knowledge database.

Huang et al. [28] proposed an efficient technique that uses WMD to learn a supervised metric. This metric is called Supervised Word Mover's Distance (S-WMD).

Wu and Li [103] present an approach called Topic Mover's Distance (TMD) for

documents inspired by WMD. TMD considers that the documents are composed of predefined topics, and a cluster of words denotes these topics. These clusters are then expanded in vector space. TMD measures how far topics need to travel from one document to another.

2.3.4 BERT

One year after the introduction of transformer architecture, a significant breakthrough in performance was made by the natural language representation model BERT, Bidirectional Encoder Representations from Transformers. Devlin et al. [16] extended Transfer Learning and Transformer techniques to create this model to pretrain word representations bidirectionally to get a better understanding of the impact how all words have on the context. It is well known that a word's placement at different position in the sentence changes its POS tagging. BERT was trained on general English corpus and fine-tuned on NLP tasks like next sentence prediction and Masked Language Modelling (MLM) [47]. BERT can be fine-tuned with an additional output layer to achieve state-of-the-art results on a wide range of tasks.

BERT is published in two sizes, BERT-Base and BERT-Large. BERT-Base has 110 million parameters, and large has 340 million parameters. Due to the considerable memory requirements and size, these models require specialized hardware for training. BERT-Large is built up with 16 attention heads and 24 encoder layers. In contrast, BERT-Base has 12 attention heads and 12 encoder layers. The two models use different hidden space sizes, where the BERT-Larger model has 1024 dimensions, and the BERT-Base model has 768 dimensions.

Transformers used by BERT ensure to give attention to words or phrases which are more important than others. The absence of such words or phrases could increase the sentence's ambiguity and is used as a hallmark of the phrase by BERT to determine a word's importance. Transformers look at the target word and understand the context of all the other words related to the original word. The target word can be focused, and the related phrases can be linked using the transformers' attention mechanism. It also takes care of the polysemous words by allocating weights to the words related to the target word. Every related word is given a weight based on the meaning they add to the target word. A sentence describing the word "bank"

will be associated with the term “river”, making it straightforward for the model to understand that it deals with nature and not the financial institution.

BERT uses MLM to ensure that the entire focus is not on the target word and creates no imbalance. MLM randomly masks a word and tries to predict the hidden word. Textual entailment or next sentence prediction is a training process that involves the pairing of sentences. The pairs can be right or wrong. For training, the model identifies the pairs if they are right or wrong, based on which the model gets a prediction score. Training the model helps BERT understand the context at the sentence level and is beneficial for Natural Language Inferencing. BERT stores information about a sentence in a unique token represented as [CLS], called a classification token.

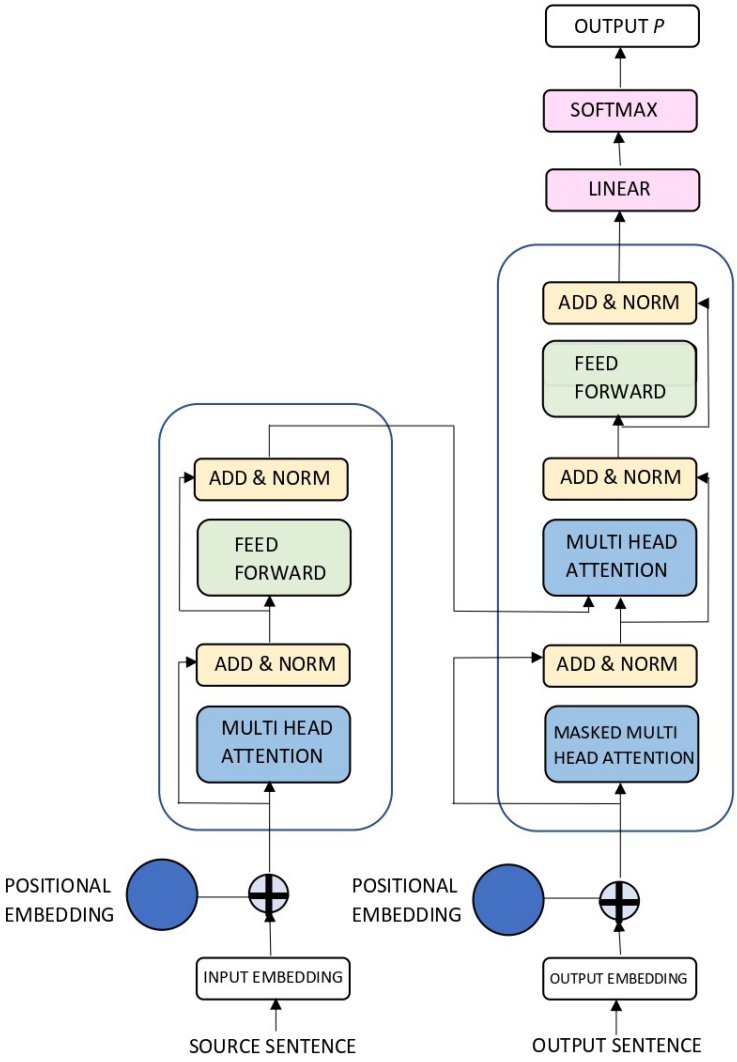

Figure 2.6: Transformer model architecture. [98]

The architecture of BERT is shown in the Figure 2.6. Both the encoder and the decoder have three Multi-Head Attention layers. The attention is defined as follow:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.7}$$

where Q is the query matrix; K is the key matrix; V is the value matrix from the attention input. Dot-product attention may have different normalization scale. The above equation uses the square root of key dimensions as the scale to normalize the compatibility/alignment score. This operation consists in the softmax-value of the normalized dot product of queries and keys. The Multi-Head Attention is the concatenation of different attention layers.

Since the release of BERT, several variants and extensions based on BERT have been proposed. RoBERTa [51] is an extension of BERT which tries to improve BERT model training. Another modification of BERT is ALBERT [42] which tries to compress the size of BERT and, at the same time, outperforms BERT. XLNet [104] is another BERT-like model that adopts a generalized auto-regressive pre-training method.

A BERT-based technique, BAE (BERT-based Adversarial Examples), for replacing words to fit the English language's overall context better, was proposed by Garg and Ramakrishnan [24]. In addition to replacing words, BAE also inserted new tokens in the sentence to improve its strength.

tBERT, an architecture for semantic similarity detection, which incorporates Topic Models and BERT and works better prominently on domain-specific cases was introduced by Peinelt et al. [68].

RecoBERT is a model build upon BERT for self-supervised pre-training of a catalogue-based language model [52]. In RecoBERT, the BERT model is adapted for a textual-based recommendation. It achieves self-supervision by utilizing a combination of MLM along with a title-description model.

Another model, Sentence-BERT (SBERT) [75], is a modification of BERT that uses Siamese and Triplet network structures to derive semantically meaningful sentence embeddings, which can be compared using cosine-similarity.

Mutinda et al. [62] used a BERT-based approach to capture the semantic similarity between texts on Japanese clinical datasets. They compared the clinical BERT model, pretrained on Japanese clinical text, and general Japanese BERT model, pretrained on Japanese Wikipedia texts. Unexpectedly, general Japanese BERT, which was pretrained on a wide range of texts, outperformed the clinical Japanese BERT on clinical dataset.

2.3.5 Hierarchical Clustering

Clustering methods partition objects into groups so that the objects in one group are similar and dissimilar from those in other groups. These objects are usually represented by multi-dimensional variables, known as features or attributes.

Hierarchical clustering algorithms build a hierarchy of clusters [30]. It starts with

some initial clusters and gradually converges to one cluster. Hierarchical clustering has two categories: agglomerative and divisive. The agglomerative approach takes each data point as an individual cluster and iteratively merges the clusters until a final cluster containing all data points is formed. It is also called the bottom-up approach based on how it merges the clusters. Divisive clustering is the opposite of agglomerative clustering and follows the top-down flow, which starts from a single cluster having all data points and iteratively splits the cluster into smaller ones until each cluster contains one data point.

Agglomerative hierarchical clustering on a set of n data points begins with a symmetric $n \times n$ distance matrix consisting of pair-wise distances between the data points, and the following steps of the clustering algorithm are followed:

1. The algorithm begins by assigning each data point to a separate cluster to obtain n clusters, each containing one data point.

2. To find the closest pair of clusters, it computes the similarity (distance) between them.

3. Similar clusters are merged to form a cluster according to the distance function.

4. Steps 2 and 3 are repeated until all data points are merged into one last cluster.

In addition to measuring the distance between individual data points, a method to compute the distance between clusters in Step 2 is also needed to merge the two most similar clusters. Such a method is referred to as linkage. Let $X_1, X_2, X_3, ..., X_m$ be the observations from cluster u and $Y_1, Y_2, Y_3, .., Y_n$ be the observations from cluster v. And $d(X, Y)$ denotes the distance between vector X and vector Y. Then, the distance between u and v can be calculated using four linkage methods:

1. Single linkage (min)

$$d_{u,v} = \min_{i,j} d(X_i, Y_j)$$

   Single linkage means the distance between two clusters is the minimum distance between one point of the first cluster and another point of the second cluster.

2. Average linkage

$$d_{u,v} = \frac{1}{kl} \sum_{i=1}^{k} \sum_{j=1}^{l} d(X_i, Y_j)$$

Average linkage calculates the distance of all data points from the first cluster with all others from the second cluster and takes the average distance as the distance between the clusters.

3. Complete linkage

$$d_{u,v} = \max_{i,j} d(X_i, Y_j)$$

Complete linkage takes a maximum distance of two data points value as the distance between two clusters.

4. Ward (max) linkage

$$d_{u,v} = E(u, v) - [E(u) + E(v)]$$

where,

$$E(u) = \sum_{i=1}^{u} |u_i - \frac{1}{u} \sum_{j=1}^{u} x_j|^2$$

Ward is similar to average linkage except that it uses the error sum of squares, E, to calculate the distance between the points [101].

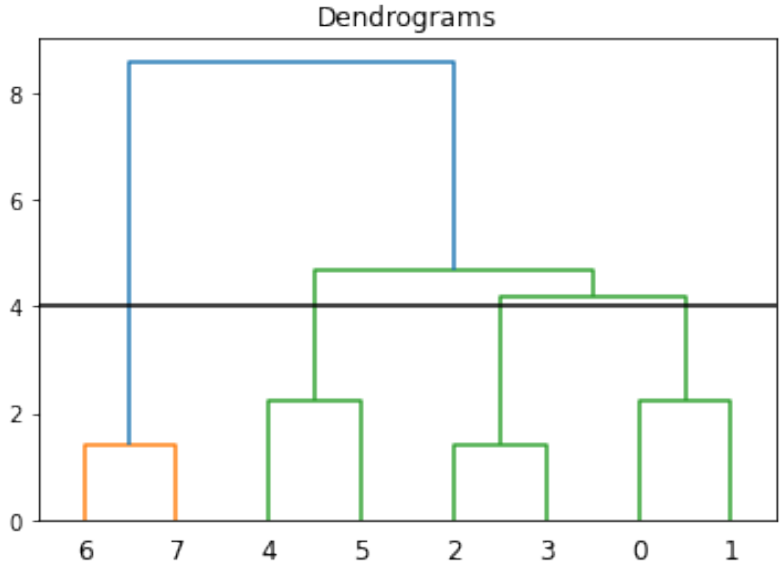

Figure 2.7: An example of dendrogram representing the clustering technique of hierarchical clustering algorithm.

A hierarchical clustering result can be visualized as a dendrogram, where inner nodes represent nested clusters with varying numbers of objects belonging to each cluster. In other words, a dendrogram hierarchically organizes clusters to provide a helpful summary of the data. The hierarchical clustering algorithm provides minimal guidance towards choosing the optimal number of clusters or the level at which to cut the dendrogram. Different decisions about dissimilarities and the cluster structure of interest can often lead to vastly different dendrograms. The Figure 2.7 displays a dendrogram representing a hierarchy of clusters, and how hierarchical clustering cut out the k clusters from the final cluster (complete tree).

2.4 Active Learning

The success or failure of supervised learning systems is largely determined by the training datasets used to train them. It is challenging to build a quality classifier without a good training dataset. Generating labeled examples for training a classifier is typically time-consuming and expensive as it involves experts to label the data. Fortunately, this is not an insurmountable problem. Active learning is a machine learning technique that can help reduce labelling efforts.

Active learning (AL) is an area of interest in machine learning, also referred to

as Query Learning or Optimal Experimental Design in statistics. AL is an iterative learning process that can build high-performance classifiers or label datasets from a larger unlabeled dataset. It strives to make learning algorithms more reliable with fewer annotations [85]. The fundamental assumption is that the algorithm can deliver comparable accuracies or performances with less training data if the training instances are informative, and hence the learning effort required is short and sweet. On the other hand, supervised learning requires hundreds and thousands of labeled instances to train the algorithms. Active learning first garnered serious research attention in the 1980s [3] and has remained a vibrant research area. Active learning is widely used in situations with vast amounts of unlabeled data, for example, image retrieval, natural language processing and text classification or where labeled training examples are expensive or time-consuming to obtain.

Active learning algorithms try to detect the most informative examples in the instance space X and ask the user to label only them. The examples that are chosen for labeling are called *queries*.

A typical semi-supervised algorithm proceeds as follows: First, it uses the base learner and a small labeled dataset, $\mathcal{L}$, to learn an initial hypothesis, h. Then h is applied to the unlabeled examples in $\mathcal{U}$, and some or all of these examples, together with the labels predicted by h, are added to $\mathcal{L}$. Finally, the entire process is repeated for a number of iterations.

2.4.1 Existing Active Learning Approaches

There are three primary forms of active learning literature considered in this section:

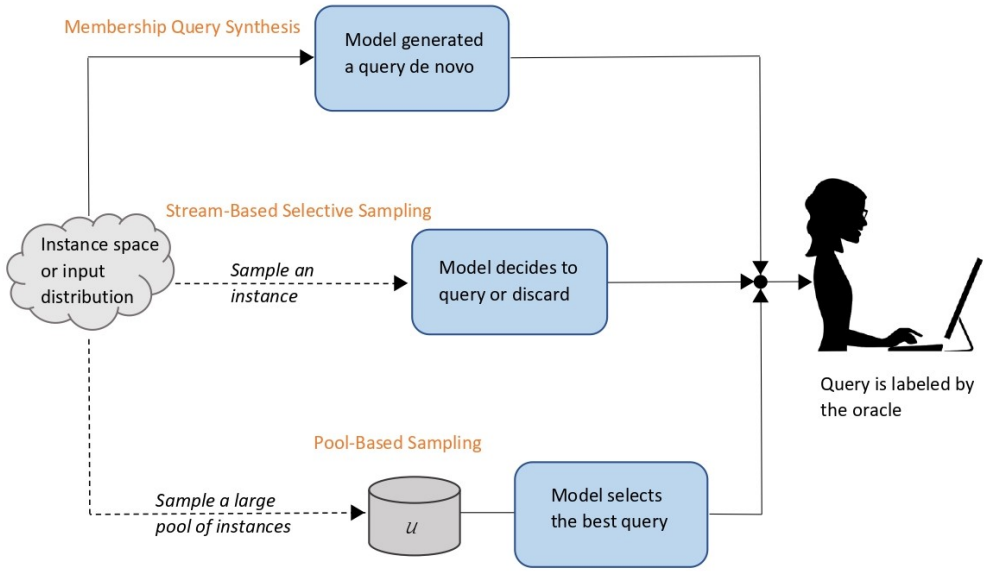

Figure 2.8: Illustration of three main active learning approaches

- Membership queries
- Stream-based
- Pool-based

Membership Queries

In AL with *membership queries*, the learner requests labels of examples in the input space, including ones the learner generates anew, from the oracle. This approach was initially involved in learning to identify an unknown concept drawn from a finite hypothesis space. Information about the unknown concept was gathered using queries. The main advantage of active learning with membership queries is that it can work in situations when unlabeled data is unavailable. This approach is almost impossible for text classification because the algorithm's documents are implausible examples with no meaningful contents.

Stream-Based Active Learning

In *stream-based* AL, queries are based on filtering a stream of unlabeled examples. The learner is given a stream of unlabeled examples, and it chooses whether or not

to ask the oracle for the labels of the examples. The advantages of stream-based active learning are that it can deal with complex and noisy data and can be used in dynamic and online learning scenarios. Sculley [84] investigated using a stream-based active learning method for spam filtering where the filter was exposed to a stream of messages. Stream-based approaches have the disadvantage that the learner cannot access all unlabeled examples when selecting the most informative examples.

Pool-Based Active Learning

The *pool-based approach* of AL is the most common for text classification since an extensive collection of text is available. In the pool-based approach, the learner has access to a large pool of unlabeled examples. Although it is one of the most common forms of AL, it cannot efficiently deal in a dynamically changing online environment.

A small set of examples, $\mathcal{L}$ labeled by an oracle, is used to initialize a selection strategy. The selection strategy assigns value to each example in the unlabeled pool, indicating how informative the example is and giving the most informative examples to the oracle for labelling. A batch size, b, determines the number of examples to be selected in each iteration. Although a smaller batch size leads to a sharper increase in performance, a larger batch size is considered more efficient. Once the examples are labeled, they are removed from the pool and added to the labeled examples. Furthermore, the informative values associated with each unlabeled sentence are updated in the pool. This process is repeated until all the examples are labeled or until some stopping criteria are met.

A framework of a generic pool-based active learning system is shown in the Figure 2.9. The resulting manually labeled dataset can train a classifier or infer the labels for the remaining unlabeled examples. Typically, the manually labeled set is used to build a classifier.

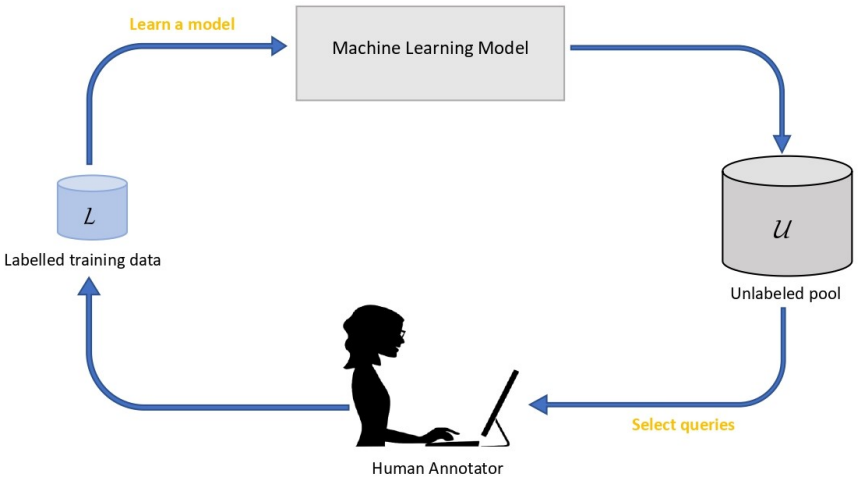

Figure 2.9: Pool based Active Learning cycle where $\boldsymbol{L}$ is the set of labeled data and $\boldsymbol{U}$ is the pool of unlabeled data.

There are three significant issues of concern in active learning:

- A technique is required to choose a small initial training set to seed the active learning process.
- A selection strategy is required to select the examples that will be labeled throughout the active learning process. These should be the examples for which labels will prove most informative as the training process progresses.
- Criteria must be established to determine when the active learning process should stop.

Active Learning first received research attention in the 1980s and has since remained a lively research area. Active Learning is widely used in situations where there are vast amounts of unlabeled data available, to name a few, image retrieval, natural language processing, text classification, bioinformatics and medical applications.

Active-Outlier, an approach for outlier detection was proposed by Abe et al. [1]. Active-Outlier uses a selective sampling mechanism based on Ensemble-based Minimum Margin Active Learning. Its outlier detection accuracy was reported high and

well suited in stream-mining setup. Active Learning has also been implemented in the field of robotics.

The Active Learning techniques have also been used to tune the behaviour and reliability capabilities of the grasping system for robot [61]. An algorithm, proposed by Singh et al. [91], uses Active Learning for time-series gene expression analysis. It uses Active Learning for the development of computable objective functions for measuring uncertainty in the estimated signal. The proposed algorithm can be applied to any continuous function with one independent variable.

Active Learning is used in the Domain of Natural Language Processing as well. Tur et al. proposed an algorithm that uses Active Learning for spoken language understanding [96]. The proposed algorithm used Boosting algorithm for call classification and achieved the same classification accuracy using less than half of the labeled data.

The first of its kind work, Active Learning for Named Entity Recognition (NER) was proposed by Shen et al. [87]. They proposed a multi-criteria-based AL approach along with SVM and effectively applied it to NER. They incorporated three criteria, informativeness, representation and diversity, using two selection strategies. The proposed approach reduced the labeling cost compared to the single criteria-based method.

2.4.2 Classifiers Used in this Research

This research mainly focus on using three types of classifiers for Active Learning:

- Support Vector Machine, a Margin classifier,
- Naïve Bayes, a Probabilistic classifier, and
- Random Forest

Support Vector Machine

The Support Vector Machine (SVM) is the most prominent approach to maximum margin classification, which is essentially specified by a separating hyperplane in the multi-dimensional input space R_k given by the feature representation $\overrightarrow{\mathrm{x}}$ of example x. The best separating hyperplane is the one that represents the most significant

separation — called margin. The distance from it to the nearest training example on each side is maximized. The hyperplane is defined in Equation 2.8.

$$< \overrightarrow{\mathrm{w}}, \overrightarrow{\mathrm{x}} > +b = 0 \tag{2.8}$$

The basic idea behind SVM is to find those examples (support vectors) that delimit the widest frontier between positive and negative examples in the feature space. Support vectors are examples that are closest to the hyperplane. The width of the classification border is known as the hyperplane margin. Equation 2.9 gives the SVM classifier.

$$f_{\overrightarrow{\mathrm{w}},b}(x) = sgn(< \overrightarrow{\mathrm{w}}, \overrightarrow{\mathrm{x}} > +b) \tag{2.9}$$

SVM classifiers can deal with non-linearly separable data by mapping the feature representations with a non-linear mapping function into a higher-dimensional feature space H, where the separability between examples may be more straightforward. This is done using a kernel function leading to a reformulation of the SVM classifier into Equation 2.10.

$$\Phi(x) = \Sigma_{i=1}^{s} \propto_i y_i K(x, w_i) + b \tag{2.10}$$

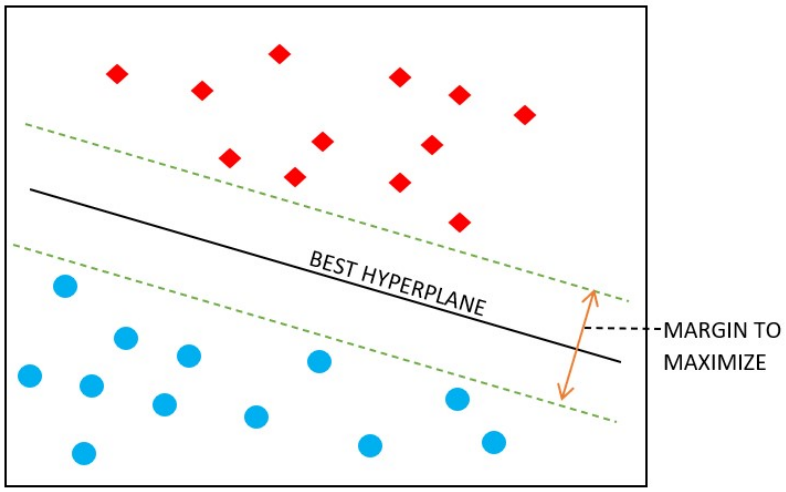

Figure 2.10: Illustration of SVM.

SVMs have been applied in many text classification tasks since most text classification problems are linearly separable, and SVMs are robust in high dimensional space and robust with sparse data.

Naïve Bayes

The Naïve Bayes classifier is a popular technique for text classification and has been found to perform surprisingly well despite its simplicity [43, 45, 54]. The Naïve Bayes classifier exists in different versions, such as Bernoulli, Multinomial and Gaussian.

The underlying theorem for Naïve Bayes classifiers is Bayes' Law as shown in Equation 2.11, which assumes that all features are conditionally independent. A probabilistic model that embodies the assumption is posited, and training examples are used to estimate the parameters of the proposed model. A new example is classified by selecting the class most likely to have generated the example.

$$P(x \mid y) = \frac{P(y)P(x \mid y)}{P(x)} \tag{2.11}$$

In a dataset, every document has the same probability, so $P(x)$ is a constant which can be eliminated from Eq. 2.11. Based on the Naïve Bayes assumption that all features are conditionally independent, the Naïve Bayes (NB) model is formulated in Eq. 2.12. This is used in text classification to determine the probability that document x is of class y just by looking at the frequencies of words in the document.

$$P(y \mid \overrightarrow{\mathrm{x}}) \propto P(y, \overrightarrow{\mathrm{x}}) = P(y)\Pi_{j=1}^{k} P(x_i \mid y) \tag{2.12}$$

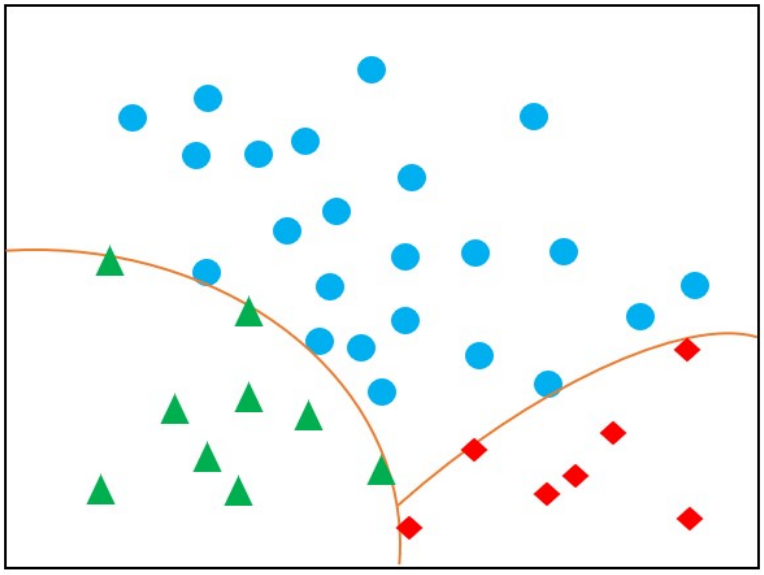

Figure 2.11: Illustration of Naïve Bayes

In the multi-variate Bernoulli model [33], an example is represented as a binary vector of term occurrences, and in the multinomial model, an example is represented as a vector of term counts [60]. Several works show that the multinomial model usually performs better than the multi-variate Bernoulli model [20, 54].

Random Forest

Breiman firstly introduced the Random Forest algorithm [7]. Decision trees can be extremely noisy when it comes to predictive performance. A lower correlation between the estimates can result in lower variability of the final prediction. A method to compensate for the extreme variability of an individual decision tree is bagging. The Random Forest Algorithm provides more precise estimates relative to the bagging of decision trees by reducing the correlation between the constructed trees, thus improving and reducing the variance.

Random Forest forms a family of predictive models that construct an ensemble of randomized decision trees. The randomness is injected in the training set and the decision tree learning procedure while maintaining the low bias of the individual models. It yields one of the most effective general-purpose predictive models.

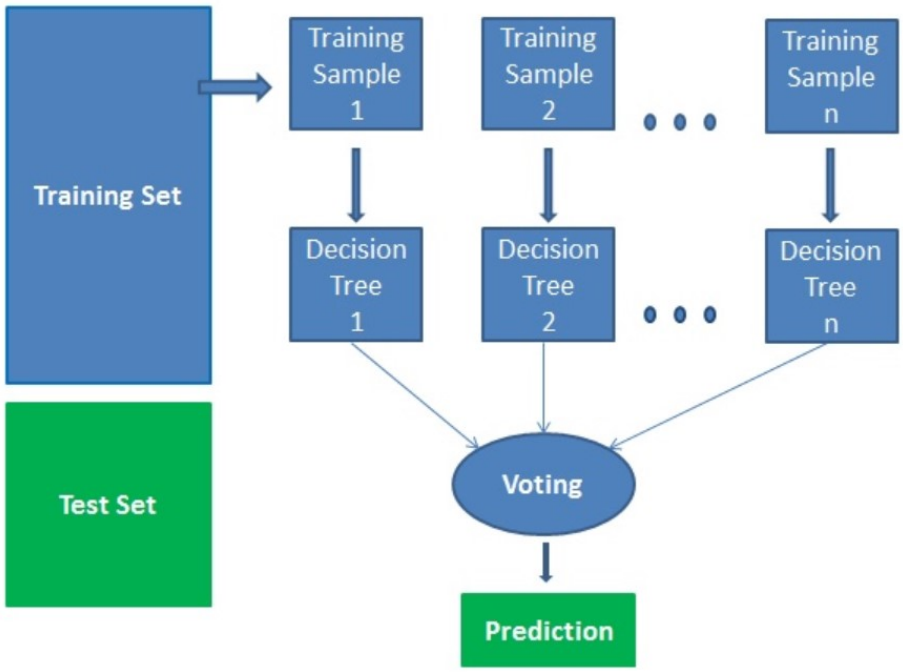

Figure 2.12: Illustration of Random Forest

The Random Forest algorithm can be used in solving both classification and regression problems. It is based on a divide and conquer approach performed on decision trees generated on random samples of a given data set. Attribute selection indicators like the Gini index, Information Gain and Gain Ratio are used for generating individual decision trees.

For a regression task, the average of all tree results is considered as the final output. The final result is based on the voting from individual decision trees in the classification task.

Random forests have a wide variety of applications. It can be used in recommendation engines, image classification, feature selection, identifying fraudulent activities and predicting diseases.

2.4.3 Selection Strategy

The essence of Active Learning is a strategy for selecting the following query to be presented to the oracle for annotation. Active Learning Selection strategies can be categorized into three approaches:

- Exploitation based selection strategies
- Exploration based selection strategies
- Strategies that use a combination of both exploitation and exploration.

Exploitation-based selection strategies build a classifier using those examples labeled by the oracle so far in the active learning process and base the selection of examples for labelling on the output generated by this classifier when used to classify all of those examples remaining in the unlabeled pool.

Exploration-based selection strategies pick representative examples from dense regions of the example space instead of focusing on examples closest to the classification boundary. Exploration-based selection strategies also favour examples distant from the current labeled set with the aim of sampling more expansive, potentially more exciting areas of the feature space. These approaches do not necessarily use a classifier in active learning selection.

In this research, an exploitation-based selection strategy is used.

2.4.4 Stopping Criterion

A stopping criterion is used to decide when to stop the active learning process. In most cases, a simple stopping criterion that allows the oracle to provide a specified number of labels, a label budget, is used. Other approaches, referred to as hold-out accuracy approaches, stop when the classifier's performance is built reaches some target performance on a hold-out test set. However, stopping criteria such as those that use the classifier's characteristics are preferable due to the difficulty of getting labeled examples because they do not require a hold-out test set. Researchers have also proposed confidence-based stopping criteria that suggest stopping the active learning process based on measuring the classifier confidence. In some cases, the active learning process is stopped when the pool is empty.

2.5 Chapter Summary

This chapter discussed the previous work related to the CLSA project and on detecting no-opinion responses. This chapter also explains the state-of-the-art Machine Learning approaches used in this thesis. Moreover, the related literature survey is also discussed in this chapter. The embeddings and existing Active Learning Approaches are also presented along with the Classifiers used in this research.

Chapter 3

Methodology

This chapter elaborates on the data collection and the dataset used for the experiments in this research. It proceeds to discuss preprocessing of data before feeding it to the classifiers. Next, the embeddings used in the experiments are discussed. Furthermore, it discusses the clustering algorithm and classifiers used in the experiments along with the metrics for evaluating the results. It also discusses the methodology and explains the implementation of the models.

3.1 CLSA

The Canadian Longitudinal Study on Aging (CLSA), an initiative of the Canadian Institute of Health Research (CIHR), was launched in 2009 in collaboration with Statistics Canada. The main objective of the CLSA is to understand the variety of factors that influence aging and study the transitions and trajectories of healthy aging among Canadians [74].

The CLSA is a nationally stratified sample of 51,338 women and men aged between 45 and 85 years of age who will be contacted every three years and will be followed up for at least 20 years. The CLSA cohort comprises two corresponding cohorts. A "tracking cohort" of 21,241 participants was randomly selected from the ten provinces and interviewed by telephone. Furthermore, a "comprehensive cohort" of 30,097 participants was randomly selected from within a 25–50km radius of 11 data collection sites (DCSs) interviewed in person, took part in in-depth physical assessments at DCSs, and provided blood urine samples.

CLSA is conducting a longitudinal study on aging, i.e., it will observe the same variables for an extended period to find behavioural patterns to understand the aging process among older Canadians. Furthermore, the data collected by the CLSA will also help understand why some people age well and others do not, how to improve health services and policies for Canadians and how non-medical factors like social

and economic status affect aging [74].

3.2 Dataset and Characteristics

The identity of the participants is kept confidential by assigning a unique identification number to protect their privacy. The data collected in the Tracking and Comprehensive cohorts via telephone and in-person interviews are stored in CSV files made available to the researchers. In CSV files, the questions are converted to columns, and participant's responses are recorded, as shown in Table 3.1.

entity_id	**AGE_ NMBR_ TRM**	**SEX_ ASK_ TRM**	**SDC_ COB_ TRM**	**GEN_HLAG_TRM**
17724724	46	F	1	exercise and proper diet
49119706	61	F	1	social, mental and generally good health
58884735	57	F	1	positive attitude, eating well and exercise
68179218	66	F	2	eating good food, exercise, being social, knowledge of what you're eating and drinking

Table 3.1: Sample data from CLSA dataset

Apart from this, there are two other files, Dictionary and Dictionary category. The Dictionary file contains details like data types, labels that give information on the answer, comments that might be useful for researchers and questions asked in the interview, as in Table 3.2. The Dictionary Category file contains the answers that are converted into numerical data for ease and their explanation as in Table 3.3.

Name:	startlanguage_COM
Label:	Language used at start of interview
Comment:	
Question:	
Name:	AGE_NMBR_TRM
Label:	Age(years)
Comment:	*Calculated: Date of interview less reported Date of Birth. The few cases of ages outside the study population range (45-85) are due to time lapse issues between the initial recruitment stage and the actual date the interview was completed.*
Question:	What is your age?
Name:	SEX_ASK_TRM
Label:	Sex
Comment:	
Question:	Are you male or female?
Name:	SDC_COB_TRM
Label:	Country of birth
Comment:	*Includes additional categories based on open text responses of other countries of birth variable.*
Question:	In what country were you born?
Name:	SDC_RELG_TRM
Label:	Religion
Comment:	*Includes additional categories based on open text responses of other religions variable.*
Question:	What, if any, is your religion?
Name:	GEN_HLAG_COM
Label:	Promote healthy aging verbatim
Comment:	
Question:	I have talked with many adults and learned something from each of them about what they think promotes healthy aging. What do you think makes people live long and keep well?

Table 3.2: Sample data from Dictonary

VARIABLE	NAME	MISSING	LABEL
startlanguaga_TRM	EN	0	English
	FR	0	French
SEX_ASK_TRM	M	0	Male
	F	0	Female
SDC_COB_TRM	1	0	Canada
	2	0	United Kingdom
	777	1	Missing

Table 3.3: Sample data from Dictionary Category

The data collected by the CLSA are generalizable to the comparable Canadian population on many vital variables. It was designed to help understand the contribution of biological, clinical, lifestyle and behaviour and social measures in aging adults in Canada. The CLSA interviewed 51,338 women and men aged between 45 and 85 years for this study. Of all the participants, 21,242 participants from Tracking Cohort took telephone interviews that lasted 60–90 minutes. Furthermore, the remaining 30,097 participants were interviewed in person, and the interview lasted roughly 90 minutes. Of the 30,097 participants, 27,170 (90.3%) and 28,783 (95.6%) provided blood and urine samples, respectively.

Several critical multidisciplinary issues for understanding the aging process were considered, focusing on questions that could only be answered with a longitudinal design. Out of all the participants, 51.5% were female, and most participants were born in Canada, accounting for 87.2%, and 80.2% spoke English at home. 38.5% had university degrees, 83.7% of participants lived in detached or semi-detached houses, and almost 70% were married or living with a common-law partner.

The questionnaire-based measure used to collect the data covers many domains, including health status, lifestyle and behaviour and health care utilization. However, the essential question was, “I have talked with many adults and learned something from each of them about what they think promotes healthy aging. What do you think makes people live long and keep well?”. The verbatim response of older Canadians to this question is used for this work.

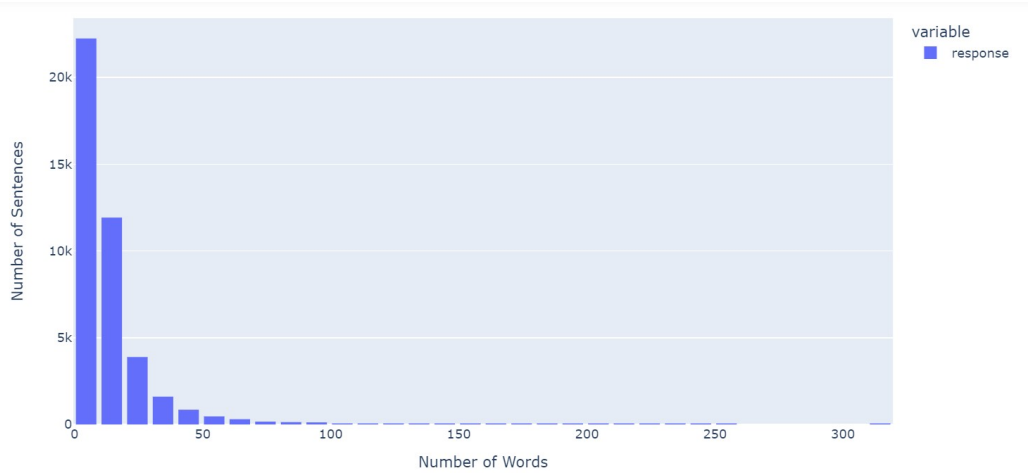

Figure 3.1: Number of words in sentences

The number of words used in responses is shown in the Figure 3.1. The 50,587 participants responded to the question using a maximum of 319 words and a minimum of one word, with a mean of 12.68 and median of 9. The responses are unstructured and have many typographical errors.

The Figure 3.2 shows the vocabulary in a word cloud with the most frequent words in the CLSA dataset, i.e., participants' responses. One interesting point, which can be appreciated, is that some of the most common words are exercise, good, diet, eating, active, healthy, being, keeping, positive, attitude and life.

Figure 3.2: Most common words in the CLSA dataset.

In addition, the CLSA also provided a "no_answer_instances" dataset which was manually selected. It consisted of 71 unique no-opinion responses from the CLSA

dataset. In this thesis, this is refered to as "no-opinion" dataset. A sample of no-opinion dataset is provided in the Table 3.4 below:

Entity Id	Text	Language
39197483	98	en
53723716	i have no idea	en
37844084	don't know	en
66279744	i'm not sure	en

Table 3.4: Sample data from No-Opinion Dataset.

3.3 Preprocessing

Data preprocessing can have a notable influence on the generalization performance of ML algorithms. Preprocessing data is intended to transform the raw data into a more accessible and more effective format for future processing steps. Before building a model, data preprocessing is necessary and consists of three steps: Data Cleaning, Data Transformation and Data Reduction. Raw data is often incomplete and inconsistent due to the human factor, program errors, or other reasons. Incomplete and inconsistent data will affect the accuracy of the predictions, so before going any further with the database, we need to do data cleaning.

Before cleaning the data, the telephonic and in-home interview data were concatenated to get one dataset. In the first step of data cleaning, missing values and inconsistencies were removed. We only considered English responses for this project, which constitutes 82.7% of the CLSA dataset, so French response, 17.3% of the CLSA dataset, were also removed in the cleaning process. Some responses had their language tags misplaced; some English responses were mislabeled as French responses. These labels were corrected, and the English responses were considered.

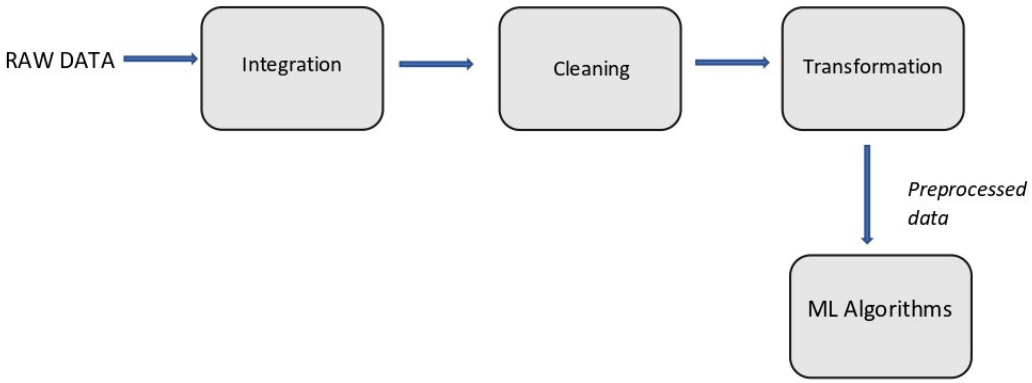

Figure 3.3: Data Preprocessing

Most text and document data sets contain unnecessary words such as stopwords and misspelling. In many algorithms, primarily statistical and probabilistic learning algorithms, noise and unnecessary features can adversely affect system performance. In the next stage, Data Transformation, responses were normalized and tokenized. Tokenization is a preprocessing method that breaks a stream of text into words, phrases, symbols, or other meaningful elements called tokens [26, 99]. For example, consider the sentence:

"I took the dog for a walk."

In this case, tokenized sentence is :

"I" "took" "the" "dog" "for" "a" "walk".

After tokenization, punctuations and stopwords were removed. Textual datasets include many words which are not significant, such as "a", "have", "what", "do", . . . , and can be removed from the texts [79]. Some words indicated if the response was a no-opinion response or no, were removed from the set of stopwords; for example, "not", "don't", and "can't". Most text datasets also contain many unnecessary characters, such as punctuation and special characters. Critical punctuation and special characters are essential for human understanding, but they can be detrimental to machine learning algorithms [67].

Words were then converted to their root form by WordNetLemmatizer. Lemmatization is an NLP process that replaces the suffix of a word with a different one or deletes the suffix of a word to get the basic word form, i.e., lemma [82, 34].

For the final preprocessing step, words were converted to their vectorized forms and stored as TF-IDF vectors. Jones [31] proposed that Inverse Document Frequency (IDF) be used with term frequency to reduce the effect of common words in the corpus. The combination of TF and IDF is known as Term Frequency-Inverse Document Frequency (TF-IDF). It assigns a higher weight to words with high or low frequencies term in the text data. The mathematical representation of the weight of a term by TF-IDF is given in equation 3.1.

$$W(d,t) = TF(d,t) \times \log\left(\frac{N}{df(t)}\right) \tag{3.1}$$

Where N is the number of documents and $df(t)$ is the number of documents containing the term t in the corpus.

After preprocessing, these 41,000 records were used for the experiments.

3.4 Unsupervised Methods

This research aims to analyze and compare different similarity measures, as has been mentioned several times. Because of this, the implementation of the measures takes significant relevance.

The objective of this research has been to analyse and compare different text similarity measures combined with Machine Learning embeddings. Measures such as CS, WMD and BERT have been analyzed. CS works directly with vectors (or embeddings); WMD applies information from embeddings to represent text weights and treating the distances as a Linear Programming Transportation Problem.

The diverse set of measures requires text representations according to their ways of working. In order to achieve this, measures which work with information from embeddings have been combined with word embeddings, word2vec, doc2vec and BERT.

The tasks performed in this project have brought to light some interesting points about the different measures. From a theoretical point of view, i.e., the properties required of the measures, it has been seen that the more general the text measure

(and embedding) is, not necessarily the better it is. This is the case for CS + doc2vec, which are clearly at a disadvantage whereas, BERT gave better results.

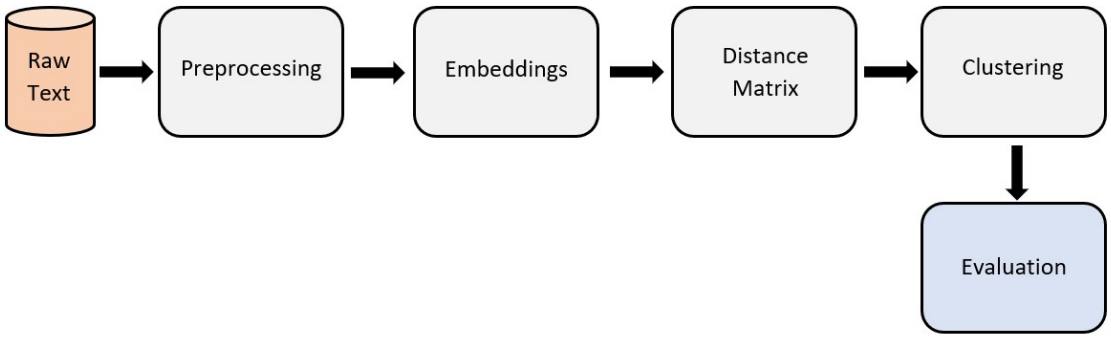

Figure 3.4: Experimental Procedure for Unsupervised Learning.

The Figure 3.4 shows the experimental setup used to generate the results. As shown in the setup, first, the documents are preprocessed. For Cosine Similarity, Word Mover's and BERT, stopwords were not removed during preprocessing. Once the data is preprocessed, embeddings of the clean data are generated. The pairwise distance between the vectors of the no-opinion dataset and the CLSA dataset is computed to get the distance matrix. The distance matrix is then used to obtain clusters using Agglomerative Clustering. The experiments were conducted with varying numbers of clusters, different affinity metrics and linkage criteria. Further, the results were analyzed using dendrograms and scatter plots. These results will be discussed later in this chapter.

For Cosine Similarity, doc2vec embeddings are generated using Gensim's doc2vec. These embeddings are then utilized to compute the pairwise distance using the cosine metric. The distance matrix is then used to obtain clusters using Agglomerative Clustering. The best results for the experiment were obtained when the parameter were: *affinity* is *cosine*, *n_clusters* is 3, and *linkage* is *complete*.

A similar approach was followed for Word Mover's Distance and word2vec embeddings were generated using pretrained word2vec model, 'word2vec-google-news-300'. A precomputed cosine matrix is given as input for clustering. For Agglomerative Clustering, parameters were: *affinity* set to *precomputed*, *n_clusters* is 3, and *linkage* set to *complete* to produced the best results.

For BERT, sentence BERT embeddings were generated using the pretrained 'all-mpnet-base-v2' model provided by Hugging Face. These embeddings were then used to compute the pairwise distance matrix using the Euclidean distance. The best results were obtained for clustering when parameters were: *affinity* set to *Euclidean*, *n_clusters* is 3, and *linkage* set to *Ward*.

3.4.1 Clustering Performance Evaluation

Measuring the performance of clustering algorithms is vital. This is especially true as clusters are often manually and qualitatively inspected to determine whether the results are meaningful.

External and Internal validity indices measure the quality of clustering results. External index measures similarity between two clusters where the first is the known cluster structure of a dataset and the second results from the clustering procedure. In contrast, internal indices measure the goodness of a clustering structure using quantities and features inherent in the dataset when external information is not available.

The metrics used for evaluating cluster performance require the knowledge of ground truth class assignment. For this, labels from active learning process are used as gold standard to evaluate clustering performance.

Mutual Information (MI): Given the ground truth knowledge, it measures the mutual dependence between the ground truth and cluster obtained after the clustering algorithm. It quantifies the amount of information obtained about one cluster by observing the other cluster. It is nonnegative, i.e., values close to zero indicate two largely independent label assignments, while values close to one indicate significant agreement. It is also known as information gain [86, 37].

The MI between two label assignments, U and V, can be calculated by the eq. 3.2:

$$MI(U,V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} \log\left(\frac{N(U_i \cap V_j)}{|U_i||V_j|}\right) \tag{3.2}$$

Adjusted Mutual Information (AMI): It is a variation of mutual information used for comparing clusterings. It accounts that the MI is generally higher for two clusterings with a more significant number of clusters, regardless of whether there

is more information shared. It is independent of the absolute values of the labels. Furthermore, it is also symmetric. The value of AMI is between 0 and 1. A value of 1 means the two clusters are identical or perfectly matched. Random partitions have an expected AMI around 0 on average. AMI is used when the ground truth clustering is an unbalanced and small cluster(s) exist [76].

Using the expected value, E, the adjusted mutual information can then be calculated by eq. 3.3:

$$AMI = \frac{MI - E[MI]}{mean(H(U), H(V)) - E[MI]} \tag{3.3}$$

where $H(U)$ and $H(V)$ are entropy or amount of uncertainty for partition sets U and V, and $E[MI]$ is expected value for MI and can be calculated by eq. 3.4, as shown below:

$$\begin{aligned} E[MI] \quad = \quad & \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \sum_{n_{ij}=(a_i+b_j-N)^+}^{\min(a_i,b_j)} \frac{n_{ij}}{N} \log\left(\frac{N \cdot n_{ij}}{a_i b_j}\right) \times \\ & \frac{a_i! b_j! (N-a_i)! (N-b_j)!}{N! n_{ij}! (a_i - n_{ij})! (b_j - n_{ij})! (N - a_i + n_{ij})!} \end{aligned} \tag{3.4}$$

V-Measure: V-Measure is defined as the harmonic mean of homogeneity and completeness of the clustering. Both these measures can be expressed in terms of the information theory's mutual information and entropy measures. The range of V-measure is between 0 and 1, where 1 corresponds to a perfect match between the clusterings. It is equivalent to the normalized mutual information when the aggregation function is the arithmetic mean [77].

$$V = \frac{(1+\beta) \times homogeneity \times completeness}{\beta \times homogeneity + completeness} \tag{3.5}$$

where *homogeneity* is

$$1 - \frac{H(C|K)}{H(C)}$$

completeness is

$$1 - \frac{H(K|C)}{H(K)}$$

and

$$\begin{aligned} H(C|K) &= -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n}\log\left(\frac{n_{c,k}}{n_k}\right) \\ H(C) &= \sum_{c=1}^{|C|}\frac{n_c}{n}\log\left(\frac{n_c}{n}\right). \end{aligned}$$

Fowlkes-Mallows Index (FMI): It is an external evaluation method used to determine the similarity between two clusterings and measures the similarity between two hierarchical clusterings or clustering and a benchmark classification. A higher value; i.e., 1, for the Fowlkes–Mallows index indicates a greater similarity between the clusters and the benchmark classifications [21].

The FMI score can be defined as the geometric mean of the pairwise precision and recall and calculated as

$$FMI = \frac{TP}{\sqrt{(TP+FP)(TP+FN)}} \tag{3.6}$$

where TP is number of *True Positives*, FP is the number of *False Positives* and, FN is the number of *False Negatives*.

3.5 Active Learning

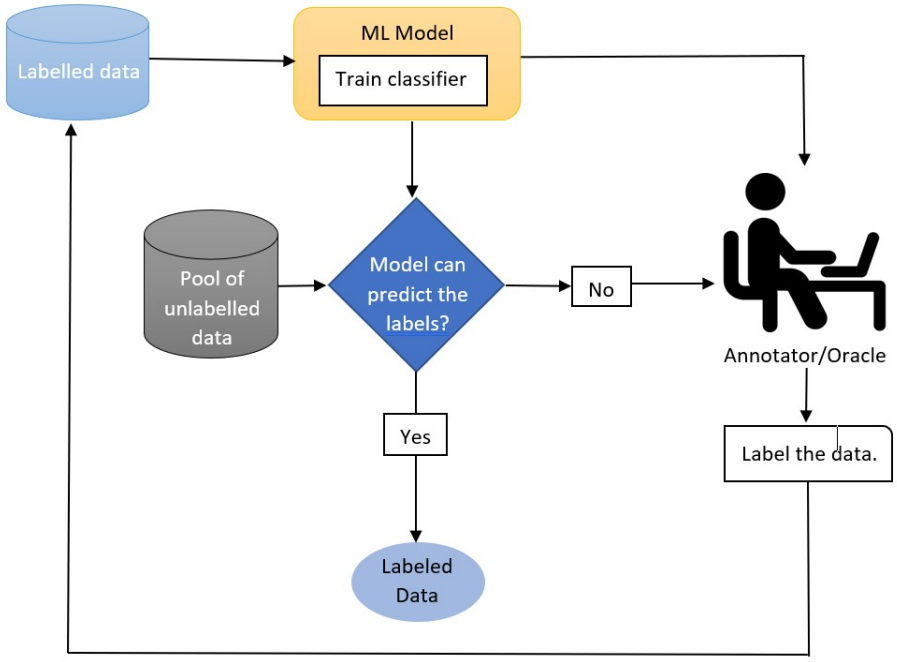

Figure 3.5: Active Learning Procedure

The active learning process begins with a small labeled dataset, $\mathcal{L}$. In this research, the labeled dataset consists of 206 examples; 112 are labeled as "1" as they are opinion responses, and 94 are labeled as "2" as they are no-opinion responses, can be seen in the Figure 3.6.

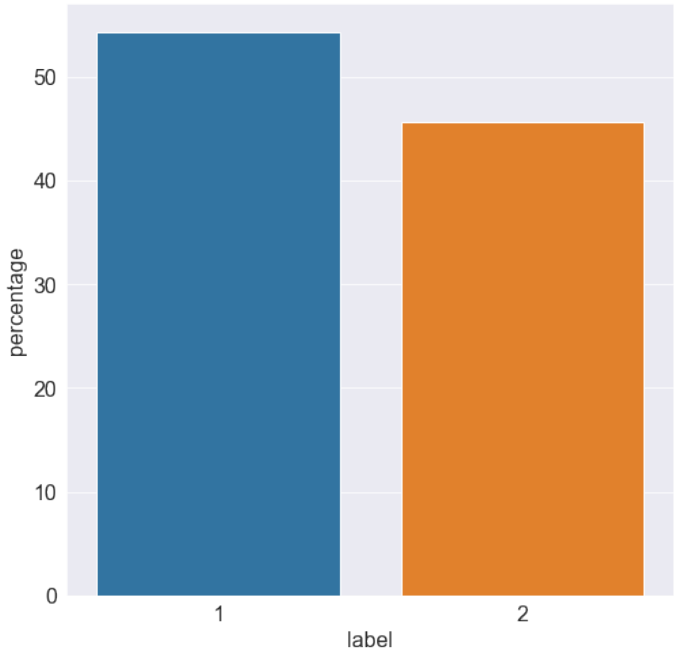

Figure 3.6: Label Count in the Labeled Dataset, $\mathcal{L}$, where "1" represents Opinion responses and "2" represents no-opinion responses.

In most cases, $\mathcal{L}$ is chosen randomly. A randomly chosen set for initial training may not be a good idea when dealing with real-world applications as it may not have the same data distribution as the whole dataset because of its small size. A randomly selected initial training set works well only when there is no class imbalance in the dataset. However, the CLSA dataset is highly imbalanced, and the initial training set instances are manually selected for this research to have the most informative examples and prevent the model from being skewed towards the majority class and ensure that the model reflects the true nature of the minority class, in this case, no opinion responses. Moreover, a balanced dataset potentially eliminates the class imbalance's adverse effects on the model's performance.

A classifier is needed to predict the labels of unlabeled examples in the pool. Uncertainty sampling, a commonly used exploitation based selection strategy in text classification, is used to train classifiers using the examples labeled by the oracle, and then the classifier is used to classify the remaining unlabeled examples and for examples for which classifications are least certain are selected for labelling by the oracle. In literature, it has been found that several selection strategies algorithms use Support Vector Machines, Logistic Regression, Naïve Bayes, Maximum Entropy and

Random Forest. As discussed in Section 3.5, this research focuses on Naïve Bayes, SVM and RF based active learners.

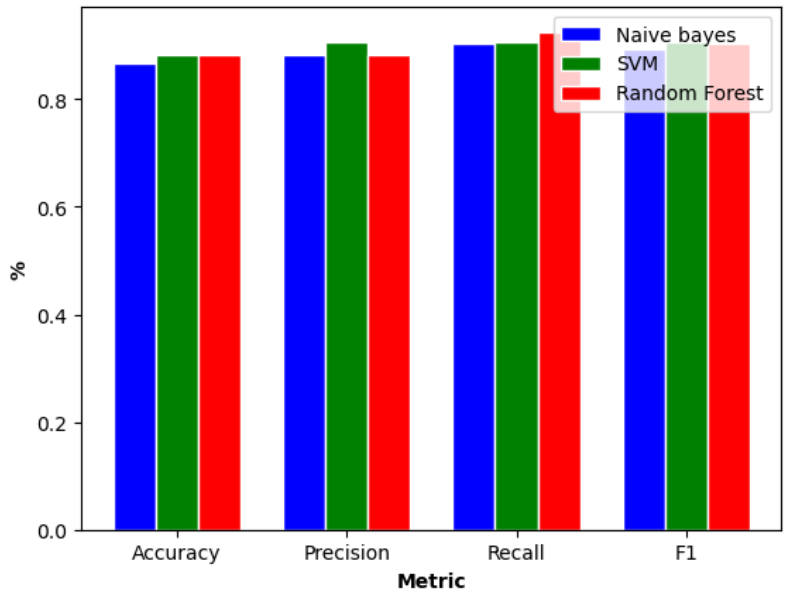

Figure 3.7: Classifiers Used in Active Learning Process.

Classfier	Accuracy	Precision	Recall	F1
Naïve Bayes	0.867	0.88	0.902	0.892
SVM	0.882	0.905	0.905	0.905
Random Forest	0.882	0.881	0.925	0.902

Table 3.5: Classifiers used for Active Learning and their performance.

The Figure 3.7 and the Table 3.5 shows the performance of the three classifiers trained on the labeled dataset, $\mathcal{L}$. The dataset was split into test and train, 30% for testing and 70% for training.

The classifier predicted the labels and class probability of the first 500 examples from the pool, and then the classifier boundary was defined. After manually going through the predicted labels, class probability and response, it was observed that mostly when the predicted label is "2" and class probability close to 0.5, the classifier was uncertain about the prediction. For this experiment, the class probability between 0.48 to 0.535 was selected as the classifier boundary. After defining the classifier boundary, five hundred examples were selected from the pool for each iteration, and the classifier predicted the labels and the prediction probability for both

classes. If the predicted label was "2" and the probability within in the classifier boundary, the Oracle was asked to label the example. The process was continued till the stopping criteria was reached, i.e., till there were no more examples in the unlabeled pool.

The Active Learning algorithm used for this research is as below:

Algorithm 1: Active Learning Algorithm

Input: L: base learner L,

$\mathcal{L}$: dataset of labeled instances,

$\mathcal{U}$: pool of unlabeled data,

k: number of iterations to be performed,

n: number of examples to be added to $\mathcal{L}$ after each iteration

for $i \leftarrow 1$ **to** k **do**

- let h be the classifier obtained by training $\mathcal{L}$ on L
- let M be the n examples in $\mathcal{U}$ for which h makes least confident predictions;

foreach $n \in M$ **do**

remove n from $\mathcal{U}$ and ask user for its label;

add $\langle n, h(n) \rangle$ to $\mathcal{L}$;

where $k = len(\mathcal{U})/500$.

3.5.1 Evaluation Metrics

A number of approaches are used to evaluate Active Learning approaches. Performance can be measured in terms of accuracy, F1 score, precision or recall. Learning curves are also used to monitor the progress of the labelling process in terms of the classifier performance. From the learning curve a Area Under the Learning Curve (AULC) score can be calculated. In this research, performance is measured in terms of accuracy, precision and recall.

Receiver Operating Characteristic

The Receiver Operating Characteristic (ROC) curve represents forecasting precision and offers a visual and statistical tool for decision-making. Each point on the ROC

curve represents a sensitivity/specificity pair. ROC curves are valuable because they permit the comparison of variables and summarize accuracy across a range of tradeoffs between correct and incorrect classification probabilities. In practice, ROC curve analysis evaluates the classification ability of one independent (predictor) variable that is continuously measured and one dependent (outcome) variable that is dichotomously measured.

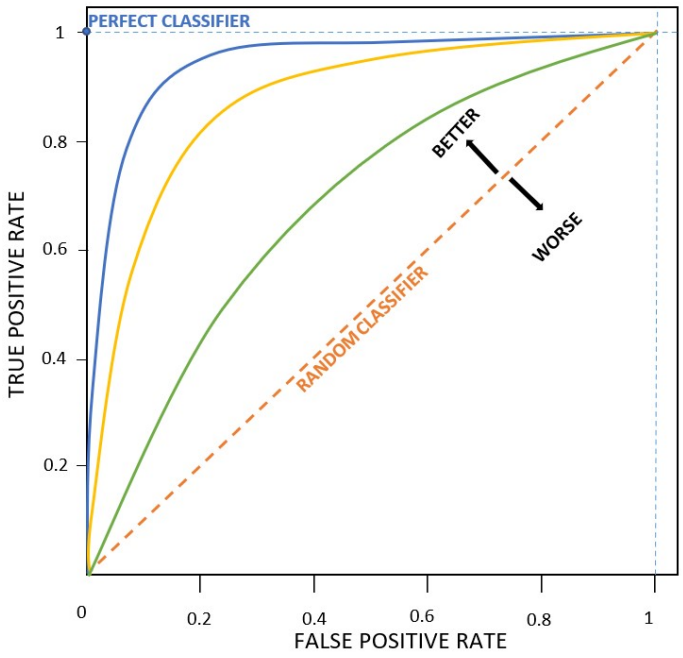

Figure 3.8: ROC Curve

The area under the ROC curve, also called ROC AUC, is typically a measure of test usefulness and provides a value to summarize the learning algorithm's performance. A larger area means a more practical test, and the area under the ROC curve is also used to compare the test's usefulness.

Precision-Recall Curve

There are many ways to evaluate the skill of a prediction model. Precision and Recall measures are helpful in applied machine learning for evaluating binary classification models. Precision-Recall (PR) curves are often used in information retrieval and have been cited as an alternative to ROC curves for tasks with a large skew in the class distribution. An important difference between ROC space and PR space is the

visual representation of the curves.

Precision is defined as a measure of the proportion of extracted items that the system got right. It is a ratio of true positives divided by the sum of true positives and false positives and describes how well a model predicts the positive class.

$$Precision = TruePositives/(TruePositives + FalsePositives)$$

Recall measures the fraction of positive examples that are correctly labeled and it is the ratio of the number of true positives divided by the sum of the true positives and the false negatives and is the same as sensitivity.

$$Recall = TruePositives/(TruePositives + FalseNegatives)$$

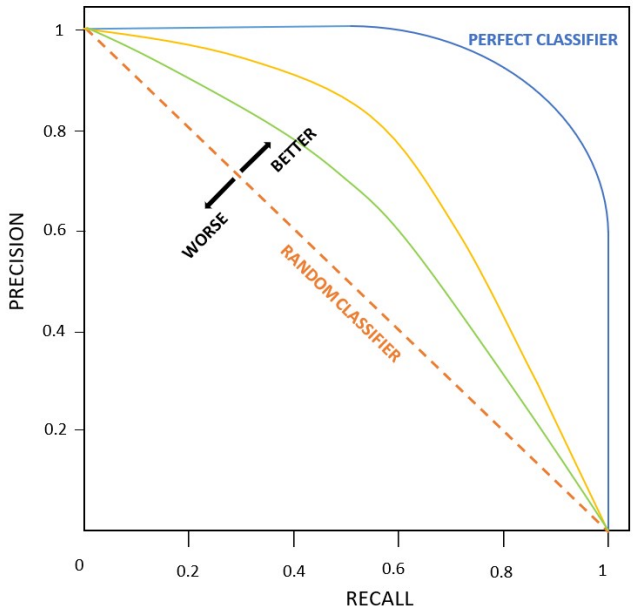

Figure 3.9: PR Curve

PR curves are helpful in cases where the datasets are highly skewed, i.e., when there is an imbalance in the observations between the two classes. It compares false positives to true positives and captures the effect of the large number of negative examples on the algorithm's performance.PR curves give a more informative picture of classifiers performance [11].

Accuracy, Precision and Recall

Accuracy [73] is the most intuitive performance measure and it is the ratio of correct predictions to the total number of predictions made by the model. The formula for calculating accuracy is:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

For binary classification, the accuracy can be calculated as:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Recall [73], also refered to as True Positive Rate or Sensitivity, is the proportion of correct positive predictions to the total number of predictions related to the positive class. High precision score relates to the low false positive rate. The formula for recall is:

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

Precision [73], also refered to as Positive Predictive value or Confidence, is the ratio of correct positive predictions to the total number of positive predictions made by the model. The formula for calculating precision is:

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

3.6 Chapter Summary

This chapter elaborates on the CLSA dataset, its characteristics, and the preprocessing steps that are taken to clean it. It also discusses the Unsupervised Approach in which the the embedding from clean CLSA dataset and the no-opinion dataset are generated. These embeddings are then used to calculate a pairwise distance matrix, using three different methods, cosine distance and doc2vec embeddings, WMD and word2ved embeddings and Euclidean and BERT embeddings. These pairwise matrices are used to generate clusters using Agglomerative Hierarchical Clustering. Further, Active Learning Approach and the metrics used to evaluate the performances are discussed. It starts with a a small balance dataset which is manually labeled and used to train three classifiers, RF, SVM and NB, which are then used to build AL model.

Chapter 4

Experimental Setup and Results

This section will provide a detailed description of the summary of experiments, results and the required analysis to interpret them. In order, we will be comparing Cosine Similarity, Word Mover's Distance, BERT and Active Learning on the CLSA dataset.

It is common to test new techniques and algorithms on well-established datasets where benchmark and state-of-the-art results are documented and corroborated. This provides a precise evaluation method for new techniques and architectures and allows for consistent results despite differences in technique. However, determining the results can be challenging when these techniques and algorithms are evaluated on a new dataset. Since the datasets used to train ML models are enormous, it is humanly impossible to manually check it for any bias, inconsistency or sparsity that might be present. Moreover, often the datasets are inaccurate and not suitable for the task at hand.

We start this chapter with a brief description of our experimental setup followed by providing a qualitative analysis and its correspondent quantitative results. We conclude this chapter by providing a performance comparison.

4.1 Experimental Setup

The research work was carried out in three steps. First, we used hierarchical clustering, which is an unsupervised ML technique, to cluster no-opinion responses in the CLSA dataset. We have experimented with three different distance measures, such as, cosine similarity, word mover's distance and Euclidean distance paired with three different embeddings, doc2vec, word2vec and BERT. Second, after evaluating the results of Unsupervised methods, the Semi-Supervised ML method was used to solve the same problem more efficiently. Finally, we assess and compare the results. Clustering helped find which embedding worked the best and cluster no-opinion responses and other responses. Using AL, we labeled the whole CLSA dataset. Moreover, using

the SVM classifier for AL, we could correctly label most of the no-opinion responses. This was manually verified.

The crucial components for running the experiments are a programming language to write and run the scripts and a platform for result analysis. Python is a cross-functional interpreted language with high readability that helps save time by typing fewer code lines to accomplish the tasks and provides extensive data analysis support. Therefore, the entire project was developed using Python v3.7.3 on Jupyter Notebook. The essential modules required for the experiments are mentioned in the Table 4.1 along with their version.

Module	**Version**
Pandas	1.1.3
Numpy	1.19.5
Matplotlib	3.3.2
Wordcloud	1.8.1
Plotly	5.3.1
NLTK	3.5
Gensim	3.8.3
Pyemd	0.5.1
SciPy	1.6.0
Scikit-Learn	0.23.2
Sentence_transformers	0.3.6
Tensorflow	2.6.0
Texthero	1.1.0

Table 4.1: List of modules used for the experiments along with their version.

Pandas is a data manipulation and analysis library. In this research, Pandas was used for preparing the dataset as it allows importing data from various file formats and provides various data manipulation operations such as merging, joining, insertion, deletion, reshaping, cleaning, slicing, indexing and subsetting large datasets.

Matplotlib is a cross-functional data visualization and graphical plotting library. *Plotly* is another visualization library used to make interactive web-based plots. Another library used in this research is *WordCloud*, a data visualization technique used for representing text data in which the size of each word indicates its frequency or

importance. Significant textual data points can be highlighted using a word cloud.

NLTK is one of the most used libraries for natural language processing and computational linguistics. It allows for text processing for tokenization, parsing, classification, stemming, tagging and semantic reasoning.

Gensim library is used for unsupervised topic modelling and natural language processing. It includes LSA, LDA, fastText, word2vec and doc2vec algorithms; the latter two are used in this research.

SentenceTransformers is a framework for generating state-of-the-art sentence, text and image embeddings. It provides several pre-trained models for over 100 languages and is helpful for textual similarity tasks. In this research, a pre-trained model has been used to obtain BERT embeddings for sentences.

Texthero is a Python toolkit to work with text-based datasets rapidly and efficiently. It offers text preprocessing, mapping texts to vectors, TF-IDF, vector visualization and custom-word embeddings. In this research, Texthero is used to obtain t-Distributed Stochastic Neighbor Embedding (t-SNE) [97], a dimensionality reduction technique, to represent high-dimensional embeddings to two-dimensional space for visualization purposes.

Scikit-learn package provided support and the necessary modules for Agglomerative Clustering and Random Forest, Support Vector Machine and Multinomial Naïve Bayes, classifiers used for Active Learning. It was also used to create features (TfidfVectorizer), calculate distance matrix and evaluation (accuracy score, precision and recall).

SciPy is used for scientific and technical computing, including optimization, interpolation, integration, and other tasks in science and engineering. In this research, dendrograms of hierarchial clustering are plotted using this library.

NumPy is most commonly used for scientific computing in Python and facilitates advanced operations on extensive data.

For most parts of this research, JupyterHub hosted on the Timberlea server was used. The server timberlea.cs.dal.ca is a general Linux server provided by Dalhousie University. For processing huge amounts of data, JupyterHub hosted on the server calvert.research.cs.dal.ca:8000 was used. The server calvert.research.cs.dal.ca:8000 is equipped with a GeForce RTX 2080 card.

However, use of Timberlea and calvert.research.cs.dal.8000 was mostly restricted to getting the results for Cosine Similarity, Word Mover's Distance and BERT.

4.2 Results

The goal of clustering is to identify highly similar groups of elements. The distance matrix was provided as input to the Agglomerative clustering method, and the dendrogram of various clusters was generated from the distance matrix of various embeddings. Dendrograms capture the essence of the groupings, but they may not be a remarkably accurate representation of the data, as we will see [27]. However, the structure of dendrograms does not provide much information about the specific clusters identified by hierarchical clustering.

Scatterplots are a widespread type of visualization designed to emphasize the distribution of data plotted in two dimensions [83]. They visualize multidimensional data by mapping data cases to graphical points in a Cartesian space defined by two or three orthogonal axes. The position of each point representing a data case depends on the data dimension assigned to each axis. The simplicity and flexibility of scatterplots make them ideal for visualizing research information [18]. One of the primary purposes of scatterplots is to visualize the data points of hierarchical clustering.

Scatterplot has been utilized to understand patterns and analyze clusters in this research as they are suitable for exploring the clusters and seeing some exciting findings. For plotting scatterplots, high dimensional embeddings are reduced using t-SNE embeddings, which is well suited for the visualization, and the data points are coloured based on the cluster assigned by hierarchical clustering.

To further evaluate the clustering performance, Mutual Information (MI), Adjusted Mutual Information (AMI), V-Measure (VM) and Fowlkes-Mallows Index (FMI) scores are used, which are discussed in the Section 3.4.1. The labels obtained after Active Learning process are taken as the gold standard to evaluate the clustering performance. Results of Active Learning process are discussed in later in this section.

A number of approaches are used to evaluate active learning approaches. Performance can be measured in terms of accuracy, F1 score, precision or recall. Learning

curves are also used to monitor the progress of the labelling process in terms of the classifier performance. From the learning curve an Area Under the Learning Curve (AULC) score can be calculated. In this research, performance is measured in terms of accuracy, precision, recall, ROC Curve and PR Curve.

4.2.1 Cosine Similarity

The Cosine distance matrix was provided as the input to the Agglomerative clustering method. The Figure 4.1 shows the dendrogram with three clusters. It can be seen that the clusters red and blue are more similar to each other as the height of the link that joins them together is smaller compared to the green cluster. However, dendrograms are challenging to read when representing such a vast dataset. Therefore, a better approach is using scatter plots.

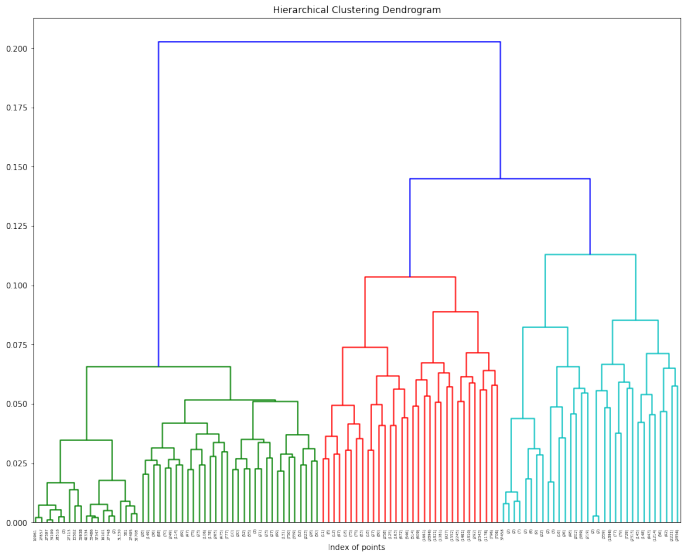

Figure 4.1: A dendrogram of complete linkage representing the clusters generated by cosine distance matrix of doc2vec embeddings. The x-axis shows the index of points of various clusters (green, red and blue). The y-axis shows the distance between the cluster at the time they were clustered.

For further analysis of clusters, t-SNE embeddings of Doc2vec embeddings were obtained and a scatterplot was plotted. On further analysis of the clusters, it was found that this approach for detecting no-opinion responses is not appropriate. The reason being that the sentences in the same clusters did not have anything in common and clustering could not cluster the no-opinion responses together.

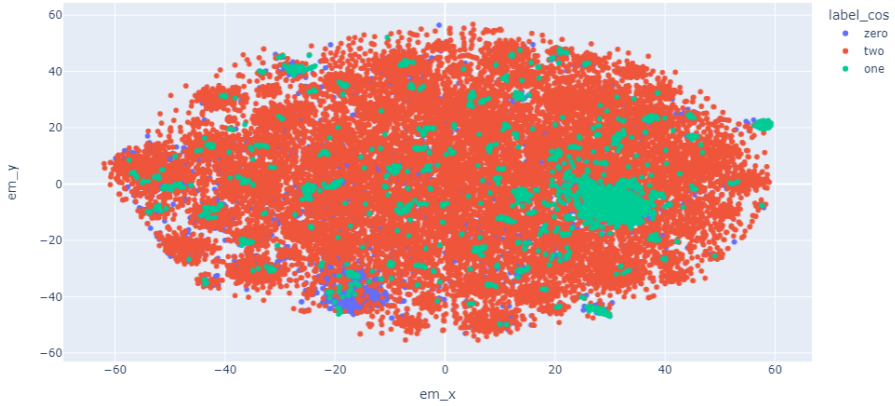

Figure 4.2: Scatter plot of clusters obtained from above dendrogram. The data is partitioned into three clusters and the linkage is complete. The cluster '0' is coloured blue, cluter '1' is green and rep represents cluster '2'.

The experiment was repeated by changing the linkage, affinity and number of clusters, but there was no improvement in the results. Although the single and average linkage results are not shown here as they performed very poorly on the clustering task.

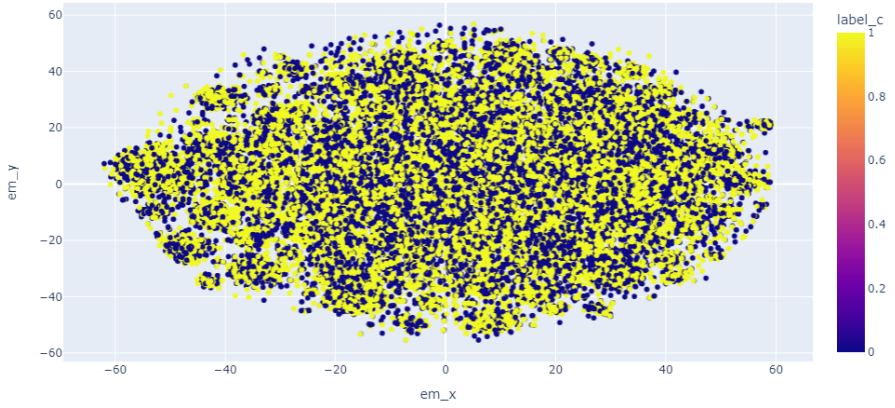

Figure 4.3: Scatter plot of doc2vec embeddings where affinity is precomputed, linkage is complete and number of clusters are two. Blue colour represents cluster '0' and yellow represents cluster '1'.

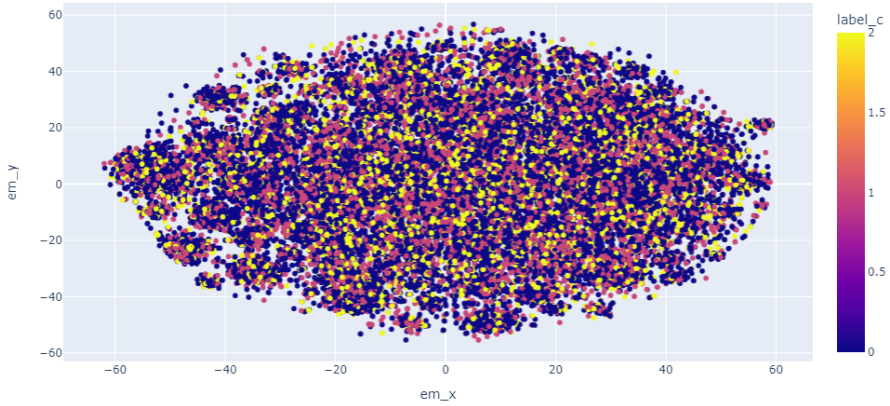

Figure 4.4: Scatter plot of doc2vec embeddings where affinity is precomputed, linkage is complete and number of clusters are three. Blue colour represents cluster '0', pink is cluster '1' and yellow represents cluster '2'.

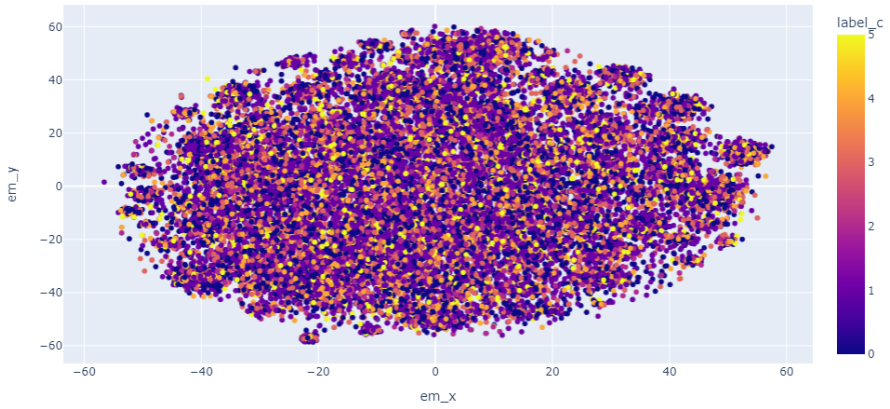

Figure 4.5: Scatter plot of doc2vec embeddings where affinity is cosine, linkage is complete and number of clusters are six.

n_cluster	MI	AMI	VM	FMI
2	0.00611	0.03017	0.03023	0.89570
3	0.00616	0.01242	0.01247	0.65977
4	0.00739	0.01374	0.01381	0.64986
5	0.00746	0.01200	0.01208	0.61059

Table 4.2: Clustering Performance Evaluation Metric Scores with varying number of clusters on doc2vec embeddings when affinity is cosine and linkage is complete.

The Table 4.2 shows the clustering evaluation performance. As it can be seen, the FMI score for two clusters is the highest and then there is a significant decrease as the number of clusters increases. The FMI score for n_cluster= 2 is closer to 1, indicating a more significant similarity between the clusters. Howerver, FMI has an undesirable property of having a very high value when the number of clusters is small, even for independent clusterings [100]. The MI, AMI and VM scores for all the clusters are very close to 0, indicating disagreement between the clusters. Another interesting thing to notice is, even though MI scores are low, they tend to increase with increase in the number of clusters.

4.2.2 Word Mover's Distance

A precomputed square Word Mover's Distance matrix was provided as the input to the Agglomerative clustering method. The Figure 4.6 shows the dendrogram where the number of clusters is three.

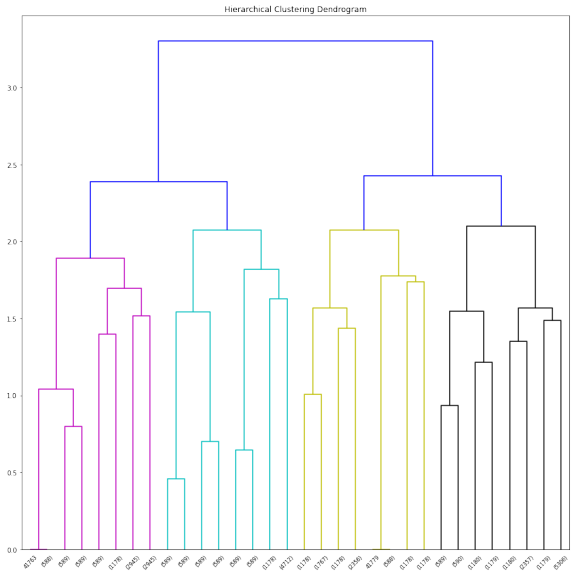

Figure 4.6: A dendrogram representing the clusters generated by WMD distance matrix of word2vec embeddings. The x-axis shows the index of points of various clusters (green, blue, yellow and purple). The y-axis shows the distance between the cluster at the time they were clustered.

For further analysis of clusters, average vectors of each word in sentences of the CLSA dataset were calculated to obtain the word2vec embeddings of the sentences.

Though there are other approaches to do so, each approach has advantages and shortcomings. According to Le and Mikolov [44], the average vectors from the word2vec approach performs poorly as it loses the word order and fails to recognize many sophisticated linguistic phenomena. In contrast, according to Kenter et al. averaging word embeddings of all words in a text has proven to be a strong baseline for tasks like short text similarity [32].

Another approach uses TF-IDF to obtain word2vec embeddings of a sentence. In this approach, the word vectors are multiplied with their TF-IDF scores, and then the average is taken to obtain the embeddings. This approach decreases the influence of the most common words.

A more advanced approach, proposed by Socher et al., combines word vectors in an order given by a parse tree of sentences using matrix-vector operations [93]. This approach works well for sentence sentiment analysis as it depends on parsing.

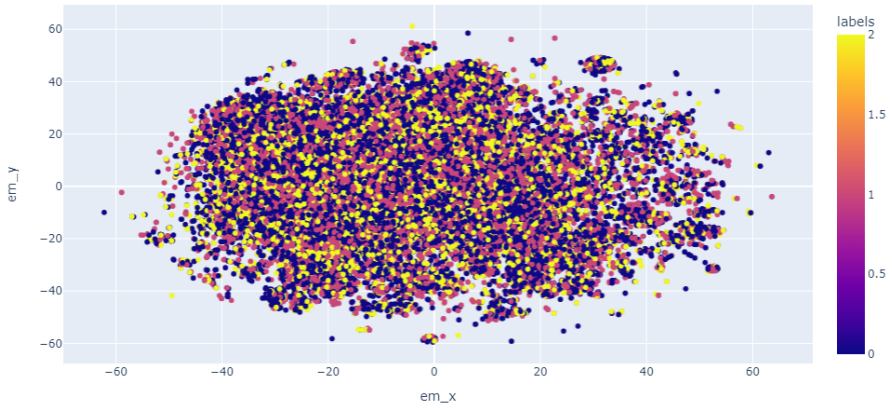

Figure 4.7: Scatter plot of averaged word2vec embeddings where affinity is precomputed, linkage is complete and number of clusters are three.

As can be seen in the Figure 4.5, clustering could not group no-opinion responses in one cluster. Further analysis of the clusters revealed no commonality in clusters. Moreover, this clustering is not appropriate, and there is room for improvement. Although the experiment was repeated by changing the linkage and number of clusters, there was no improvement in the results. Further, single and average linkage performed poorly on the clustering task.

n_cluster	MI	AMI	VM	FMI
2	0.346053e-05	-2.27755e-05	0.926580e-05	70302.0e-05
3	1.33717e-05	-8.53086e-05	10.0e-05	94829.0e-05
4	1.32119e-05	-3.17461e-05	1.84634e-05	50195.0e-05
5	1.41731e-05	-4.17613e-05	1.75277e-05	46071.0e-05

Table 4.3: Clustering Performance Evaluation Metric Scores with varying number of clusters on word2vec embeddings.

The Table 4.3 shows the clustering evaluation performance. The MI and VM scores are very low, the AMI scores are in negative. This indicates that the clusters are independent and the cluster partition is very random; which is very evident from the scatterplot in the Figure 4.7.

4.2.3 BERT

The Euclidean distance matrix was provided as the input to the Agglomerative clustering method. The Figure 4.8 shows the dendrogram with three clusters.

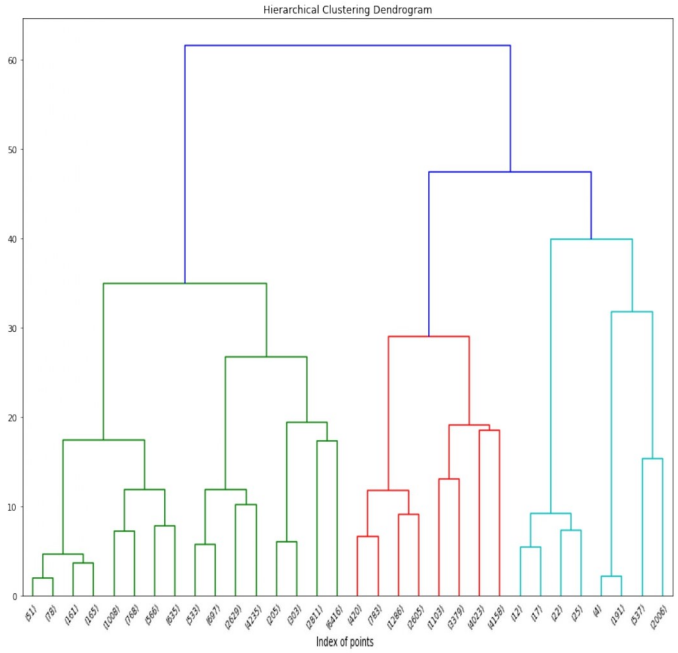

Figure 4.8: A dendrogram representing the clusters generated by the Euclidean distance matrix of BERT embeddings generated by all-mpnet-base-v2 model. The x-axis shows the index of points of various clusters (red, green and blue). The y-axis shows the distance between the cluster at the time they were clustered.

For further analysis of clusters, t-SNE embeddings of BERT embeddings were obtained and a scatter plot was plotted. As can be seen in the Figure 4.9, clusters are neatly separated.

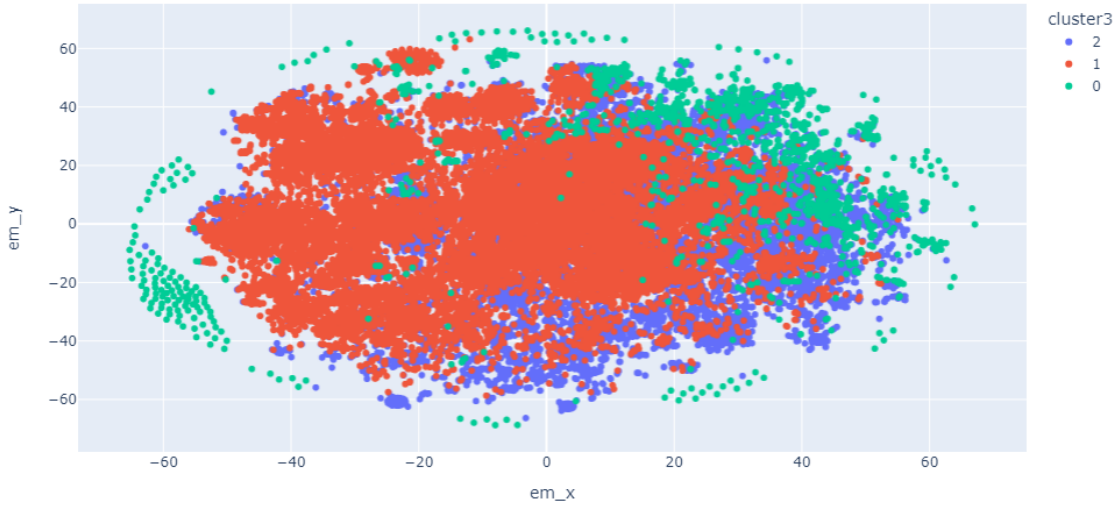

Figure 4.9: Scatter plot of BERT embeddings generated by 'all-mpnet-base-v2' model and number of instances in each cluster. Affinity is Euclidean, linkage is ward and number of clusters are three. The green colour represents cluster '0', red colour represents cluster '1' and blue colour represents cluster'2'. Cluster 0 consists of no-opinion responses along with the opinion responses from the CLSA dataset.

n_cluster	MI	AMI	VM	FMI
2	0.00275	0.00728	0.00731	0.69998
3	0.01429	0.03012	0.03017	0.66686
4	0.01833	0.03825	0.03833	0.66700
5	0.01834	0.03098	0.03106	0.61199

Table 4.4: Clustering Performance Evaluation Metric Scores with varying number of clusters on BERT embeddings.

Though the scatterplot in Figure 4.9 shows that the clustering is done efficiently, the Table 4.4 gives more insights into the Clustering Performance. As shown in the Table 4.4, the FMI score is less when compared to the Clustering Performance of doc2vec and word2vec embeddings. The MI, AMI and VM scores are better than the two clusterings discussed previously but close to zero.

4.2.4 Comparing Results of Unsupervised Techniques

In this experiment, Various metrics were used to evaluate the clustering performances, namely, MI, AMI, VM and FMI. Even though the FMI scores seemed promising, they also have an undesirable property of having a very high value when the number of clusters is small, even for independent clustering. Silke and Dorothea state that the measures based on information-theoretical considerations are promising because they do not suffer from the drawbacks of counting pairs or set overlaps [100]. Therefore, in Table 4.5 below, we use the MI score to evaluate and compare the Clustering approaches. It can be seen that Word Mover's Distance with word2vec embeddings performed very poorly as the scores are almost 0. In contrast, BERT embeddings with Euclidean Distance performed the better than the two discussed before only when the number of clusters were three or more. Moreover, it was also able to cluster no-opinion responses in a cluster.

Number of Clusters	Cosine Similarity	Word Mover's Distance	BERT
2	0.00611	0.0000034	0.00275
3	0.00616	0.0000133	0.01429
4	0.00739	0.0000132	0.01833

Table 4.5: Comparison of Clustering approaches using MI scores as Evaluation Metric.

4.2.5 Active Learning

This subsection discusses the results obtained from Active Learning experiments conducted with three different classifiers.

Active Learning with Random Forest Classifier

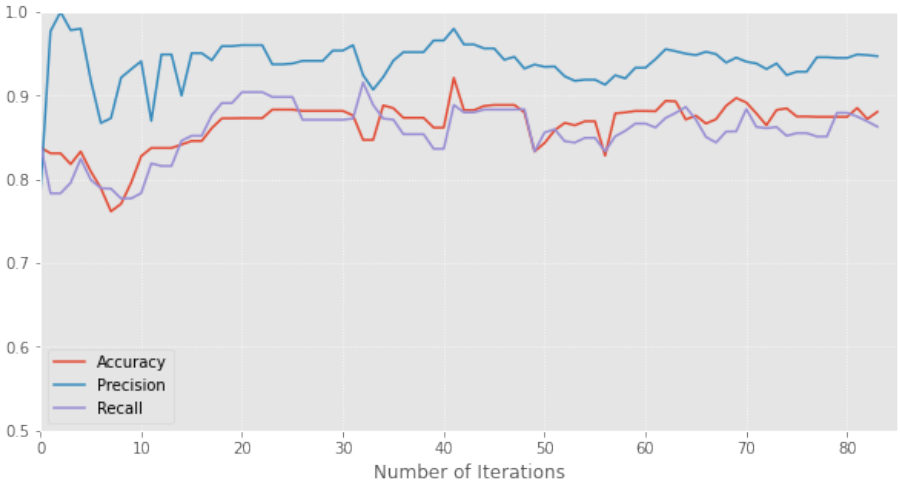

Figure 4.10: Performance of Random Forest Classifier on Test Dataset.

Figure 4.10 shows the performance of the RF model over eighty-five iterations in one epoch. Initially, the model gives an accuracy of 83.82%, precision of 78.57% and recall of 84.28% for the first iteration. As the number of iterations increases, the model better predicts the labels. Accuracy, precision, and recall values drop significantly in the first few iterations, as the model is unsure about the labels and the annotator is asked to label the instances. The RF labeled 163 instances as "no-opinion" responses, accounting for less than 0.4% of the total responses in the CLSA dataset.

Furthermore, the model was uncertain about 127 instances, from a total of 41,641 instances in the pool, and the annotator was asked to label these instances. Moreover, a significant increase can be seen in the accuracy and precision in the last iteration, where the accuracy, precision, and recall on the test data are 88.06%, 94.71%, and 86.29%, respectively.

Figure 4.11 summarizes the ROC Curve of Random Forest Classifier with an AUC score of 0.89. At the same time, in the figure 4.11, AUC-PR summarizes the PR Curve and has a score of 0.81.

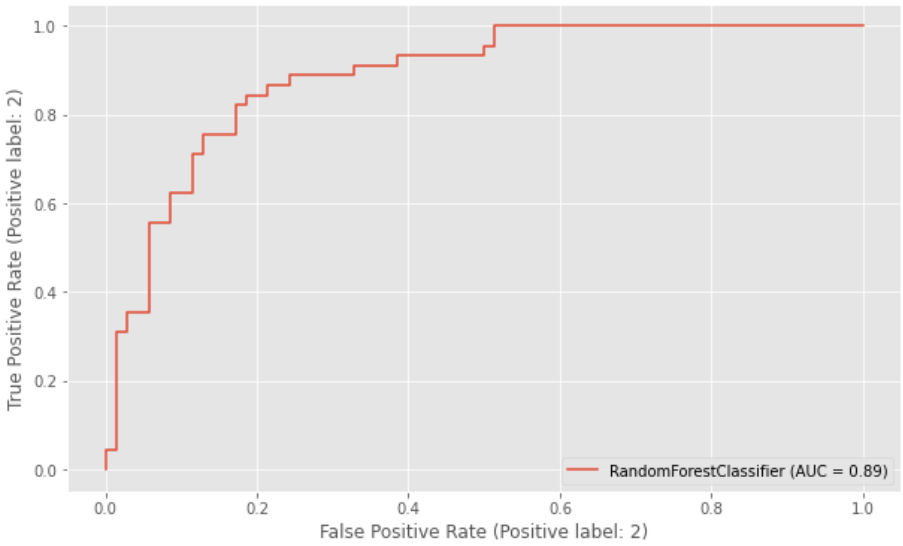

Figure 4.11: ROC Curve for Random Forest Classifier.

For a perfect classifier, ROC and PR Curves should have a score of 1. Further, according to the results obtained, Random Forest performed well at classifying the responses into two classes.

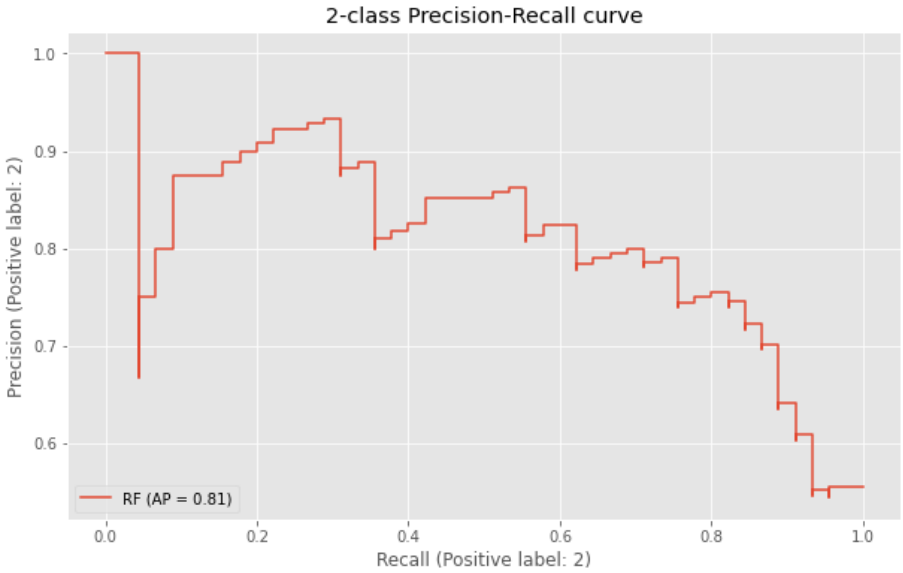

Figure 4.12: PR Curve for Random Forest Classifier.

However, on manually going through the instances labeled as “no-opinion” by the Random Forest classifier, it was found that the classifier mislabeled a lot of opinion responses and no-opinion and vice-versa. Hence, so few instances of the minority

class.

Active Learning with SVM Classifier

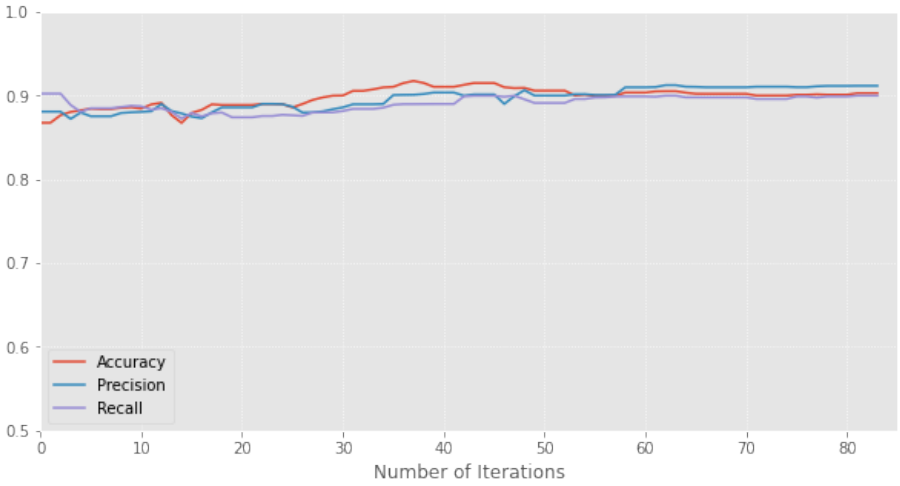

Figure 4.13: Performance of SVM Classifier on Test Dataset.

Figure 4.13 shows the performance of the SVM model over eighty-five iterations in one epoch. Initially, the model gives an accuracy of 86.76%, precision of 88.09% and recall of 90.24% for the first iteration. With the number of iterations, the model better predicted the labels. Accuracy, precision and recall values drop significantly in the first few iterations. The SVM Classifier labeled 1157 instances as "no-opinion" responses, roughly 3% of the total responses collected by the CLSA in English. Further, the model was uncertain about 47 instances, and the annotator provided the labels.

Moreover, there was not much change in the recall, but a steady increase in accuracy and precision can be seen. The last iteration's accuracy, precision, and recall on the test data were 90.29%, 91.18%, and 90%, respectively.

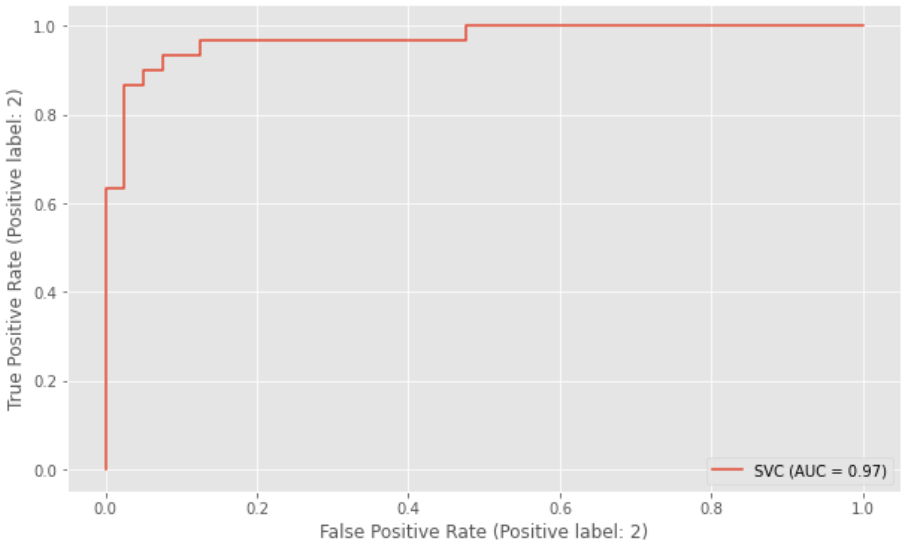

Figure 4.14: ROC Curve for SVM Classifier.

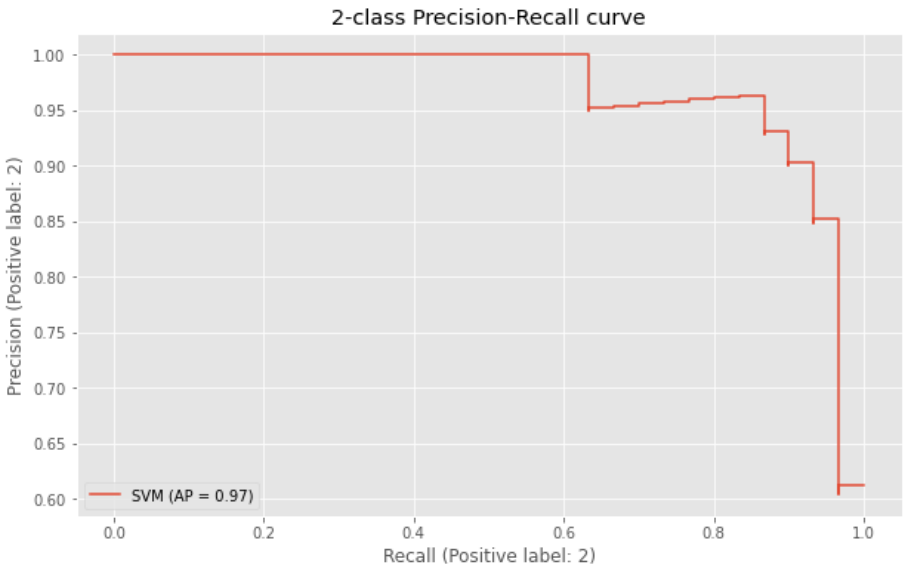

Figure 4.15: PR Curve for SVM Classifier.

In the case of the SVM classifier, both ROC and PR Curves had an AUC score of 0.97, indicating that SVM performed much better than the Random Forest classifier. Moreover, manually going through the instances labeled as no-opinion, it was discovered that the SVM classifier was able to classify more responses and classify them correctly.

Active Learning with Naïve Bayes Classifier

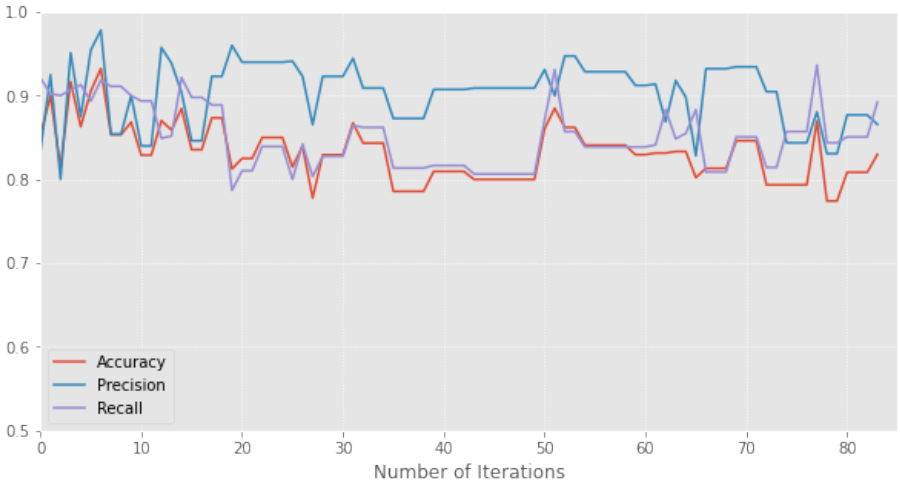

Figure 4.16: Performance of Naïve Bayes Classifier on Test Dataset.

The figure 4.16 shows the performance of the Naïve Bayes model over eighty-five iterations in one epoch. Initially, the model gives an accuracy of 85.29%, precision of 83.33% and recall of 92.1% for the first iteration. Initially, the model's performance fluctuated, dropping significantly in the first few iterations, as the model was uncertain about the labels of the instances. As the number of iterations increased, the uncertainty reduced. The Naïve Bayes Classifier labeled 450 instances as "no-opinion" responses, roughly 1% of the total responses. Further, the model was uncertain about 78 instances, and the annotator provided the labels. However, the last iteration's accuracy, precision, and recall on the test data were 82.97%, 86.56% and 89.23%, respectively.

The Figures 4.17 and 4.18 show the ROC and PR curves of the Naïve Bayes classifier. For the ROC curve, the AUC score is 0.9, and for the PR curve, the AUC score is 0.83, signifying that the Naïve Bayes classifier performed well.

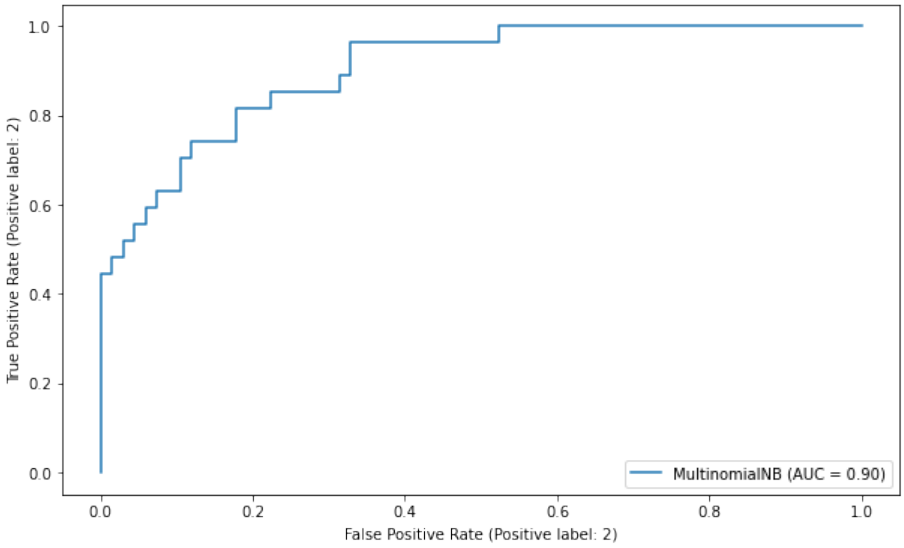

Figure 4.17: ROC Curve for Naïve Bayes Classifier.

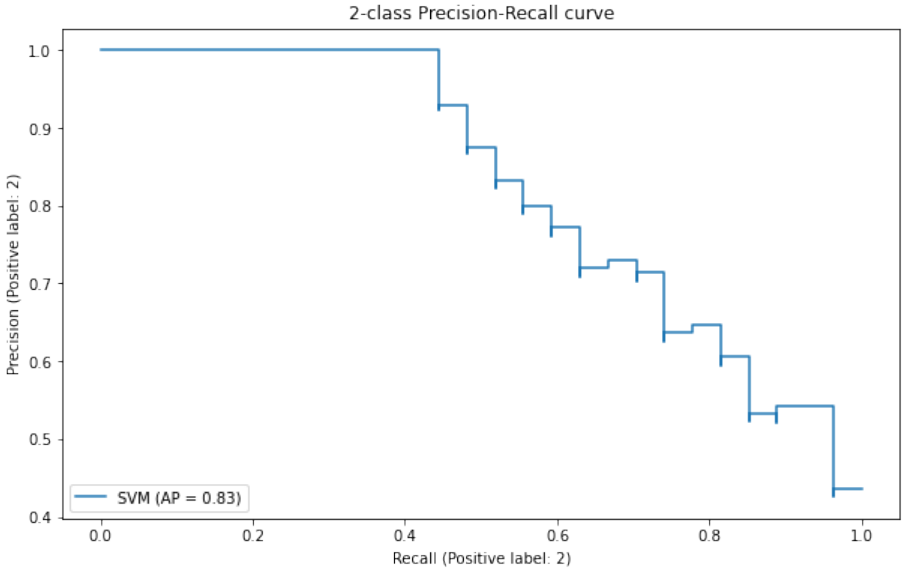

Figure 4.18: PR Curve for Naïve Bayes Classifier.

4.2.6 Comparing Results of Active Learning

Metric	RF	SVM	NB
ROC	0.89	0.97	0.9
PR	0.81	0.97	0.83

Table 4.6: Comparison of Active Learning approach with different Classifiers.

We discussed in the section 3.5.1 that the PR Curve is a better metric when the dataset is highly imbalanced. Based on that, the Naïve Bayes classifier's performance was better than that of Random Forest but it did not outperform the SVM classifier, which performed the best among the three classifiers. As can be seen in the Table 4.6, AUC score of PR curve is high for SVM classifier, i.e., 0.97, and labeled 1157 instances as "no-opinion" responses. Whereas AUC of PR curve for RF was the lowest, at 0.81, and labeled 127 instances as "no-opinion" responses.

4.3 Chapter Summary

In this chapter, all the results were detailed. Key points include:

- Three distance measures paired with embeddings, cosine measure with doc2vec, Word Mover's Distance with word2vec and Euclidean with BERT were clustered, their results are presented and discussed.
- The labels obtained from the Active Learning approach were used as the gold standard to evaluate the performance of unsupervised methods.
- The MI score was used to evaluate and compare the results, and it was found that the Euclidean distance with BERT embedding performed better with an MI score of 0.01429 for three clusters.
- The Active Learning approach experimented with three classifiers, RF, SVM and NB. Their results are presented and discussed.
- The PR curve was used to evaluate and compare AL results. The AL model with SVM classifier outperformed the models with RF and NB classifiers.
- The AUC score for the model with SVM classifier was 0.97, whereas for RF and NB it was0.81 and 0.83 respectively. Moreover, SVM classifier labeled 1157 instances as no-opinion responses in the CLSA dataset, compared to 127 instances by RF classifier and 450 instances by NB classifier. Furthermore, the results were manually validated.

Chapter 5

Conclusion and Future Work

This chapter concludes this research work by providing the summary, limitations, and future work.

The first section reviews the research and provides a summary of the problem, solutions, and results. The second section discusses the shortcomings of the research and possible solutions to it. The last section discusses how the research can be improved by looking into the solutions for some limitations and analyzing the results using alternate techniques.

5.1 Conclusion

This thesis tries to solve the problem of automated detection of no-opinion responses in the open-ended survey data collected by the Canadian Longitudinal Study on Aging (CLSA). To our knowledge, this research is the first of its kind. To achieve the objective of detecting the no-opinion responses, various ML techniques, including similarity measures coupled with embeddings and the Active Learning approach, have been explored in this thesis.

Initially, distance measures and embedding were used to detect no-opinion responses in the CLSA data. Cosine distance between the doc2vec embeddings of sentences was used to get the clusters. Further, the Word Mover's Distance between the sentences was calculated and used to get the clusters. These two methods resulted in poor results, whereas using BERT embeddings, the agglomerative clustering provided better clustering with no-opinion responses largely being grouped in one cluster.

To further evaluate the clustering performance, labels from the Active Learning process were used as the gold standard and metrics like Mutual Information, Adjusted Mutual Information, V Score and Fowlkes-Mallows Index were used. Although clustering on BERT embeddings worked better, the clustering performance

was not impressive enough.

The Active Learning approach was used in this thesis to label the dataset and achieve the goal of this thesis. The active learning process starts with a small labeled dataset, which is used to train a classifier. The labeled dataset, $\mathcal{L}$, was balanced to prevent the model from being skewed. In this experiment, three classifiers, RF, SVM and NB, are trained and used to classify the instances in the pool of unlabeled datasets, $\mathcal{U}$. Various metrics like accuracy, precision, recall, ROC curve and PR curve were used to evaluate the Active Learning models. Since the CLSA dataset is highly imbalanced, the PR curve is a better metric for evaluating model performance. SVM clearly outperformed Random Forest and Naïve Bayes which was verified by manual checking of the instances labeled as no-opinion responses. Additionally, the AUC score of the PR curve classifier was 0.97 for the SVM classifier, better than that of Random Forest and Naïve Bayes, which scored 0.81 and 0.83, respectively.

5.2 Limitations

The Active Learning approach requires human assistance, and hence, it relies on the human expert's knowledge. Occasionally, the expert can be biased, affecting the results obtained. Further, a severely imbalanced dataset makes it challenging to collect and label a subset of the dataset for training, and it also affects the results obtained from the Unsupervised approach. However, the most challenging task is evaluating the clustering performance. Since the labels obtained from Active Learning are used as the gold standard, the clustering performance metrics cannot be wholly accurate.

5.3 Generalizing our Approach

The Active Learning approach mentioned in this thesis can also be applied to surveys other than CLSA as well as the datasets containing unlabeled textual data.

5.4 Future Work

As discussed in the earlier sections, a significant contribution was detecting no-opinion responses in the CLSA dataset. However, every solution naturally generates more questions. We hope that the work presented in this thesis will motivate further

work to explore the uses of Active Learning within Machine Learning and Statistics. Therefore, this section introduces some of the research directions which are closely related to the work in this thesis and appear promising:

- It will be interesting to extend this research and analyze the results with different similarity measures coupled with different embeddings.
- Combine exploitation-based methods with exploration-based methods. Exploration-based selection strategies are preferable in the initial learning stage, and with more labeled examples, exploitation-based selection strategies are more powerful. A better choice would be a combined method of exploration with exploitation.
- Considerable work has already been done researching stopping criteria for active learning. The stopping criterion establishes the balance between the number of labels provided by the user and the accuracy of the labels applied by the system. The gradient of the performance estimate can be a good stopping criterion; though the studies have been conducted only on NER, it can be applied to any active learning setting based on uncertainty sampling.
- Different initialization data can be tested to improve the initial model stability and representativeness of the whole data pool. The seed data initialization influence on the active learning performance must also be researched.
- Since the training dataset is small, Deep Learning cannot be used. But, Siamese Recurrent Neural Network [64] can be used for text similarity to find the most distinct sentences within the data pool and label those instances. This approach would be interesting to look at. Another approach, a combination of Active Learning and Generative Adversarial Neural Networks, could be tried on textual data since it already shows promising results on the MNIST image dataset [106].
- The Active Learning approach mentioned in this thesis can also be applied to surveys other than the CLSA. In order to do so, the labeled dataset, **L**, should be selected and labeled. The classifier selection depends on the problem

being solved. Further, adjustments to the classifier boundary should be made accordingly.

Bibliography

[1] Naoki Abe, Bianca Zadrozny, and John Langford. Outlier Detection by Active Learning. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 504–509, 2006.

[2] Fawaz S Al-Anzi and Dia AbuZeina. Toward an Enhanced Arabic Text Classification using Cosine Similarity and Latent Semantic Indexing. *Journal of King Saud University-Computer and Information Sciences*, 29(2):189–195, 2017.

[3] Dana Angluin. Queries and Concept Learning. *Machine learning*, 2(4):319–342, 1988.

[4] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt. Online Handwriting Recognition with Support Vector Machines- A Kernel Approach. In *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 49–54. IEEE, 2002.

[5] Paul Beatty and Douglas Herrmann. A Framework for Evaluating "Don't Know" Responses in Surveys. In *Proceedings of the Section on Survey Research Methods*, pages 1005–1010. American Statistical Association Washington, DC, 1995.

[6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[7] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[8] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. Universal Sentence Encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[9] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. *arXiv preprint arXiv:1705.02364*, 2017.

[10] Mick P Couper. Technology Trends in Survey Data Collection. *Social Science Computer Review*, 23(4):486–501, 2005.

[11] Jesse Davis and Mark Goadrich. The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240, 2006.

[12] Edith D De Leeuw. Reducing Missing Data in Surveys: An Overview of Methods. *Quality and Quantity*, 35(2):147–160, 2001.

[13] Edith D De Leeuw, Gideon J Mellenbergh, and Joop J Hox. The Influence of Data Collection Method on Structural Models: A Comparison of a Mail, a Telephone, and a Face-to-Face Survey. *Sociological Methods & Research*, 24(4):443–472, 1996.

[14] Deanna C Denman, Austin S Baldwin, Andrea C Betts, Amy McQueen, and Jasmin A Tiro. Reducing "I Don't Know" Responses and Missing Survey Data: Implications for Measurement. *Medical Decision Making*, 38(6):673–682, 2018.

[15] Michel Deudon. Learning Semantic Similarity in a Continuous Space. In *Advances in Neural Information Processing Systems*, pages 986–997, 2018.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[17] Bruce A Draper. Image-based Feedback for Learning Object Recognition Strategies. In *VMV*, pages 55–56, 2000.

[18] Niklas Elmqvist and Jean-Daniel Fekete. Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, 2009.

[19] Jae-Hong Eom and Byoung-Tak Zhang. Pubminer: Machine Learning-Based Text Mining System for Biomedical Information Mining. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 216–225. Springer, 2004.

[20] Susana Eyheramendy, David D Lewis, and David Madigan. On the Naive Bayes Model for Text Categorization. In *International Workshop on Artificial Intelligence and Statistics*, pages 93–100. PMLR, 2003.

[21] Edward B Fowlkes and Colin L Mallows. A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[22] K Kramer Franklin. Multivariate Correlation Analysis of a Student Satisfaction Survey. Technical report, ERIC, 1994.

[23] Terrence S Furey, Nello Cristianini, Nigel Duffy, David W Bednarski, Michel Schummer, and David Haussler. Support Vector Machine Classification and Validation of Cancer Tissue Samples using Microarray Expression data. *Bioinformatics*, 16(10):906–914, 2000.

[24] Siddhant Garg and Goutham Ramakrishnan. Bae: BERT-based Adversarial Examples for Text Classification. *arXiv preprint arXiv:2004.01970*, 2020.

[25] Wael H Gomaa, Aly A Fahmy, et al. A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, 68(13):13–18, 2013.

[26] Gaurav Gupta and Sumit Malhotra. Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example). *Int. J. Comput. Appl*, 975:8887, 2015.

[27] Mark Heckmann and Richard C Bell. A New Development to Aid Interpretation of Hierarchical Cluster Analysis of Repertory Grid Data. *Journal of Constructivist Psychology*, 29(4):368–381, 2016.

[28] Gao Huang, Chuan Quo, Matt J Kusner, Yu Sun, Kilian Q Weinberger, and Fei Sha. Supervised Word Mover's Distance. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4869–4877, 2016.

[29] Ruth E Hubbard, Nader Fallah, Samuel D Searle, Arnold Mitnitski, and Kenneth Rockwood. Impact of Exercise in Community-Dwelling Older Adults. *PLoS One*, 4(7):e6174, 2009.

[30] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[31] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 1972.

[32] Tom Kenter, Alexey Borisov, and Maarten De Rijke. Siamese CBOW: Optimizing Word Embeddings for Sentence Representations. *arXiv preprint arXiv:1606.04640*, 2016.

[33] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. Technical report, Stanford InfoLab, 1997.

[34] Tuomo Korenius, Jorma Laurikkala, Kalervo Järvelin, and Martti Juhola. Stemming and Lemmatization in the Clustering of Finnish Text Documents. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, pages 625–633, 2004.

[35] Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. Machine Learning: A Review of Classification and Combining Techniques. *Artificial Intelligence Review*, 26(3):159–190, 2006.

[36] Kenneth L Kraemer. *The information systems research challenge (vol. III) survey research methods*. Harvard University Graduate School of Business Administration, 1991.

[37] J. Kreer. A Question of Terminology. *IRE Transactions on Information Theory*, 3(3):208–208, 1957.

[38] Jon A Krosnick. The Causes of No-Opinion Responses to Attitude Measures in Surveys: They are Rarely What They Appear to Be. *Survey nonresponse*, pages 87–100, 2002.

[39] Jon A Krosnick, Allyson L Holbrook, Matthew K Berent, Richard T Carson, W Michael Hanemann, Raymond J Kopp, Robert Cameron Mitchell, Stanley Presser, Paul A Ruud, V Kerry Smith, et al. The Impact of "No Opinion" Response Options on Data Quality: Non-Attitude Reduction or an Invitation to Satisfice? *Public Opinion Quarterly*, 66(3):371–403, 2002.

[40] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From Word Embeddings to Document Distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

[41] Alfirna Rizqi Lahitani, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. Cosine Similarity to Determine Similarity Measure: Study Case in Online Essay Assessment. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6. IEEE, 2016.

[42] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. *arXiv preprint arXiv:1909.11942*, 2019.

[43] Pat Langley, Wayne Iba, Kevin Thompson, et al. An Analysis of Bayesian Classifiers. In *Aaai*, volume 90, pages 223–228. Citeseer, 1992.

[44] Quoc Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In *International Conference on Machine Learning*, pages 1188–1196. PMLR, 2014.

[45] David D Lewis and Marc Ringuette. A Comparison of Two Learning Algorithms for Text Categorization. In *Third Annual Symposium on Document Analysis and Information Retrieval*, volume 33, pages 81–93, 1994.

[46] Huiying Li, Dechang Li, Changhai Zhang, and Shubin Nie. An Application of Machine Learning in the Criterion Updating of Diagnosis Cancer. In *2005 International Conference on Neural Networks and Brain*, volume 1, pages 187–190. IEEE, 2005.

[47] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast Text Detector with a single Deep Neural Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[48] Petra Lietz. Research into Questionnaire Design: A Summary of the Literature. *International Journal of Market Research*, 52(2):249–272, 2010.

[49] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 1932.

[50] Roderick JA Little and Donald B Rubin. The Analysis of Social Science Data with Missing Values. *Sociological Methods & Research*, 18(2-3):292–326, 1989.

[51] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.

[52] Itzik Malkiel, Oren Barkan, Avi Caciularu, Noam Razin, Ori Katz, and Noam Koenigstein. RecoBERT: A Catalog Language Model for Text-Based Recommendations. *arXiv preprint arXiv:2009.13292*, 2020.

[53] Marica Manisera and Paola Zuccolotto. A proposal for the treatment of "don't know" responses. Technical report, Syrto Working Paper Series, 2013.

[54] Andrew McCallum, Kamal Nigam, et al. A Comparison of Event Models for Naive Bayes Text Classification. In *AAAI-98 Workshop on Learning for Text Categorization*, volume 752, pages 41–48. Citeseer, 1998.

[55] James L McClelland, David E Rumelhart, PDP Research Group, et al. *Parallel Distributed Processing*, volume 2. MIT press Cambridge, MA, 1986.

[56] LJ McIntyre. The Practical Skeptic: Core Concepts in Sociology (p. 304), 2011.

[57] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, 2013.

[58] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[59] Angie L Miller and Amber D Lambert. Open-Ended Survey Questions: Item Non-Response Nightmare or Qualitative Data Dream? *Survey Practice*, 7(5):2859, 2014.

[60] Tom Mitchell. *Machine Learning*. McGraw Hill Burr Ridge, 1997.

[61] Antonio Morales, Eris Chinellato, Andrew H Fagg, and Angel P del Pobil. Active Learning for Robot Manipulation. In *ECAI*, volume 16, page 905, 2004.

[62] Faith Wavinya Mutinda, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. Semantic Textual Similarity in Japanese Clinical Domain Texts Using BERT. *Methods of Information in Medicine*, 60(S 01):e56–e64, 2021.

[63] Md Nasir, Sandeep Nallan Chakravarthula, Brian Baucom, David C Atkins, Panayiotis Georgiou, and Shrikanth Narayanan. Modeling Interpersonal Linguistic Coordination in Conversations using Word Mover's Distance. *arXiv preprint arXiv:1904.06002*, 2019.

[64] Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. Learning Text Similarity with Siamese Recurrent Networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157, 2016.

[65] Masaki Oguni, Yohei Seki, and Yu Hirate. Character 3-gram Mover's Distance: An Effective Method for Detecting Near-duplicate Japanese-language Recipes. *arXiv preprint arXiv:1912.05171*, 2019.

[66] Jesús Oliva, José Ignacio Serrano, María Dolores Del Castillo, and Ángel Iglesias. Symss: A Syntax-Based Measure for Short-Text Semantic Similarity. *Data & Knowledge Engineering*, 70(4):390–405, 2011.

[67] Bhumika Pahwa, S Taruna, and Neeti Kasliwal. Sentiment Analysis- Sstrategy for Text Pre-Processing. *Int. J. Comput. Appl*, 180:15–18, 2018.

[68] Nicole Peinelt, Dong Nguyen, and Maria Liakata. tBERT: Topic Models and BERT Joining Forces for Semantic Similarity Detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7047–7055, 2020.

[69] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[70] E Philip. Converse,'The Nature of Belief Systems in Mass Publics'. *Ideology and Discontent*, 206:215, 1964.

[71] Alain Pinsonneault and Kenneth Kraemer. Survey Research Methodology in Management Information Systems: An Assessment. *Journal of Management Information Systems*, 10(2):75–105, 1993.

[72] Gail S Poe, Isadore Seeman, Joseph McLaughlin, Eric Mehl, and Michael Dietz. "Don't Know" Boxes in Factual Questions in a Mail Questionnaire: Effects on Level and Quality of Response. *Public Opinion Quarterly*, 52(2):212–222, 1988.

[73] David MW Powers. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[74] Parminder S Raina, Christina Wolfson, Susan A Kirkland, Lauren E Griffith, Mark Oremus, Christopher Patterson, Holly Tuokko, Margaret Penning, Cynthia M Balion, David Hogan, et al. The Canadian Longitudinal Study on Aging

(clsa). *Canadian Journal on Aging/La Revue Canadienne du Vieillissement*, 28(3):221–229, 2009.

[75] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*, 2019.

[76] Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. Adjusting for Chance Clustering Comparison Measures. *The Journal of Machine Learning Research*, 17(1):4635–4666, 2016.

[77] Andrew Rosenberg and Julia Hirschberg. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, 2007.

[78] Donald B Rubin. *Multiple Imputation for Non-Response in Surveys*, volume 81. John Wiley & Sons, 2004.

[79] Hassan Saif, Miriam Fernández, Yulan He, and Harith Alani. On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter. *European Language Resources Association (ELRA)*, 2014.

[80] Mohammad Nazmus Sakib, Shahin Shooshtari, Philip St John, and Verena Menec. The prevalence of multimorbidity and associations with lifestyle factors among middle-aged Canadians: an analysis of Canadian Longitudinal Study on Aging data. *BMC Public Health*, 19(1):243, 2019.

[81] Priscilla Salant, I Dillman, and A Don. *How to Conduct Your Own Survey*. Number 300.723 S3. Wiley, 1994, 1994.

[82] Geoffrey Sampson. *The 'Language Instinct' Debate: Revised Edition*. A&C Black, 2005.

[83] Alper Sarikaya and Michael Gleicher. Scatterplots: Tasks, Data, and Designs. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):402–412, 2017.

[84] D Sculley. Online Active Learning Methods for Fast Label-Efficient Spam Filtering. In *CEAS*, volume 7, page 143, 2007.

[85] Burr Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[86] Claude Elwood Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[87] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew Lim Tan. Multi-Criteria-based Active Learning for Named Entity Recognition. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 589–596, 2004.

[88] Shahin Shooshtari, Verena Menec, Audrey Swift, and Robert Tate. Exploring ethno-cultural variations in how older canadians define healthy aging: The Canadian Longitudinal Study on Aging (CLSA). *Journal of Aging Studies*, 52:100834, 2020.

[89] Stephen Shum, Najim Dehak, Reda Dehak, and James R Glass. Unsupervised Speaker Adaptation based on the Cosine Similarity for Text-Independent Speaker Verification. In *Odyssey*, page 16, 2010.

[90] Grigori Sidorov, Alexander Gelbukh, Helena Gómez-Adorno, and David Pinto. Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. *Computación y Sistemas*, 18(3):491–504, 2014.

[91] Rohit Singh, Nathan Palmer, David Gifford, Bonnie Berger, and Ziv Bar-Joseph. Active Learning for Sampling in Time-Series Experiments with Application to Gene Expression Analysis. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 832–839, 2005.

[92] Alex Smola and SVN Vishwanathan. Introduction to Machine Learning. *Cambridge University, UK*, 32(34):2008, 2008.

[93] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

[94] Ray J Solomonoff. A new method for discovering the grammars of phrase structure languages. In *Communications of the ACM*, volume 2, pages 20–20. Assoc Computing Machinery 1515 Broadway, New York, NY 10036, 1959.

[95] Tsegaye Misikir Tashu and Tomás Horváth. Pair-Wise: Automatic Essay Evaluation using Word Mover's Distance. In *CSEDU (1)*, pages 59–66, 2018.

[96] Gokhan Tur, Dilek Hakkani-Tür, and Robert E Schapire. Combining Active and Semi-Supervised Learning for Spoken Language Understanding. *Speech Communication*, 45(2):171–186, 2005.

[97] Laurens Van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

[98] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[99] Tanu Verma, Renu Renu, and Deepti Gaur. Tokenization and Filtering Process in RapidMiner. *International Journal of Applied Information Systems*, 7(2):16–18, 2014.

[100] Silke Wagner and Dorothea Wagner. *Comparing Clusterings: An Overview*. Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.

[101] Joe H Ward Jr. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[102] Ross Wilkinson and Philip Hingston. Using the Cosine Measure in a Neural Network for Document Retrieval. In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 202–210, 1991.

[103] Xinhui Wu and Hui Li. Topic Mover's Distance based Document Classification. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pages 1998–2002. IEEE, 2017.

[104] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized Autoregressive Pretraining for Language Understanding. *Advances in Neural Information Processing Systems*, 32, 2019.

[105] C Yones, Georgina Stegmayer, and Diego H Milone. Genome-wide pre-miRNA discovery from few labeled examples. *Bioinformatics*, 34(4):541–549, 2018.

[106] Jia-Jie Zhu and José Bento. Generative Adversarial Active Learning. *arXiv preprint arXiv:1702.07956*, 2017.