Analysis of how semantic similarity methods can be used to test call routing in Interactive Voice Response Systems

Brian Mullins

Student Number: 19225741

A thesis submitted to the University of Limerick in fulfilment of the requirement for the degree of Masters in Artificial Intelligence

Supervisor: Dr. Petar Iordanov

Submitted to the University of Limerick, August 2021.

Abstract

Analysis of how machine learning methods can be used to test call routing in Interactive Voice Response Systems

Interactive Voice Response (IVR) systems are self-service telephony systems that allow a machine to interact with customers. These systems are highly used in contact centers globally as it is an effective way to reduce costs and increase efficiency but studies have shown that customers' experience is affected due to technical and navigational weakness in the system . Testing these systems to find these issues before customer complaints is the ideal scenario but many IVR developers are still manually testing these as there is a lack of automated testing in this field.

This study looks into using NLP and semantic similarity methods to conduct call route testing for IVR systems. The idea being if the user intent does not highly correlate with the subsequent branch in the IVR then we can assume it is not correct. For example. The IVR menu provides me with 2 options, ‘Press 1 for sales’ and 'Press 2 for support”. If the user intent is for sales but the subsequent branch has no words related to sales or we get transferred to the support route then they are dissimilar and can assume they are incorrect. Basically the idea is to mimic human interpretation of the IVR routing systems to make sure the route is correct and that the IVR menu is designed well.

Five pre-trained models were used with three different semantic similarity metrics to calculate the semantic similarity between the user intent and the subsequent branch on the IVR system. Threshold values ranging from 50% to 90% in intervals of 5% were used to calculate the accuracy of these methods.

The study concluded that semantic similarity measures can be used for call route testing with the best performing model was GloVe achieving an 86% accuracy when the semantic similarity threshold is set to 75%.

Declaration

I hereby declare that the work contained in this thesis is my own, and was completed without collaboration or assistance from others, other than my supervisor, Dr. Petar Iordanov, from the Department of Electronic and Computer Engineering, University of Limerick. This work has not been submitted to any other University or Higher Education Institute, or for any other academic award with this University.

_____

Brian Mullins

_____

Dr. Petar Iordanov

Acknowledgments

Firstly, I would like to extend my sincere thanks to Dr. Petar Iordanov for supervising me and providing me guidance when required along this study.

Next, I would like to thank Arash Jooracbchi for providing me with some technical advice early on in this project. This advice put me in the right direction.

I would like to thank all of my colleagues and lecturers I have met in the MSc. in Artificial Intelligence. Our collaborations and feedback provided on each other's work helped tremendously.

I would like to also thank my employers, Spearline, for giving me the opportunity and for providing me with the support needed to complete this course.

Finally, I would like to thank my family and friends for their encouragement and support, especially to my fiance Katie who always believed in me when there were times I did not.

Table of Contents

List of figures

Chapter 1 - Introduction

1.1. Background

Customer experience is a subjective response that customers have to any interaction with an organisation. These interactions occur in person or over the telephone and represent an organisation's product, service, and brand. When a customer is buying a product or service, they will have an experience whether it's good or bad (Carbone and Haeckal, 1994). A consumer report carried out by Twilio in 2017 shows that "67% of customers will give more business to a company as a result of positive communication experiences. But after a poor communication experience, 38% of customers will switch to a competitor or cancel orders or services, 66% will tell a friend about their experience, and 41% will stop doing business with the company altogether" (Bridging the Customer Communication Divide - Twilio 101 - Twilio, 2017).

Interactive Voice Response (IVR) systems were first introduced in the 1980s and have become a crucial part of this customer experience in the industry since. These are specialised self-service systems that are used in telephony networks that allow a machine to interact with customers through pre-recorded voice messages. Companies make use of this technology as it is an effective way to reduce costs and increase efficiency within their call centre while improving their customer service levels (Khudyakov et al., 2010).

Despite these benefits, studies have shown that IVR systems have technical and navigational weaknesses that defer customers from using them effectively (Matto, 2020). According to a study called "*What's wrong with IVR self-service*", which is considered to be the first study on this topic, a survey was conducted and the results showed that customers consider transactions through an IVR system take longer and are less customisable compared to a human operator and that they believe IVR systems were developed to benefit the company more than customers (Dean, 2008). They are often considered as frustrating and time consuming (Corporation, 2020) which results in a less than satisfactory customer experience, loss of sales and creates a negative impact on the organisation.

For these reasons, proactively testing and monitoring IVR systems for organisations plays a vital role in ensuring they provide a high quality service to the customer. It was found "by using the right testing strategies and proven best practices as illustrated in the paper, organizations can

avoid difficulties that can result in financial losses and customer dissatisfaction" (Mittal, 2010). The aim of this research is to evaluate if semantic similarity methods can be used to test call routing in IVR systems, and if so to compare and contrast the performance of the different methods in this domain.

1.2. Research Motivation & Objectives

The motivation behind this research stems from working in the telecommunication industry where my employer tests and monitors their clients IVR systems and reports any issues found. Currently this is a very manual and labour intensive process in terms of testing where we dial a number through our telephony software , we connect to the clients IVR prompt where it would state for example "press 1 for sales, press 2 for support" etc.. We would record this part of the IVR and have an employee manually transcribe the recording and report issues, if any. From there the employee directs it to the next branch of the client's IVR system through a user interface where the number would be automatically redialled and be navigated through the same IVR path until we reach the next branch to test and we repeat the cycle until we reach an agent..

If one can use NLP and Machine Learning techniques to test, document IVR systems and report back to contact centres this takes the pain point away from companies and would improve testing efficiency significantly, reduce labour costs and human error. Also we don't have to rely on clients giving us the information on the structure of their IVR system, which has been problematic in the past.

So the end goal for this research would be to develop an intent classification model using semantic similarity methods to test if the call routing between IVR branches is correct. The specific objectives of this research are as follows:

1. To explore semantic similarity methods for this type of application.
2. To evaluate which methods yield the best performance.
3. To determine the feasibility of conducting IVR call route testing using semantic similarity methods.

1.3. Overview of Theses Structure

This research paper consists of four chapters:

1. Introduction
2. Literature Review
3. Specification and Design
4. Conclusion and Future Work

The literature review will demonstrate my knowledge and understanding of the current literature and the state of research around IVR testing and different semantic similarity methods used. This will show who were the people that have done this research before, what they have said and the state of their arguments.

The specification and design chapter outlines what dataset was used during this study, how it was sourced, the implementation of each semantic similarity method with the corresponding result.

The conclusion and future work the researcher will analyse the results gathered and provide a final overview of the key findings based on those results with recommendations for future work in this field of study.

Chapter 2 - Literature Review

2.1. Call Route Testing in Interactive Voice Response Systems

To understand how to test call routing we firstly need to understand how IVR systems are used in industry. Figure 1 shows how the service chain in the call centre works. Customers dial the contact centre number and will connect to the IVR system where they will be prompted with options from the IVR menu. Here they can either input dual-tone multi-frequency (DTMF) tones from their dial pads or use voice commands to get to the service they require. This service should be sufficient for most calls but the remainder of calls will be put on hold in a queue system called Automatic Call Distributor (ACD) where they will be routed to the next available human agent (Colladon et al., 2013) .

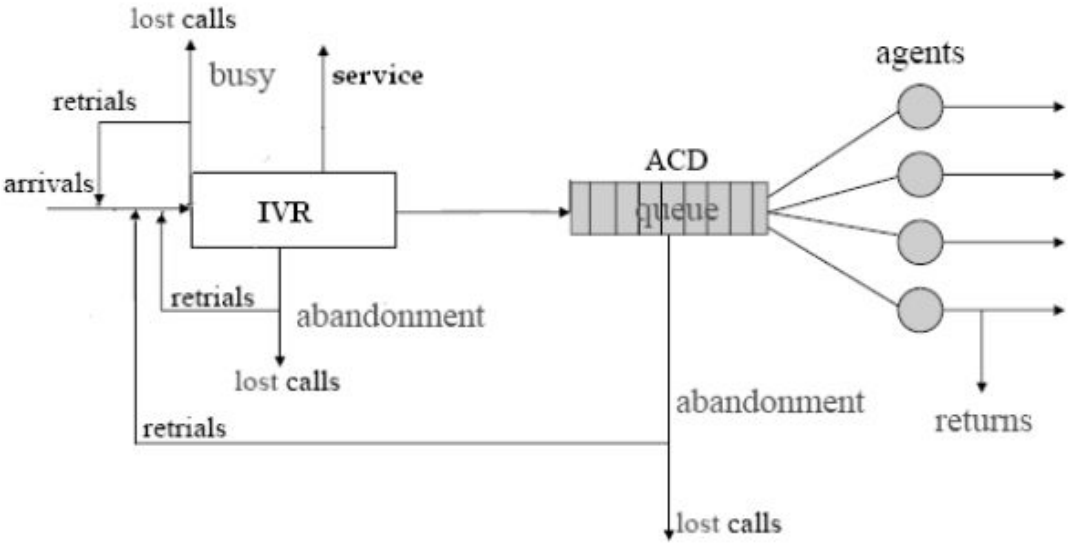

Figure 1 - Service Chain in Call Center (Colladon, Naldi and Schiraldi, 2013)

At any stage of the call flow a customer can abandon the call. One of the main reasons for these abandoned calls is due to poor navigation (Buesing et al., 2020, Mead, 2020) which is also referred to as call routing. If call centres are experiencing high transfer rates or low self-service utilisation, the problem may be that your customers are not behaving as your IVR designers hoped they would and therefore your IVR is failing to deliver them to the right place.

One way for call centers to check for navigational weaknesses is by conducting call route testing on their IVR system to make sure the IVR is navigating the user to the right location. After reviewing the literature Ashthana and Singh have been the main researchers in this field of automatically testing IVR systems. They present two solutions to emulate calls and mimic user behaviour, MockTell (Ashthana and Singh, 2014) and the Maareech (Ashthana and Singh, 2015), with MockTell being the preliminary design of Maareech. They argue in these papers that there is a lack of automated usability testing tools for voice-based systems and that many developers are still doing manual testing or writing customised test scripts for each IVR application which is costly and time consuming. However, Maareech's downfall is that it cannot be used for verification of voice and sound quality.

2.2. Natural Language Processing and Semantic Similarity

Natural language processing is a branch of AI that explores how machines can read, decipher, understand, and make sense of human languages. Assessing the semantic similarity between sentences has been perhaps the most difficult and open research problem in the field of NLP. The diverse context in which sentences can be spoken in languages makes it hard to characterise rule-based techniques for calculating semantic similarity. From my research on the literature there are three methods of calculating semantic similarity which are knowledge-based, corpus-based and deep neural networks.

2.2.1. Knowledge-based methods

Knowledge-based semantic similarity methods calculate semantic similarity between two terms from information obtained from underlying knowledge sources such as lexical ontologies/databases, thesauri, dictionaries, etc. The underlying knowledge-based source offers these techniques an organized structure of terms associated with semantic relations where the particular meaning of those terms is taken into consideration. The most widely used lexical database is WordNet (Miller, 1995) however it is limited to just the English language. One lexical database which overcomes this limitation is BabelNet (Navigli and Ponzetto, 2012) which combines WordNet with data available on Wikipedia.

2.2.2. Corpus-based methods

Corpus-based semantic similarity methods calculate the degree of similarity between words using information exclusively derived from large corpora. Word embeddings are used to create the vector representation of the corpora and these vectors retain the underlying relationship between words. Some of the most widely used pre-trained word embeddings will be briefly discussed in the next section.

2.2.2.1. Word Embeddings

Word2vec (Mikolov et al., 2013) is a neural network model developed from Google News dataset. This approach converts words in a vector of fixed size based on the underlying corpus. The length of the vector must be defined prior to training. The longer the vector the more complex the information you can store in this vector but if you have too big of a vector space then this can lead to sparsity, in that there would be regions in the vector space that will have no word vector in them. There are two different models of word2vec, Continuous Bag of Words (CBOW) and Skip-gram.

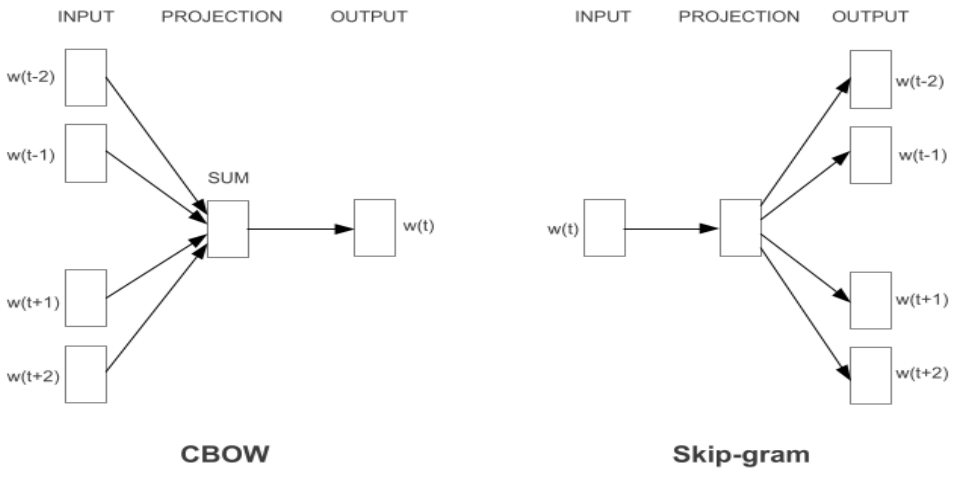

Figure 2 - CBOW vs Skip-gram Model structures (Mikolov, Chen, Corrado & Dean, 2013)

Figure 2 shows how each model is structured. CBOW takes a window of words around a central word while Skip-Gram takes the opposite approach in that the central word is the input and it predicts the surrounding words.. The number of surrounding words is a hyper-parameter which must be set before the user starts training. The neural network must have at least three layers so the input layer would be a one-hot encoded vector the size of the vocabulary and the output is a softmax layer with the same size of the vocabulary. The activations from the hidden layer represent the encoded vector for that word and the embedding with the highest probability then is the predicted word.

A weakness of Word2Vec is that a word can only have one associated vector but there are words that can have more than one meaning. BERT (Devlin et al., 2019) approach can handle this issue as it takes the whole sentence and embeds their meaning. So like Word2Vec, we are interested in the encoding part of the word to produce the embedding but the word embedding is actually a byproduct of BERT as we are encoding the entire sentence. BERT is a big improvement on Word2Vec in the sense the same words can have different meanings but this model has quite a large amount of trainable weights which takes a long time to train.

Another widely used word embedding is GloVe which relies on a global word co-occurrence matrix formed based on the corpus used. It was developed by Stanford University and calculates the semantic similarity based on the fact that words similar to each other occur together. In the paper that introduced GloVe the authors argued that Word2Vec and Glove "*are not dramatically different at a fundamental level since they both probe the underlying co-occurrence statistics of the corpus, but the efficiency with which the count-based* methods *capture global statistics can be advantageous*" (Pennington et al., 2014). After conducting an experiment using the same corpus, vocabulary, window size and training time for both methods they found that GloVe consistently outperformed Word2Vec.

2.2.2.2. Latent Semantic Analysis (LSA)

LSA (Landauer & Dumais 1997) is one of the widely used corpus-based techniques used for measuring semantic similarity between sentences, paragraphs, and documents. A word co-occurrence matrix is created where the columns represent the paragraphs, the rows represent the words, and the cells are populated with word counts. This matrix is made with an underlying corpus, and a technique called Singular Value Decomposition (SVD) carries out dimensionality reduction. Figure 3 illustrates how SVD is represented as a product of three matrices, where two matrices represent the rows and columns as vectors derived from their eigenvalues and therefore the third matrix is a diagonal matrix that has values that would reproduce the first matrix when multiplied by the opposite two matrices. SVD preserves the semantic similarity among the words by reducing the amount of columns while retaining the amount of rows. Semantic similarity is calculated by using the cosine value between the vectors representing the value of the corresponding row for each word.

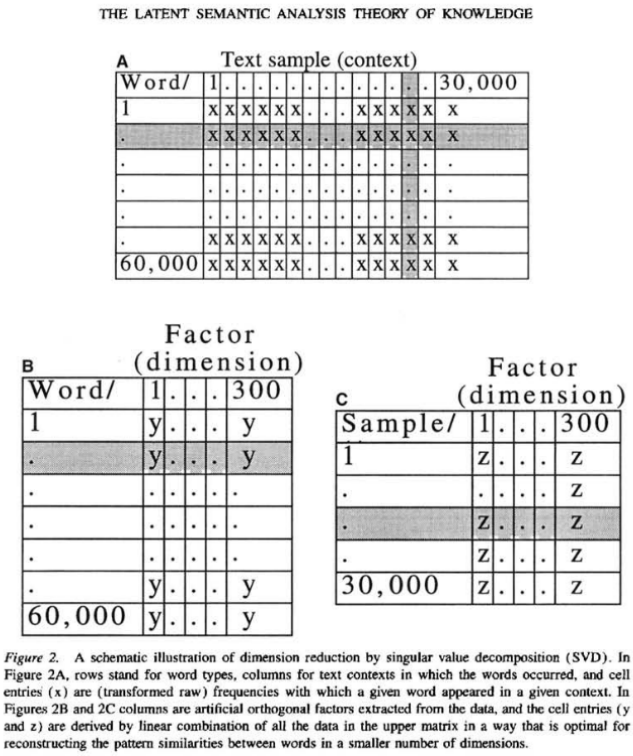

Figure 2. A schematic illustration of dimension reduction by singular value decomposition (SVD). In Figure 2A, rows stand for word types, columns for text contexts in which the words occurred, and cell entries (x) are (transformed raw) frequencies with which a given word appeared in a given context. In Figures 2B and 2C columns are artificial orthogonal factors extracted from the data, and the cell entries (y and z) are derived by linear combination of all the data in the upper matrix in a way that is optimal for reconstructing the pattern similarities between words in a smaller number of dimensions.

Figure 3 - Latent Semantic Analysis Dimension Reduction (Landauer & Dumais 1997)

2.2.2.3. Word Attention Models

Word attention models capture the importance of the words from underlying corpora before calculating the semantic similarity. Different techniques such as word frequency, alignment, and word association are used to capture the attention weights of the text in consideration. Most recently, Attention Constituency Vector Tree (ACV-Tree) (Quan et al. 2019) is an example of a word attention model which based on a tree-like structure where one word of a sentence is made the root and the remainder of the sentence is broken as a Noun Phrase (NP) and a Verb Phrase (VP). Three different attributes of the root word are stored in the leaf nodes. They are the word vectors determined by the underlying corpus, the "modification-relations" of the words in a sentence and the attention weight. In this paper they conducted experiments on 19 textual similarity datasets compared to classic methods and state-of-the-art methods in which 12 out of the 19 datasets showed better performance using their method.

2.2.3. Deep Neural Network Methods

With recent advancements in Deep Neural networks, semantic similarity methods have exploited these to enhance performance. The most widely used techniques include Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM) and Bidirectional Long Short Term Memory (Bi-LSTM). The following section describes some of the deep learning methods used to calculate the semantic similarity between texts. Even though the methods outlined use word embeddings with an underlying large corpus, they are classified differently to corpus-based methods because the networks are used to estimate the similarity between the word embeddings.

2.2.3.1. Long Short Term Memory (LSTM)

LSTM networks (Hochreiter & Schmidhuber, 1997) are a type of Recurrent Neural Network (RNN) used in the field of deep learning. In NLP it is important to understand the context of the textual data provided so it is essential to remember the previous words in a given sentence to capture its context. Simple RNN have the capability to do so however not all previous words have significance over the next which is a downfall for RNN's in this case. Figure 4 shows the LSTM architecture which overcomes this issue by including a memory cell

built of gates that allow the network to choose the content it needs to remember. Figure 4 shows the architecture of an LSTM network.

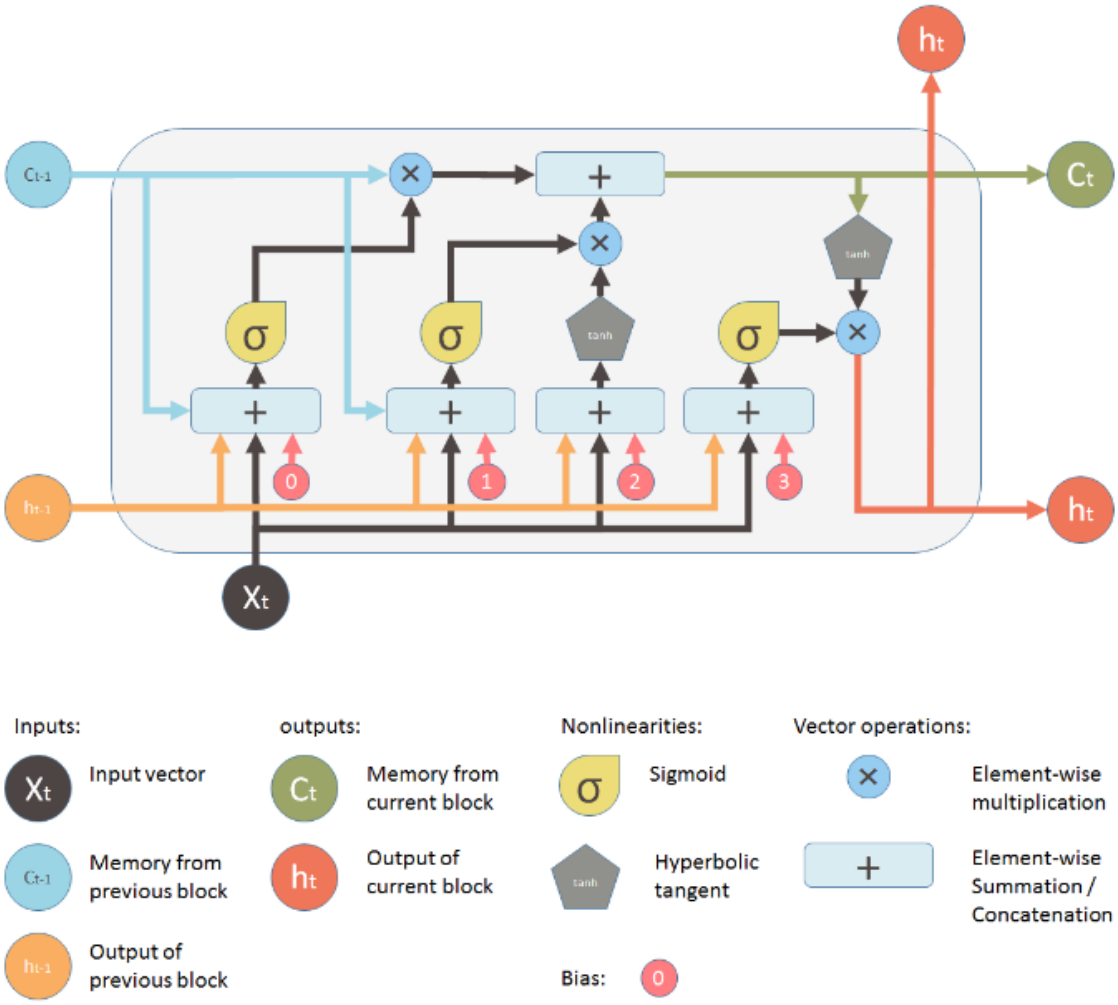

Figure 4 - LSTM Architecture ("Understanding LSTM and its diagrams" 2021)

In recent years, LSTM networks have become the state-of-the-art models for many machine learning problems such as handwriting recognition (Graves et al., 2009), translating languages (Luong et al., 2015) and analysis of audio (Marchi et al., 2014) as well as to measure semantic similarity between blocks of text. A LSTM network was proposed in a paper (Taiet al., 2015) to calculate the semantic similarity between two sentences. Here the sentences were converted into sentence representations using Tree-LSTM over the parse tree of the sentences.

These sentence representations are then inputted into a neural network that calculates the absolute distance between the vectors and the angle between the vectors. More recently a paper proposed a network made up with a LSTM and CNN to form a sentence embedding from pre-trained word embeddings followed by an LSTM to predict their similarity (Tien et al., 2019).

2.2.3.2. Transformer-based Models

In a paper called "*Attention is all you need*" (Vaswani et al., 2017), a transformer model was proposed that relies on attention mechanisms to capture the semantic properties of word embeddings. The transformer is constructed with an autoencoder, that is it comprises an encoder and decoder part. The encoder consists of layers of multi-head attention mechanisms followed by a fully connected feed-forward neural network. The decoder is similar to the encoder with one additional layer of multi-head attention that captures the attention weights in the output of the encoder. This model was originally proposed for a machine translation, however the transformer model was used in a later paper to generate BERT word embeddings (Devlin et al., 2019). From that there were a few more variations of BERT models that surfaced such as TinyBERT (Jiao et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019) and SciBERT (Beltagy et al., 2019), a BioMedical domain-specific variation. From these ALBERT outperformed the rest until another transformer model called T5-11B (Raffel et al. 2019) was later introduced. This was built on a well-defined corpus called "Colossal Clean Crawled Corpus". This method adopted a "text-to-text framework" where an identity token is attached to the input sequence so that it will know what NLP task to be performed. This eliminates the pretraining and fine tuning stages when modelling. Their findings concluded that this model achieved state-of-the-art results by outperforming all other transformer-based models.

2.3. Discussion

Calculating semantic similarity between texts has been a challenging task in NLP with many researchers proposing different methods. This literature review's purpose was to help the reader understand some of the different types and methods used for calculating semantic

similarity between texts, their advantages and disadvantages and the state of IVR testing in a call center environment.

To summarise, the author presented the problem of call routing in IVR systems which can be seen as one of the weaknesses in IVR systems. IVR systems that have poor navigation can result in poor customer service. Even though IVR's are being tested on an infrastructure level, most of the testing to check from a customer perspective are manual and labour intensive. The author's hypothesis for automatically testing call routing in IVR systems is to use semantic similarity methods between IVR branches and if they are above a certain threshold value then it is deemed as the correct route from the user perspective.

The author discussed three different semantic similarity methods, knowledge-based, corpus-based and deep neural network methods. Knowledge-based semantic similarity methods are highly dependent on the underlying corpus which requires them to be updated frequently which in turn takes time and computational resources. Unlike corpus-based systems which are language and domain-independent. This makes these methods easily adaptable across languages, however these methods do not take the meaning of the words into consideration and they need to be built on large corpora which is also time-consuming and resource dependent. Deep learning methods have outperformed most traditional methods with the recent success of semantic similarity using transformer based models. The disadvantages of these methods are that they require large computational resources and are hard to interpret as most deep learning methods come "out of the box" so to speak.

This research is significant as there seems to be a lack of research around the area automatically testing IVR systems in general, especially using NLP or deep learning techniques. The methods that were proposed so far to date, were built using JAVA, thus showing a gap in the research.

Chapter 3 - Specification and Design

3.1. Proposed work

The previous chapter meets objective 1 of this study by exploring the semantic similarity methods from the literature. This chapter focuses on objective 2 which looks to evaluate which methods yield the best performance. For the purpose of this study the author has proposed to use corpus-based methods and transformer-based methods to acquire the word embeddings and then calculate the semantic similarity using three similarity based metrics, cosine similarity, pearson's correlation and spearman's Correlation. Figure 5, shows the proposed implementation for this study.

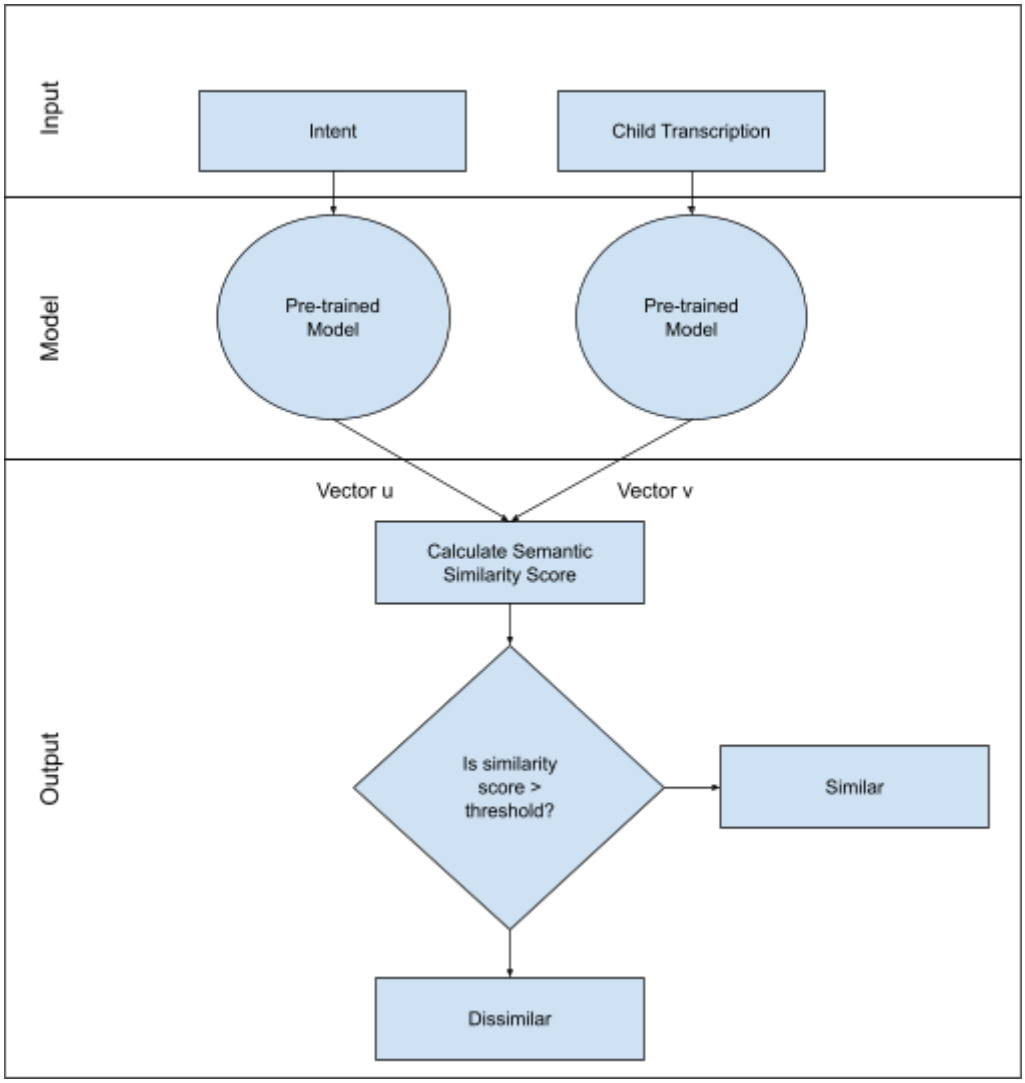

Figure 5 - Proposed implementation

As you can see, there are three parts to this implementation; the input, the pre-trained model and the output. The output will be a similarity score in which threshold values ranging from 50% to 90% will be used to check the results.The following sections in this chapter will outline the steps taken to clean and prepare the IVR input data that will be used for the pre-trained models and the semantic similarity measures used to obtain the results.

3.1.1. Dataset

The dataset was sourced from a database where 15,386 IVR tests have been conducted across multiple languages. Each test has a corresponding audio recording which has been manually transcribed. The dataset consisted of the following fields:

- **child_id** - test id of the current IVR branch
- **parent_id** - test id of the branch where user selected an intent option from IVR menu
- **number** - the clients number dialled to reach their IVR system
- **company** - the company name to which the test was carried out for
- **country** - the name of the country the test was carried out
- **number_type** - the type of number that was dialled e.g toll or toll-free number
- **option** - the option from IVR menu that was selected
- **IVR_traversal** - the path information to reach the current branch of IVR
- **call_start_time** - timestamp of when call was initiated
- **call_connect_time** - timestamp of when call was connected
- **call_end_time** - timestamp of when call was ended
- **child_transcription** - transcription of branch following the option in parent selected by the user
- **parent_transcription** - transcription of parent branch where option was selected
- **description**- gives a description of any issues that were found on test
- **created_on** - timestamp of when the test was created
- **parent_recording** - audio test recording of parent branch
- **child recording** - audio test recording of child branch

Before using the pre trained models for semantic similarity, a number of pre-processing steps to clean up the data were carried out and will be discussed in the next section.

3.1.2. Data Cleaning

Data cleaning is an important step in any machine learning project. The main goal of data cleaning is to handle missing data, to identify and remove any duplicated data or errors in order to create a reliable dataset fit for the model. Figure 6 shows an overview of the dataset used and outlines any missing data that needs to be handled. The following is how the data was cleaned:

- Removed all entries with no parent id as this is the welcome message on IVR
- Values with description equal to null are tests with no issues so impuated these as "*Connected*" description
- Remove empty child transcriptions
- Remove transcriptions from different languages and kept only english transcriptions
- Because of the nature of the business test data could be duplicated as the same number can be tested so remove any duplicate rows that had same child and parent transcriptions

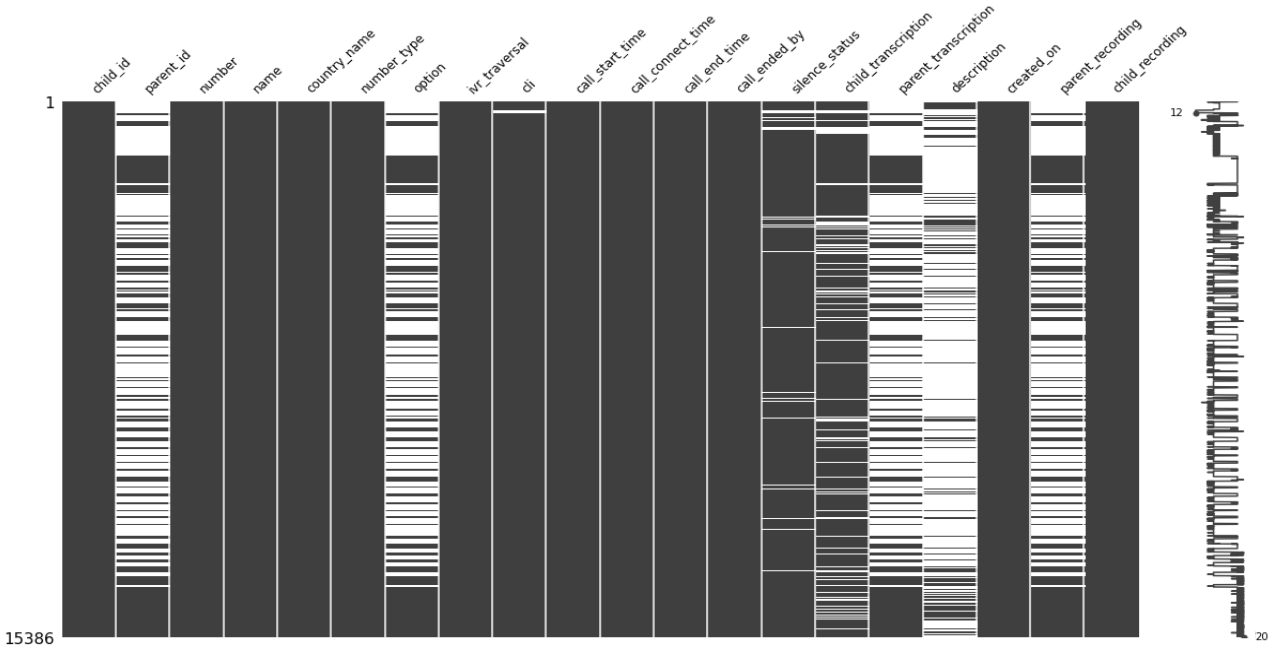

Figure 6 - Overview of dataset

3.1.3. Data Preparation

Once the data is cleaned, the author prepares the data to be inputted into the pre-trained model. Firstly, feature selection is carried out which allows us to narrow our dataset to use only the features we would like. This will be the following :

- child_id
- option
- child_transcription
- parent transcription

Next, the author identifies the intent from the parent transcription using a generic regular expression based on the value set in the option column and creates a new feature called intent to the dataset. Because this was a generic expression it did not capture all of the intents from the cleaned dataset and were left with 926 intents in our dataset.

Lastly, stop words were removed from the intent and child_transction columns which by doing so can help improve the performance as there are fewer and more meaningful tokens left. These two features will be inputted separately into the model to obtain the vector representation of each.

3.2. Pre-trained models

The author proposed to use pre-trained models for this task. This is due to the fact that developing large scale datasets is a great challenge for most NLP tasks and instead of training a model from scratch, the author can use these models to achieve the same or better performance, faster and with a lot less unlabeled data. For the purpose of this study, the author looks at corpus based and transformer-based models as from the literature these methods proved to be most successful.

3.2.1. Corpus-based Models

The two corpus based pretrained models I have used are GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013). These pre-trained models were downloaded using the downloader API module in python's gensim library (Gensim: topic modelling for humans, 2021). The GloVe pretrained model is based on the Wikipedia 2014 + Gigaword dataset, which contains 400k records with 5.6 billion tokens (see appendix B), whereas the word2vec model is based on the Google News dataset, which contains about 100 billion words and contains 300-dimensional vectors for 3 million words and phrases (see appendix C).

3.2.2. Transformer Based Models

For this study the author has used the pre-trained BERT model and variations (TinyBert, RoBERTa and ALBERT) of this model to generate the vector presentations. These were downloaded using the sentence-transformer package in python (sentence-transformers, 2021). These models map sentences and paragraphs to a 768 dimensional dense vector space. Once the models are loaded, the author loops through the IVR dataset, inputs the intent and child description separately into the pretrained model to obtain the vector representation and then calculates the semantic similarity score between the two vectors generated for each test.

3.3. Results

To calculate the semantic similarity between the two vectors (intent and child transcription), the author has used three different metrics to compare, cosine similarity, pearson's correlation and spearman’s correlation.

3.3.1. Cosine Similarity

In information retrieval, cosine similarity is an extensively used metric. By calculating the cosine of the angle between two vectors, this metric determines how similar two texts are regardless of their size. Cosine similarity is 1 when vectors point in the same direction; 0 when vectors are perpendicular; and -1 when vectors point in opposing directions. Cosine similarity is

the complement of cosine distance in positive space: cosine similarity = 1 - cosine distance. The scipy.spatial.distance function.was used to find the cosine distance between vectors **u** and **v**. The formula to calculate cosine distance is as follows:

$$1 - \frac{u \cdot v}{||u||_2||v||_2}.$$

where u · v is the dot product of u and v. To cosine similarity was calculated by subtracting this value from 1. Figure x shows the result gathered from using this metric on all pretrained models.

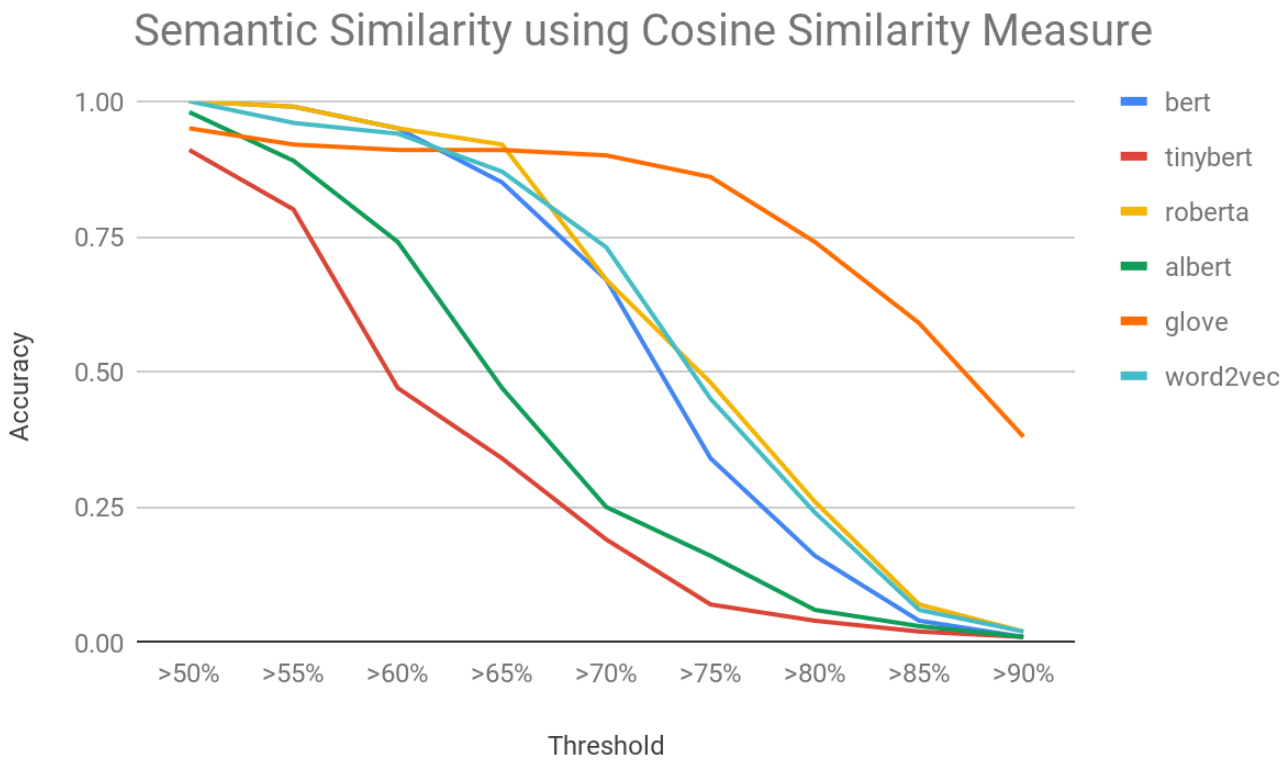

Figure 7 - Semantic similarity using cosine sImilarity

3.3.2. Pearson's correlation

Another similarity based metric is Pearson's Correlation.The Pearson correlation coefficient is calculated by dividing the covariance of two vectors by their standard deviations. This is a parametric metric for determining the strength of a linear relationship between two variables.. It can take a range of values from +1 and -1. Just like cosine similarity, a value less than 0 indicates poor correlation, a value of 0 indicates no association between the vectors and a value greater than 0 indicates a direct correlation. The formula to calculate pearson's correlation is as follows:

$$r = \frac{\sum(x - m_x)(y - m_y)}{\sqrt{\sum(x - m_x)^2 \sum(y - m_y)^2}}$$

where **mx** is the mean of the vector x and **my** is the mean of the vector y. Figure x shows the result gathered from using this metric on all pretrained models.

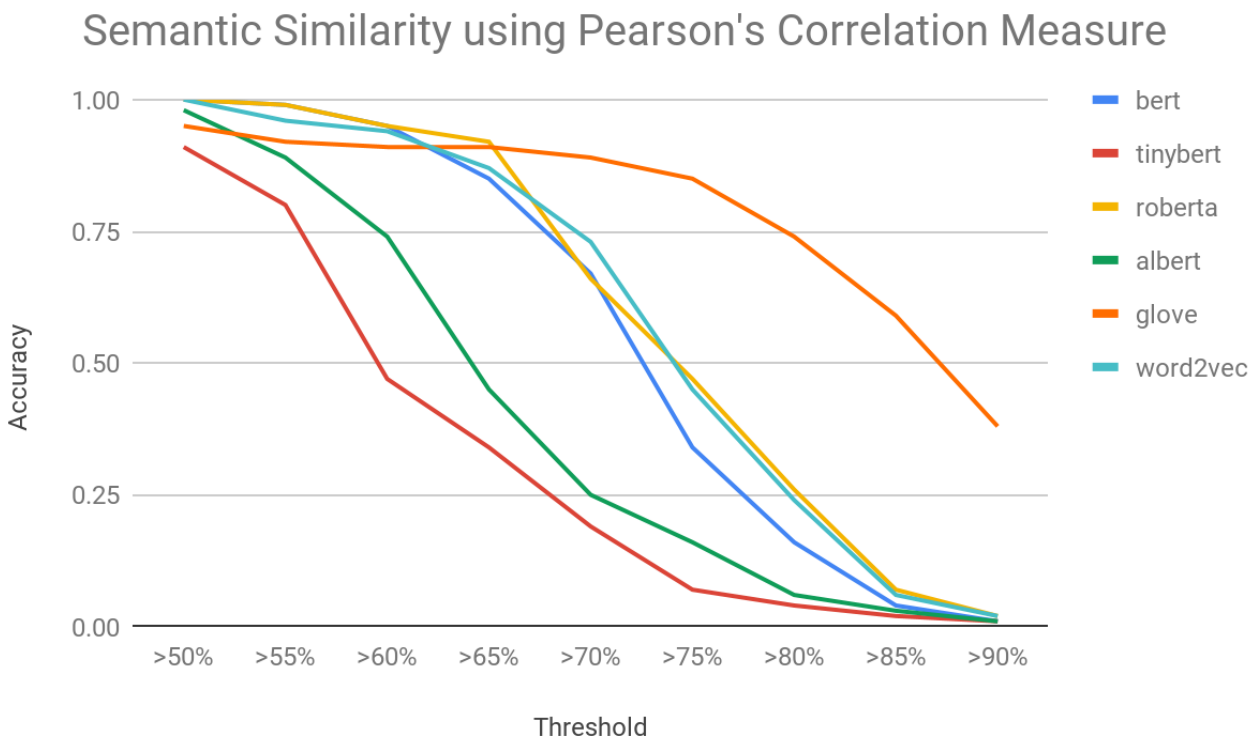

Figure 8 - Semantic similarity using pearson's correlation

3.3.3. Spearman's Correlation

The final semantic similarity metric the author used in this study is Spearman's Correlation which is a nonparametric measure of the monotonicity of a relationship between two variables.. The Spearman correlation does not assume that the data is regularly distributed, unlike the Pearson correlation. This correlation coefficient, like the others, ranges from -1 to +1, with 0 denoting no correlation. A monotonic relationship is implied by a correlation of -1 or +1. Positive correlations imply that as x increases, y will increase as well. Negative correlations indicate that as x increases, y decreases. Figure x shows the result gathered from using this metric on all pretrained models.

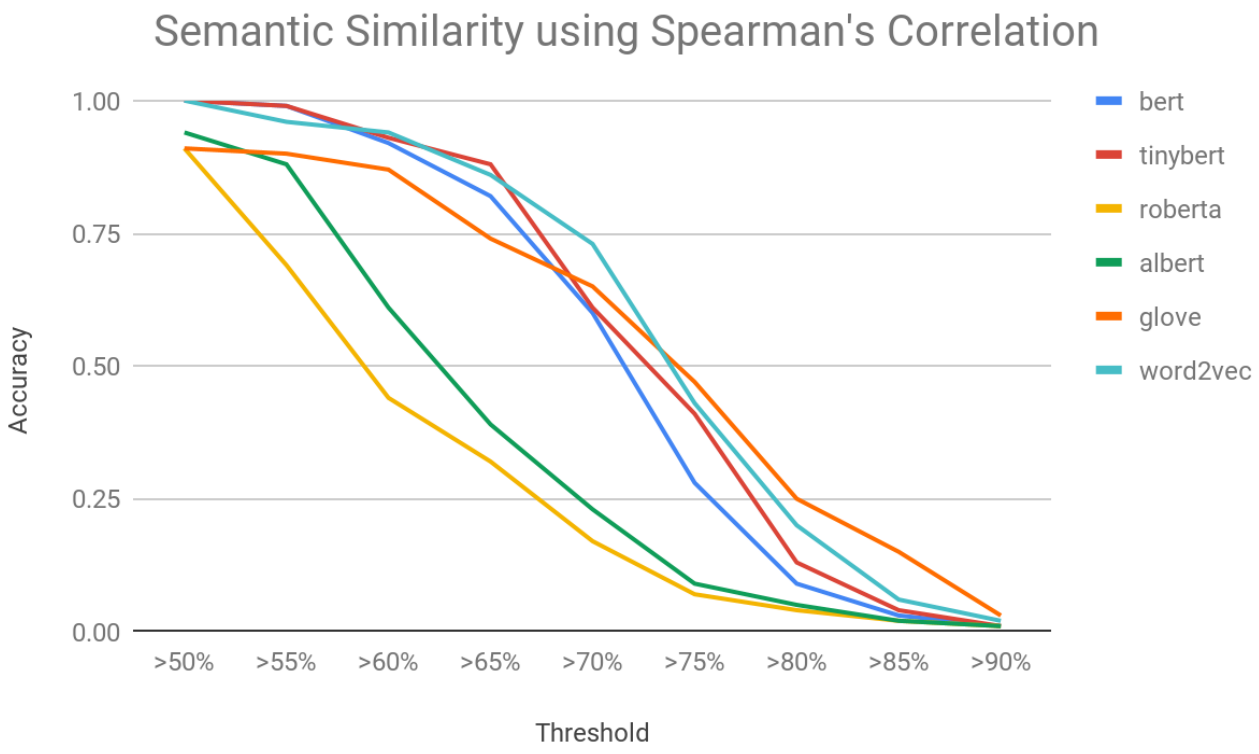

Figure 9 - Semantic similarity using spearman's correlation

Chapter 4 - Conclusion and Future Work

4.1. Outline

In this final chapter, the author will conclude the study. The next section will summarise the key findings from the results of the semantic similarity methods used and section 4.3 will discuss the future work in this area.

4.2. Key findings and conclusion

In the introduction chapter, the author set out three main objectives:

1. To explore semantic similarity methods for this type of application.
2. To evaluate which methods yield the best performance.
3. To determine the feasibility of conducting IVR call route testing using semantic similarity methods.

Objective one was covered in the literature review, where three different semantic similarity methods were discussed; knowledge-based, corpus-based and deep neural networks. From the literature, it was found that transformer based models achieved state-of-the-art results however looking at the results obtained in this study, the GloVe pre-trained model seemed to outperform the transformer-based models when using cosine similarity or Pearson's Correlation as the metric achieving a 86% accuracy when the semantic similarity threshold is set to 75% while with using Spearman's correlation, BERT, TinyBERT and Word2vec model performed better. Glove also had some issues with 34 instances of this which when looking at these occurrences seems to be that the model is unable to infer vectors for unfamiliar words. As IVR systems can be quite diverse and have unfamiliar words I don't think this model would be ideal. The next best model that did not have any issues was the Word2Vec which with all semantic similarity metrics achieved the same performance of 73% with a semantic similarity threshold set to 70%.

One interesting finding is that the results for using cosine similarity and pearson's correlation are nearly identical on this dataset and from researching this pearson's correlation is

considered the equivalent to centered cosine similarity where the inputs, u and v in our case, are centered versions.

From this study, Objective 3 is to determine the feasibility of conducting IVR call route testing using semantic similarity methods in which we can see can be achieved using these methods with reasonable accuracy however there was one issue with this dataset. After cleaning the dataset, it was found that the description for “Incorrect announcement” was only present on tests which only consisted of the welcome message of the IVR and parent transcription was not present and therefore the cleaned dataset consisted of samples of only correctly routed tests.

4.3. Future work

From this study, the author came across a number of thoughts on reflection which are as follow:

Firstly, this study could be refined by having a larger dataset where we have examples of both incorrect and correctly routed IVR tests to determine its feasibility to conduct autonomous call route testing during an IVR test call.

Secondly, due to lack of time and resources the T5-11B model could not be implemented and from the literature, this outperformed other transformer-based models to achieve state-of-the-art results so it would be of interest to see if this would perform better that other models in this domain specific task.

Thirdly, as this study just focused on the english language only, further research can be conducted to see how semantic similarity methods perform across multiple languages for this domain.

Forthly, the ultimate goal here would be to mimic customer experience. This application would be a part of a larger test suite. For example audio quality is crucial for monitoring customer experience in modern telecommunication networks. Subjective audio quality measurement is time consuming, expensive and not suitable for test automation in audio processing. Therefore an non-intrusive means of carrying out audio quality testing is required.

Lastly, with the recent introduction of Intelligent Virtual Assistants (IVA) it's quite possible that contact centers will move from traditional DTMF IVR systems to IVA systems therefore the need to test and monitor customer experience will be required for these systems as well.

References

Ashthana, S., & Singh, P. (2014). *Automated Testing of Interactive Voice Response Applications*. http://www.call-center-tech.com/ivr-tests.

Asthana S., Singh P. (2015) Maareech: Usability Testing Tool for Voice Response System Using XML Based User Models. In: Marcus A. (eds) Design, User Experience, and Usability: Design Discourse. Lecture Notes in Computer Science, vol 9186. Springer, Cham. https://doi.org/10.1007/978-3-319-20886-2_10

Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3615–3620. http://arxiv.org/abs/1903.10676

Buesing, E., Gupta, V., Kleinstein, B. and Mukhopadhyay, S., 2020. Getting The Best Customer Service From Your IVR: Fresh Eyes On An Old Problem . [online] Available at: <https://www.mckinsey.com/business-functions/operations/our-insights/getting-the-best-customer-service-from-your-ivr-fresh-eyes-on-an-old-problem> [Accessed 13 July 2020].

Chandrasekaran, D., & Mago, V. (2021). Evolution of Semantic Similarity-A Survey. In *ACM Computing Surveys* (Vol. 54, Issue 2, p. 41). Association for Computing Machinery. https://doi.org/10.1145/3440755

Carbone, L.P. and Haeckal, S.H (1994). “Engineering Customer Experience.” Marketing Management 3(3): 8-19.

Colladon, A., Naldi, M. and Schiraldi, M., 2013. Quality Management in the Design of TLC Call Centres. International Journal of Engineering Business Management , 5, p.48.

Corporation, I., 2020. *Consumer Study Finds Overwhelming Dissatisfaction With IVR*. [online] Prnewswire.com. Available at: <https://www.prnewswire.com/news-releases/consumer-study-finds-overwhelming-dissatisfaction-with-ivr-124423133.html> [Accessed 11 July 2020].

Dean, D.H. (2008), "What's wrong with IVR self‐service", *Managing Service Quality: An International Journal*, Vol. 18 No. 6, pp. 594-609.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, *1*, 4171–4186. https://github.com/tensorflow/tensor2tensor

Radimrehurek.com. 2021. *Gensim: topic modelling for humans*. [online] Available at: <https://radimrehurek.com/gensim/downloader.html> [Accessed 3 August 2021].

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(5), 855–868. https://doi.org/10.1109/TPAMI.2008.137

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2019). *TinyBERT: Distilling BERT for Natural Language Understanding*. 4163–4174. http://arxiv.org/abs/1909.10351

Khudyakov, P., Feigin, P. and Mandelbaum, A., 2010. Designing a call center with an IVR (Interactive Voice Response). *Queueing Systems*, 66(3), pp.215-237.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. http://arxiv.org/abs/1909.11942

Landauer, T. K., & Dumais, S. T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, *104*(2), 211–240. https://doi.org/10.1037/0033-295X.104.2.211

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. http://arxiv.org/abs/1907.11692

Luong, M. T., Sutskever, I., Le, Q. V., Vinyals, O., & Zaremba, W. (2015). Addressing the rare word problem in neural machine translation. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, *1*, 11–19. https://doi.org/10.3115/v1/p15-1002

Marchi, E., Ferroni, G., Eyben, F., Gabrielli, L., Squartini, S., & Schuller, B. (2014). Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2164–2168. https://doi.org/10.1109/ICASSP.2014.6853982

Matto, G., Ramadhani, A. & Mjema, L. (2020). Optimizing navigation strength of IVR systems through the use of Non-Chronological Pre-Programmed Prompt Approach, *East African Journal of Social and Applied Sciences*, 2(1), 31-37

Mead, J., 2020. *IVR Systems | 11 Problems & How To Solve Them | Inform Comms*. [online] Inform Comms. Available at: <https://inform-comms.com/ivr-system-problems/> [Accessed 13 July 2020].

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, January 16). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*. http://ronan.collobert.com/senna/

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, *38*(11), 39–41. https://doi.org/10.1145/219717.219748

Mittal, A. (2010). Manual Testing and Quality Monitoring of Interactive Voice Response (IVR) applications. In *International Journal of Computer Applications* (Vol. 4, Issue 6). https://www.academia.edu/download/31595265/pxc3871124.pdf

Navigli, R., & Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, *193*, 217–250. https://doi.org/10.1016/j.artint.2012.07.001

Pennington, J., Socher, R., & Manning, C. D. (2014.). GloVe: Global Vectors for Word Representation. In *aclweb.org*. Retrieved June 4, 2021, from http://nlp.

PyPI. 2021. *gensim*. [online] Available at: <https://pypi.org/project/gensim/> [Accessed 3 August 2021].

PyPI. 2021. *sentence-transformers*. [online] Available at: <https://pypi.org/project/sentence-transformers/0.3.0/> [Accessed 3 August 2021].

Quan, Z., Wang, Z. J., Le, Y., Yao, B., Li, K., & Yin, J. (2019). An efficient framework for sentence similarity modeling. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *27*(4), 853–865. https://doi.org/10.1109/TASLP.2019.2899494

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, *21*, 1–67. http://arxiv.org/abs/1910.10683

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, *1*, 1556–1566. http://arxiv.org/abs/1503.00075

Tien, N. H., Le, N. M., Tomohiro, Y., & Tatsuya, I. (2019). Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity. *Information Processing and Management*, *56*(6), 102090. https://doi.org/10.1016/j.ipm.2019.102090

Twilio. 2017. *Bridging the Customer Communication Divide - Twilio 101 - Twilio*. [online] Available at: <https://www.twilio.com/learn/twilio-101/bridging-the-communication-divide-infographic-2017> [Accessed 1 June 2021].

Understanding LSTM And Its Diagrams [online] (2021) Available at: <https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714> [Accessed 10 June 2021].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-December*, 5999–6009. https://arxiv.org/abs/1706.03762v5

Appendices

Appendix A - Intent identification code

```
#Identify the intent on each parent transcription and create intent features based on option value -
# 926 intents/utterances using generic regex expression - if option selected = 1 then (.*)(?=press [option selected
for index, row in df_new.iterrows():
    intent = None
    #print(row['parent_transcription'])
    sentences = row['parent_transcription'].split(".")
    #print(sentences)
    for sentence in sentences:
        #check if press is in sentence then we know its the IVR menu
        if 'press' in sentence:
            if(int(row['option'])==1):
                #print()
                intent = re.search(r"(.*)(?=press 1)",str(sentence))
                #df_new['intent'] =
            else:
                option = int(row['option'])
                previous_option = int(option) - 1
                intent = re.search(r"(?<=press "+ str(previous_option) +")(.*)(?=press " + str(option) + ")",str(se

        if intent:
            clean_intent = intent.group(1).replace(",","").lstrip()
            print(clean_intent)
            df_new.loc[index, 'intent'] = clean_intent
```

Appendix B - Glove-wiki-gigaword-50 info


```
{
    "num_records": 400000,
    "file_size": 69182535,
    "base_dataset": "Wikipedia 2014 + Gigaword 5 (6B tokens, uncased)",
    "reader_code": "https://github.com/RaRe-Technologies/gensim-data/releases/download/glove-wiki-gigaword-50/__in
it__.py",
    "license": "http://opendatacommons.org/licenses/pddl/",
    "parameters": {
        "dimension": 50
    },
    "description": "Pre-trained vectors based on Wikipedia 2014 + Gigaword, 5.6B tokens, 400K vocab, uncased (http
s://nlp.stanford.edu/projects/glove/).",
    "preprocessing": "Converted to w2v format with `python -m gensim.scripts.glove2word2vec -i <fname> -o glove-wi
ki-gigaword-50.txt`.",
    "read_more": [
        "https://nlp.stanford.edu/projects/glove/",
        "https://nlp.stanford.edu/pubs/glove.pdf"
    ],
    "checksum": "c289bc5d7f2f02c6dc9f2f9b67641813",
    "file_name": "glove-wiki-gigaword-50.gz",
    "parts": 1
}
```

Appendix C - Word2vec-google-news-300 info


```
{
    "num_records": 3000000,
    "file_size": 1743563840,
    "base_dataset": "Google News (about 100 billion words)",
    "reader_code": "https://github.com/RaRe-Technologies/gensim-data/releases/download/word2vec-google-news-300/__
init__.py",
    "license": "not found",
    "parameters": {
        "dimension": 300
    },
    "description": "Pre-trained vectors trained on a part of the Google News dataset (about 100 billion words). Th
e model contains 300-dimensional vectors for 3 million words and phrases. The phrases were obtained using a simple
data-driven approach described in 'Distributed Representations of Words and Phrases and their Compositionality' (h
ttps://code.google.com/archive/p/word2vec/).",
    "read_more": [
        "https://code.google.com/archive/p/word2vec/",
        "https://arxiv.org/abs/1301.3781",
        "https://arxiv.org/abs/1310.4546",
        "https://www.microsoft.com/en-us/research/publication/linguistic-regularities-in-continuous-space-word-rep
resentations/?from=http%3A%2F%2Fresearch.microsoft.com%2Fpubs%2F189726%2Frvecs.pdf"
    ],
    "checksum": "a5e5354d40acb95f9ec66d5977d140ef",
    "file_name": "word2vec-google-news-300.gz",
    "parts": 1
}
```

Appendix D - Results

	Cosine Similarity						Pearson correlation						Spearmans Correlation					
Accuracy	bert	tinybert	roberta	albert	glove	word2vec	bert	tinybert	roberta	albert	glove	word2vec	bert	tinybert	roberta	albert	glove	word2vec
>50%	1	0.91	1	0.98	0.95	1	1	0.91	1	0.98	0.95	1	1	1	0.91	0.94	0.91	1
>55%	0.99	0.8	0.99	0.89	0.92	0.96	0.99	0.8	0.99	0.89	0.92	0.96	0.99	0.99	0.69	0.88	0.9	0.96
>60%	0.95	0.47	0.95	0.74	0.91	0.94	0.95	0.47	0.95	0.74	0.91	0.94	0.92	0.93	0.44	0.61	0.87	0.94
>65%	0.85	0.34	0.92	0.47	0.91	0.87	0.85	0.34	0.92	0.45	0.91	0.87	0.82	0.88	0.32	0.39	0.74	0.86
>70%	0.67	0.19	0.67	0.25	0.9	0.73	0.67	0.19	0.66	0.25	0.89	0.73	0.6	0.61	0.17	0.23	0.65	0.73
>75%	0.34	0.07	0.48	0.16	0.86	0.45	0.34	0.07	0.47	0.16	0.85	0.45	0.28	0.41	0.07	0.09	0.47	0.43
>80%	0.16	0.04	0.26	0.06	0.74	0.24	0.16	0.04	0.26	0.06	0.74	0.24	0.09	0.13	0.04	0.05	0.25	0.2
>85%	0.04	0.02	0.07	0.03	0.59	0.06	0.04	0.02	0.07	0.03	0.59	0.06	0.03	0.04	0.02	0.02	0.15	0.06
>90%	0.01	0.01	0.02	0.01	0.38	0.02	0.01	0.01	0.02	0.01	0.38	0.02	0.01	0.01	0.01	0.01	0.03	0.02