WEB CONFERENCE SUMMARIZATION THROUGH A SYSTEM OF FLAGS

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Annirudh Ankola

March 2020

© 2020

Annirudh Ankola

ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE:	Web Conference Summarization Through A System Of Flags
AUTHOR:	Annirudh Ankola
DATE SUBMITTED:	March 2020
COMMITTEE CHAIR:	Christian Ekhardt, Ph.D. Assistant Professor of Computer Science
COMMITTEE MEMBER:	Franz Kurfess, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Jonathan Ventura, Ph.D. Assistant Professor of Computer Science

Abstract

Web Conference Summarization Through A System Of Flags

In today's world, we are always trying to find new ways to advance. This era has given rise to a global, distributed workforce since technology has allowed people to access and communicate with individuals all over the world. With the rise of remote workers, the need for quality communication tools has risen significantly. These communication tools come in many forms, and web-conference apps are among the most prominent for the task. Developing a system to automatically summarize the web-conference will save companies time and money, leading to more efficient meetings. Current approaches to summarizing multi-speaker web-conferences tend to yield poor or incoherent results, since conversations do not flow in the same manner that monologues or well-structured articles do. This thesis proposes a system of flags used to extract information from sentences, where the flags are fed into Machine Learning models to determine the importance of the the sentence with which they are associated. The system of flags shows promise for multi-speaker conference summaries.

Acknowledgments

I would like to acknowledge Cal Poly, San Luis Obispo for providing the environment to learn and develop my thesis and Christian Eckhardt for his guidance and advice throughout my thesis. I would also like to thank Memoria Inc for providing the resources needed to conduct this thesis and Remi Berson, who has been a great advisor in suggesting ways in which I could improve the research that took place.

Contents

List of Tables viii

List of Figures ix

1 Introduction 1

2 Background 3

2.1 Web Conference Apps . . . 3

2.2 Natural Language Processing . . . 4

2.3 Machine Learning . . . 4

2.3.1 Decision Trees . . . 5

2.3.2 Random Forest . . . 5

2.3.3 Naive Bayes . . . 6

2.3.4 Support Vector Machines . . . 7

2.3.5 Multi-Layer Perceptron . . . 7

2.3.6 K-Nearest Neighbor . . . 8

2.4 Tools . . . 9

2.4.1 NLTK . . . 10

2.4.2 Scikit-Learn . . . 10

2.5 External Data . . . 11

2.5.1 The General Inquirer . . . 11

3 Related Works 12

3.1 A System for Natural Language Unmarked Clausal Transformation in Text-to-Text Applications . . . 12

3.2 Predicting Music Genre Preferences Based on Online Comments . 13

3.3 Keeping Meeting Summaries on Topic: Abstractive Multi-Modal Meeting Summarization . . . 13
3.4 Impact Of Automatic Sentence Segmentation On Meeting Summarization . . . 15
3.5 A Keyphrase Based Approach To Interactive Meeting Summarization 15

4 Methodology 17

4.1 Pre-processing . . . 17
4.2 Flagging . . . 18
4.2.1 Overall_Sentiment . . . 19
4.2.2 Contains_Sent_Word . . . 19
4.2.3 Strong_Time . . . 20
4.2.4 Weak_Time . . . 20
4.2.5 Speaker_Convo_Weight . . . 20
4.2.6 Length_Score . . . 21
4.2.7 Outcome . . . 21
4.3 Annotating . . . 21
4.4 Determining Importance . . . 22

5 Experimental Setup 24

6 Results 27

6.1 Experimental Results . . . 27
6.2 Sample Predictions . . . 35
6.3 Analysis of Results . . . 37

7 Future Works 40

7.1 Text and NLP Based . . . 40
7.2 Multimodal . . . 42
7.2.1 Visual . . . 42
7.2.2 Sound . . . 42

8 Contributions 44

9 Conclusion 45

Bibliography 46

List of Tables

6.1 Training and Testing Results 27
6.2 Confusion Matrix Results by Model 33
6.3 Precision and Recall Results by Model 34

List of Figures

2.1 Decision Tree Example 6
2.2 Naive Bayes Formula 7
2.3 Support Vector Machine Visualization 8
2.4 MLP Visualization with 1 Hidden Layer 9
2.5 KNN Example 10

4.1 Sample Block From Original Json 18
4.2 Sample Block From Flagged Json 22

5.1 Formula for Gaussian Naive Bayes 26

6.1 Confusion Matrix for Decision Tree 28
6.2 Confusion Matrix for Random Forest 28
6.3 Confusion Matrix for SVM 29
6.4 Confusion Matrix for Gaussian Naive Bayes 29
6.5 Confusion Matrix for MLP 30
6.6 Confusion Matrix for 11 Neighbor KNN 30
6.7 Confusion Matrix for 21 Neighbor KNN 31
6.8 Confusion Matrix for 51 Neighbor KNN 31
6.9 Confusion Matrix for 101 Neighbor KNN 32
6.10 Formulas for Precision and Recall 33

Chapter 1

Introduction

In today's world, we are always trying to find new ways to advance. This era of technology has allowed people to communicate with others all over the globe, and has given rise to a global and distributed workforce. With the rise of remote workers, the need for quality communication tools have risen significantly. These communication tools come in many forms, and web-conference apps are among the most prominent for the task. There are many web-conference and live video types of apps that exist in today's marketplace, although the majority of them lack the features that would ultimately help optimize them for the user.

The web-conference app, Memoria Inc, aims to help optimize the offerings of traditional web-conference services. The company has developed a platform that is able to generate a high quality transcription from the audio of a web-conference after the conference has ended. This thesis aims to help develop novel ways to summarize the web-conference. Traditional web-conference apps do not engage in post-meeting processes, creating opportunities with regards to what can be done with the data the web-conference generates. The summarization side of web-conference has been relatively untapped, especially since the structure of web-conferences is unlike a lot of texts that are used for summarization. With the recent advances in Natural Language Processing and fields like Machine Learning,

there are new approaches that could be developed to handle the problem of summarization that web-conference platforms face.

Since people already use web-conference tools, the Memoria app will help people save time by automatically transcribing their meeting and providing a summary of it. The users of the app will be able to stay more focused on the meeting, rather than having to take notes on what all took place while attempting to pay attention to what is being said. This feature allows them to save time and have more efficient meetings, which leads to the users having a more productive work environment. Ultimately, the goal of this thesis is to help develop new technologies for a web-conference application that will lead to a more efficient workplace.

Chapter 2

Background

2.1 Web Conference Apps

Web-conference Apps are communication tools that are used for people to have a chance to speak face to face with one another, when they are physically unable to. Modern apps cater to a wide range of applications, and although they are most commonly used for personal conversations and business meetings, their application has expanded into other areas like home security. Most of the bigger players in the market, like Facetime, Zoom, and Webex offer products that allow a user to hold meetings, where they can share the screens, speak face to face, and chat with one another. Often times the users can even record their meeting or be on a call with multiple participants at a given time.

However, as these meetings have participants, or are more important topics are being discussed there rises a need to document the meeting, which some of the current apps lack. Even with recordings of the call, users have to sift through the meeting to review a topic that was discussed, remember where it was in the meeting, and who was talking about it. All of these contribute to less productive meetings as people currently have to take notes and pay close attention to the meetings, rather than presenting information and discussing solutions. Through

automating the transcript generating and providing a generated summary, this dramatically cuts down on the unproductive aspects of the meeting. With these features, users of the app can focus on expressing ideas without fear of missing out on information. The summarization aspect my thesis is focusing on will also help to allow individuals or other relevant parties to understand what took place in the meeting, if they were unable to attend. Overall, the development of these aspects with regards to.

2.2 Natural Language Processing

Natural Language Processing is a field in computer science in which there has been a lot of development in recent years. It covers areas like speech generation, speech understanding and more. With relevance to my thesis, Natural Language Processing, or NLP, includes the field of text processing. Within text processing there are many subfields, but the ones most relevant to my thesis are entity extraction and summarization. Entity extraction generally aims to identify portions of text into categories like organization, person, etc. For the purpose of this thesis extraction is done in a similar manner, but with the categories as flags. Summarization is another field within NLP that is focused around reducing the amount of text present through various techniques. However, it is with noting that most of the summarization work today involves summarizing articles, rather than conversational meetings.

2.3 Machine Learning

Machine Learning is a field in computer science that focuses on developing an understanding from given data and generating predictions on new data. It

encapsulates a variety of models from simple statistic based models to those that learn from the data. There are supervised and unsupervised and several categories in which the models can fall. In general, the majority of machine learning models are designed in a similar fashion where they are fed in training data which consists of variables and their labels. The model gets fit to this data, and then the testing data is fed in. The accuracy of the model is determined by comparing the predicted labels of the testing variables with their true labels. With regards to this thesis there were five different models that were used.

2.3.1 Decision Trees

Decision Trees are a type of supervised learning model that operates by learning rules from the data. They are simple to understand, require little process and run quickly. Unfortunately, they are prone to overfitting in that they do not generalize data well, they are unstable because small variations may create entirely new trees, and among other issues are prone to bias if the distribution of the classes is uneven [1]. A sample for the structure of what a basic decision tree looks like is as follows [2]:

2.3.2 Random Forest

Random Forests are constructed in a manner similar to Decision Trees but constructions of the tree may involve randomly subsampling from the set of features, and building trees where the samples have been drawn with replacement [3]. This helps to control some of the variance that is associated with trees, and alleviate some of the overfitting problems that may occur. However, this variance reduction may make the model biased with regards to the features that are

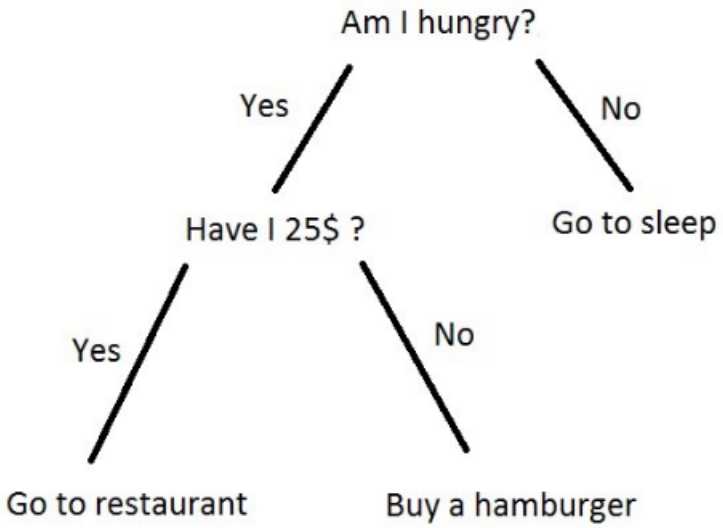
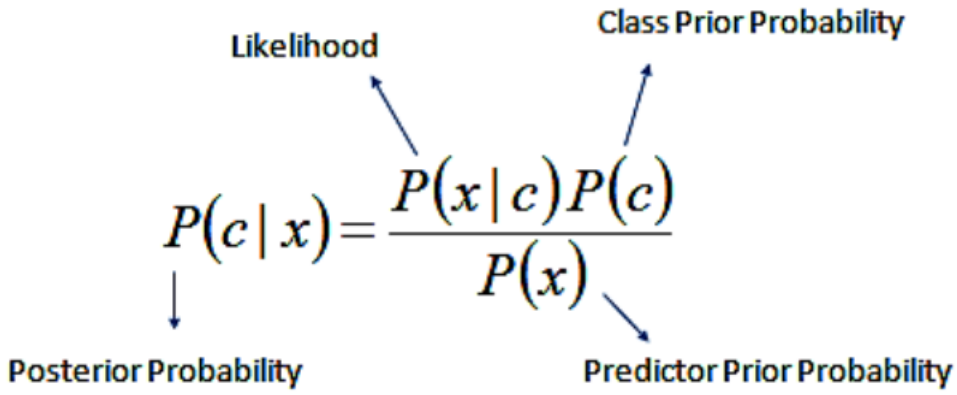


Figure 2.1: Decision Tree Example

selected. Compared to traditional Decision Trees, Random Forests tend to make a better model.

2.3.3 Naive Bayes

Naive Bayes is a statistically based supervised learning model that operates on the principle of conditional independence. It looks at the probability of a class occurring given a set of features. The model is simple, fast and does not need a large amount of data to yield decent results for classification, although it is not the best for estimation[3]. There are variants within the model where they vary in the distribution for the probability of an event given certain features. The standard formula for the model is as follows [4]:



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 2.2: Naive Bayes Formula

2.3.4 Support Vector Machines

Support Vector Machines are supervised learning models that operate by trying to find the hyperplane between the classes. The hyperplane represents the separation that optimizes the distances between the classes. A good hyperplane will maximize the distance between the classes. SVMs are useful in high dimensional spaces, are memory efficient in how they construct the decision function, and allow for custom kernels. On the negative, they are prone to overfitting in some cases and they don't directly provide probability estimates [5]. A sample of what they look like is as follows [6]:

2.3.5 Multi-Layer Perceptron

The Multi-Layer Perceptron is a type of supervised learning model that operates through a neural network. Unlike some of the other models, where the inputs and outputs are easy to follow, the Multi-Layer Perceptron contains hid-

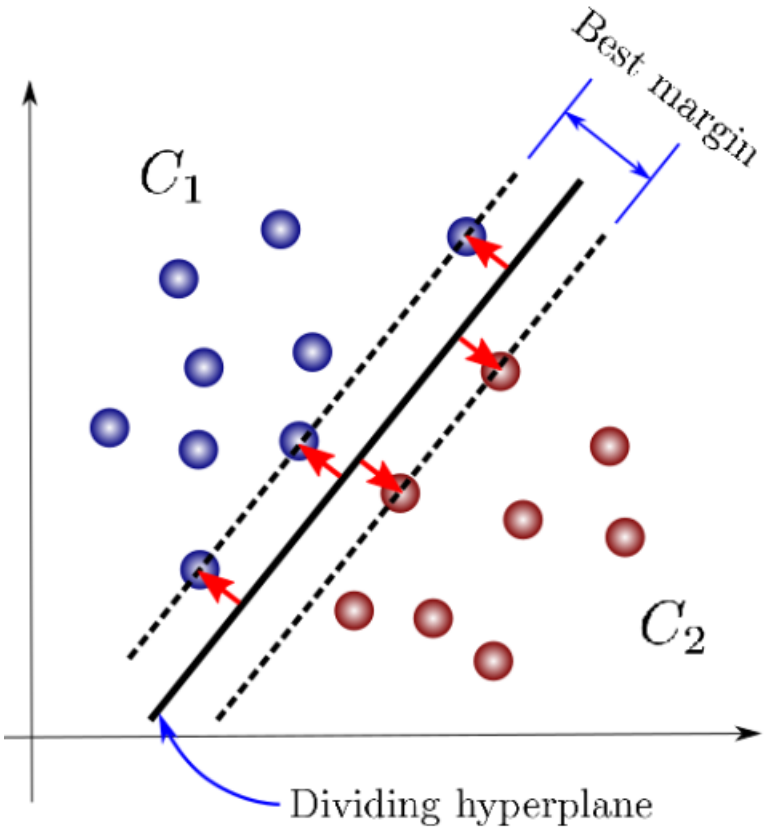


Figure 2.3: Support Vector Machine Visualization

den layers as well. This helps it to learn non-linear models, but are prone to errors in accuracy from the initial random weights that are set and the model also sometimes needs fine tuning for its attributes [7]. The following is what the structure of a one layer model is:

2.3.6 K-Nearest Neighbor

K-Nearest Neighbor is a type of model that isn't a model in the traditional sense. It works by storing instances for the training data and classifies by voting on an instance for testing by looking at the nearest neighbors for the input. There is no standard optimal for the number of k neighbors that are used for voting.

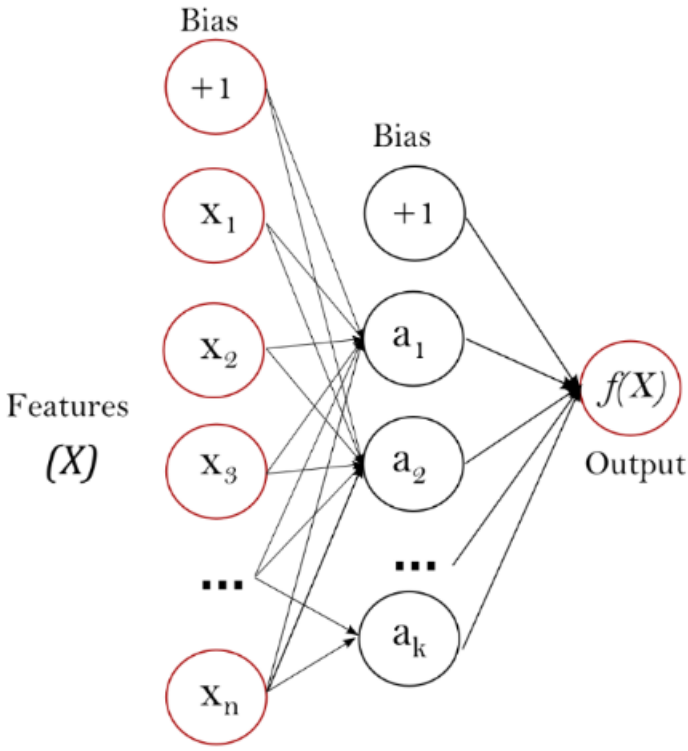


Figure 2.4: MLP Visualization with 1 Hidden Layer

Therefore when using this model some experimentation in the number of neighbors is needed to ensure that the noise is properly reduced but the boundaries are still distinct[8]. In the sample diagram below, once can see how the class of the prediction can change depending on the number of neighbors. In this case if the k is 3, the prediction will be set to Class B, but if the k is 7 the class will change to Class A [9].

2.4 Tools

The following tools were used at various stages of development for this thesis.

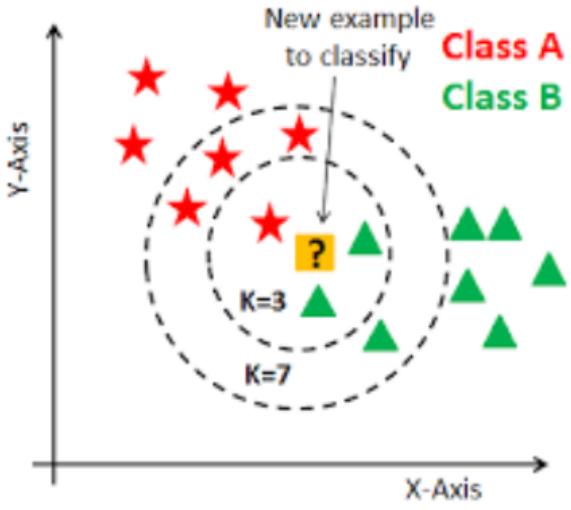


Figure 2.5: KNN Example

2.4.1 NLTK

NLTK is a Python based tool kit that helps people solve problems in Natural Language Processing. It includes corpora, along with libraries for things like classification, tokenization, stemming, and more [10]. It also includes the VADER library within it, which allows for sentiment detection.

2.4.2 Scikit-Learn

Scikit-Learn is one of the most common tool kits used for machine learning. It is a python based library that hosts a wide variety of models to approach challenges related to predictive data analysis. It includes modules for areas including preprocessing, dimensionality reduction, model selection, classification, regression, and clustering [11]. For this thesis the models used included Decision Trees, Naive Bayes, Support Vector Machines, Multi Layer Perceptron, and K-Nearest Neighbors.

2.5 External Data

Outside of the data that is discussed in this section, all data and datasets used for this thesis were either manually generated or provided by Memoria Inc.

2.5.1 The General Inquirer

One of the flags I develop was based on data provided through the General Inquirer Dataset. The dataset was developed by Harvard which includes a total of over 10,000 words for which informational data has been made available. This data includes 15 different categories ranging from basic sentiment, to over and understatement, to pleasure, pain, virtue, and vice, and more [12]. However, it is worth nothing that not every word has a value for every category. Due to this factor, I looked to select the most saturated category, since it would give the most data to use for flagging. Simply selecting the basic sentiment category resulted in 4200 words that could be screened against the text to generate the flag.

Chapter 3

Related Works

3.1 A System for Natural Language Unmarked Clausal Transformation in Text-to-Text Applications

This thesis was a project that focused on summarizing text. The project went over a variety of methods used for solving problems in NLP including rule based methods, statistical methods, and machine learning methods [13]. Part of speech tagging was used to identify groups of words and look at common patterns in a more rule based method, but were beat out in terms of keeping the sentence's meaning by machine learning approaches. More rule based or hard coded methods still had high error rates. The research also suggests that better results take place when the model grabs clauses over one word tokens. With regards to this project, implementing machine learning approaches appears to be the optimal direction to go with. Given what is available, looking for clauses over individual word tokens when appears to be another practice that can be applied to my thesis.

3.2 Predicting Music Genre Preferences Based on Online Comments

This paper focused on the idea of "convergence", a situation where people tend to speak more like one another when talking about something related[14]. The study looked at comments of music that was hosted on Soundcloud. The rationale behind the study was that each genre of music will experience some degree of convergence in the comments and due to that aspect, a person could look at the comments and determine the genre that the song or piece of music was in. The study achieved accuracies of around 40% when classifying for 8 genres. Of the genres examined, one of them was the "world" genre which while distinct from the others, may throw off the accuracy of the model by being too broad. There would have been some phrases that get classified as "world" since they seem different, but may have been correctly classified if "world" was not one of the options. Overall for the 8 genre model, if the model was just randomly guessing, the accuracy would be around 12.5% not 40%. Therefore, the paper's idea of convergence seems to hold some degree of truth with regards to musical genre, and may hold up in other disciplines as well. In web conferences, organizations may converge in their manner of speech when it comes to discussing deadlines or other important events. A high importance flag can be placed if the transcript is displaying signs of similarity to other transcripts, or in other words convergence, with regards to a deadline or important event.

3.3 Keeping Meeting Summaries on Topic: Abstractive Multi-Modal Meeting Summarization

This paper focuses on natural language processing with relevance to meetings with multiple people. The study aims to use a hierarchy when determining the

summary going from topic sentence to utterance and word [15]. Since meetings have multiple people talking with each other and don't follow the structure that normal articles would, the paper claims traditional extraction is not viable and tends to produce incoherent results. This is in part due to transcripts often having utterances that dont consist of actual sentences. The model mimics human summarization by breaking down what is said to a segment before summarizing. Working in this fashion limits the attention in each segment, allowing for more controlled summarizations.

One of the techniques they use to gauge importance involves looking at the people in the meeting who are not the speaker and tracking their eyes to see what it is they are looking at. If meeting members who are not the speaker were focused on the speaker, it was interpreted as the speaker is saying something important. Their model works to measure the eye gaze and head orientation in every frame, in an effort to determine what they are looking at. The data they used was in a relatively consistent format, where each meeting was 30 minute long and consisted of the same number and type of participants. To serve as a baseline, the researchers used the CoreRank extractive summarization method and the PGN generation model. Their model with the visual consideration performs noticeably better than CoreRank and PGM. It learns to place importance on utterances that receive higher scores based on the visual determinant. Overall the researchers models also construct more coherent sentences than the existing approaches. Their models form complete and natural sentences, while traditional approaches often had fragmentations or other aspects that made them incoherent. Their multi-model approach to summarization that involves taking visual data into account does perform better than the existing models.

Since my thesis involves working with a web-conference app, there is access

to the video data that goes along with the transcript. It could be possible that down the road, some approach to track faces and eye positions could be used in helping to summarize the transcript. This eye tracked score can be factored into a flag that is used in determining the importance of a portion of dialogue. This project also gives ideas for future works by demonstrating the value that multimodal techniques have in meeting summarization.

3.4 Impact Of Automatic Sentence Segmentation On Meeting Summarization

In this study, the authors look at the role that sentence segmentation has with regards to meeting summarization[16]. They separate sentences through a hidden Markov Model and use an extraction technique based on marginal relevance. The results of the computer generated summary are compared to the human generated one. The quality of the summarization is reduced with done systematically, although sentence segmentation is a requirement for automatic extraction. With regards to my thesis, this study indicates that the first step should involve separating the sentence so a system can be put in place to identify if it belongs in the summary . It would be expected that the summarization would not be perfect, there are still likely ways to discover important info in the transcript.

3.5 A Keyphrase Based Approach To Interactive Meeting Summarization

This study was focused on multi-document summarization and employed the marginal relevance algorithm for summarization. The technique they used in-

volved automatically looking for keyphrases that are used to query in the algorithm [17]. This keyphrase approach worked well and outperformed the baseline and centroid based systems. For my thesis, the keyphrase functions like a flag that can be used to search for important strings to include in the summary. It also demonstrates the value that flagging certain words and phrases can have in summarization systems.

Chapter 4

Methodology

The methodology for this thesis consisted of the following steps:

1. Pre-processing
2. Flagging
3. Annotating
4. Determining Importance

4.1 Pre-processing

The data provided by Memoria came in the form of a json file. The file is structured in the blocks where each block consists of the user, their dialogue, a timestamp and a confidence score. With the dialogue provided they have an unfixed number of words and sentences in each instance. In order to maximize the amount of non-important information that could be removed, the first step was to break up the dialogue blocks by sentence. The tokenization of the sentences was done through the function available in NLTK that was designed for that specific purpose. This was done by making a new json file where each block consisted of a single sentence with all the information that was contained in its origin block.

Figure 4.1: Sample Block From Original Json

4.2 Flagging

Compared to traditional summarization techniques that work with one well structured article, conference meetings can have multiple people who are all trying to talk over one another to get their point across. Since there is no one factor that can contribute to designating a sentence as important, this thesis revolves around using a multi-flag approach to extract usable information from each sentence. The flags in place incorporate areas including sentiment, time, speaker dominance and sentence length. There were six flags created for each sentence block and they were placed into a new json file. Within the json the flags included: Overall_Sentiment, Contains_Sent_Word, Strong_Time, Weak_Time, Speaker_Convo_Weight, and Length_Score.

4.2.1 Overall_Sentiment

This flag was created through using the VADER sentiment tool available in NLTK. It analyzes a sentence or phrase and returns a score that can be mapped to a sentiment. For this flag, scores were mapped to one of five categories: Very Negative, Negative, Neutral, Positive, and Very Positive. The rationale behind this being that when someone is speaking, it is likely for them to feel some way about what is being said, and this feeling can be tracked through their dialogue. The more insignificant and unimportant something being said is, the higher the odds are that there will be no feeling reflected in the speech, meaning it will be neutral. Conversely, the stronger a person feels about a particular topic the more likely it is that they will be carried in their speech. Therefore, with this logic neutral speech in phrases can been seen as unimportant, and sentences with very positive or very negative sentiments can be seen as having high importance

4.2.2 Contains_Sent_Word

This flag is another sentiment based flag that uses data from a publicly available dataset. The dataset used is called The General Inquirer which has a set of words tagged with sentiment, emotion, etc. The set contains approximately 4200 words, which with regards to sentiment are flagged as positive, negative, or have no sentiment association. This flag works by looking at the words within a particular sentence and looks to see if any of them show up in the dataset of words with sentiment. If they do, the flag is assigned a "Mark" value and if not it is assigned a "Pass" The rationale is that if a sentence has information that is important it will contain a word within it that has a sentiment associated with it. Since the dataset in use is reasonably large, one can assume that it contains a

majority of common language words that have sentiments. If a sentence was to not flag any commonly used word with a sentiment, it is likely of low importance.

4.2.3 Strong_Time

This flag is a time associated flag that looks for words that are significantly related to time. These "significant time" words include units of time, days of the week, and months of the year. Since the app aims to cater to the professional world, if dates and times pop up it will be captured by this flag. In a corporate environment knowing when things are due or when things are happening is very important, so this flag is one that serves for establishing if a sentence has a high importance level.

4.2.4 Weak_Time

This flag is another time associated flag but rather than look at words that are directly related to a point in time like the Strong time flag does, looks at words and phrases that a related to time (soon, until, etc). It uses the same rationals that identifying time words are important in the corporate setting, but since these time words are not directly related to a point in time, they are not of the highest importance. That being said, they can not be ignored and still display some degree of importance.

4.2.5 Speaker_Convo_Weight

This flag aims to place a weight on the speaker's importance by looking at dominance in their speech during the transcript. Since during a meeting, there is usually one individual or group trying to display their topic to everyone, it is

logical that the presenter will be doing most of the talking. Therefore this flag assigns a score based off how much that person spoke relative to the total number of words spoken in the meeting. The rationale is that a speaker's dominance in the web-conference will be captured

4.2.6 Length_Score

Since the json blocks were separated by sentence in the transcript, looking for those blocks which were longer than average could serve as an indication that important topics are being explained. Shorter sentence blocks have a limited amount of things that can be expressed, and are therefore limited in the scope of importance they can have. For calculating this score, sentences of average length were given a medium importance value. Shorter sentences were scaled lower based on the difference in the number of words in the sentence and that of the average. Longer than average sentences were scaled up by calculating the difference in sentence from the maximum after taking consideration for what the average score was.

4.2.7 Outcome

The following is a sample output after all the flags have been created:

4.3 Annotating

Once the flags were in place and all the sentence segments were separated the next step was to label the data so that it could be used for training and testing purposes. This involved converting the output json into a csv file then going through it and labelling the sentence as being important or not. Importance was

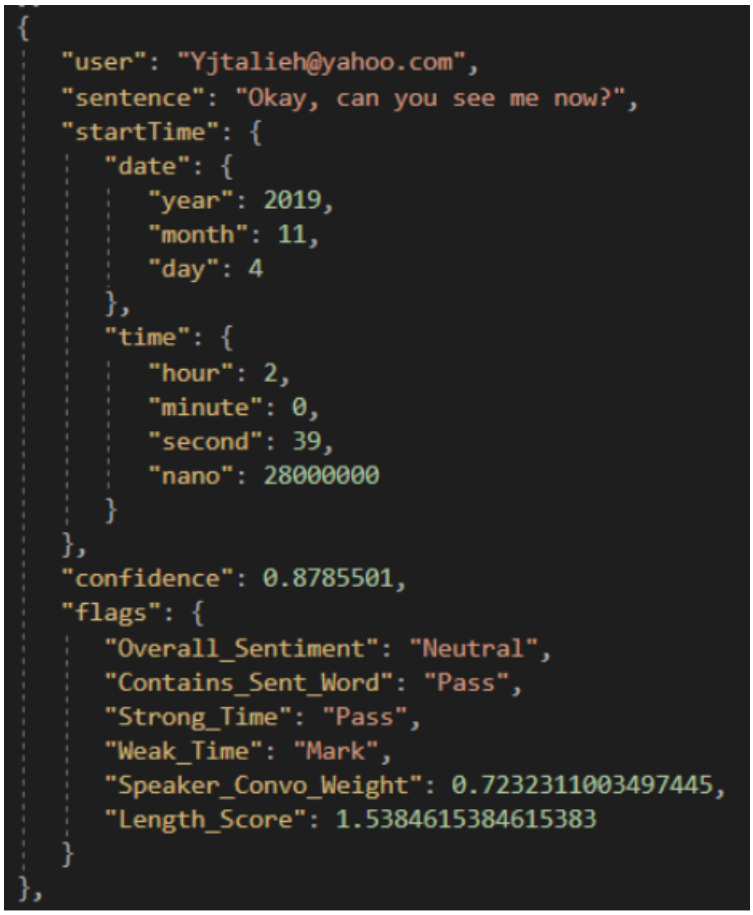


Figure 4.2: Sample Block From Flagged Json

given to those sentences that appear business related where not including them would detract from the message of the meeting. If the sentence was not related to business, did not hold anything substantial, or the message of the meeting could be preserved without it would be classified as non-important.

4.4 Determining Importance

After sufficient data was generated for training the next phase was to determine the importance of a particular string. This was done through using machine learning models that are available through the Scikit-Learn library. From the csv

that was generated by annotation step the data was first processed so that it would be in a format that can be understood by the models within Scikit-Learn. This meant reassigning the strings in the data to numerical values. From there the data was randomized and training and testing splits were created, and the models were run.

Chapter 5

Experimental Setup

For testing out the methodology in place for the thesis, the first step involved selecting a high quality meeting transcription to use. Due to limitations in data available along with not having someone to annotate the transcript, it was important to select a transcript that would be able to provide enough data to train on. To do this I opted to select the largest transcript that was available at the time. After appropriate selection I ran the transcript through the methodology in the previous section.

Once the preprocessing and flagging steps had been completed, there were over 1000 individual sentences with flags that needed to be annotated. These flagged sentences were converted from the json they were structured in, as per the company's request, to a csv so that they could be easily viewed and annotated. The annotation was done manually, where I attempted to include everything around business talk as important and consider everything that wasn't as unimportant. Below are some examples for what I considered to be important vs unimportant phrases.

The following are examples for important phrases:

- At the beginning did it ask for permission from you for your camera and you?
- Coming to Miami and December for a conference for two days.
- So a couple things on this number one, you have to if you want to if you if it's a patient any patient identifying information you have to go through like is AWS HIPAA compliant servers.
- Yeah, so so that's a great that's a great point.

The following are examples for non-important phrases:

- I'll screen share.
- Oh, yeah, dude, David.
- I like the beard.
- Wow, well they have a nice week there to be warm all the kids coming.

After the annotation step was complete, the data was run through a variety of models provided by Scikit-Learn. For the experiment, five different types of models were used and for one of the models it was run with several variants in the parameters. The models used where the Decision Tree, Support Vector Machine, Naive Bayes, Multi-Layer Perceptron, and K-Nearest Neighbors. The standard versions for the Decision Tree and Multi-Layer Perceptron were used but this was not the case for all the models. For the Support Vector Machine, a linear version was employed meaning it uses a "one-vs-rest" strategy when determining the hyperplane. The Naive Bayes model used was a Gaussian one, meaning that

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Figure 5.1: Formula for Gaussian Naive Bayes

the feature likelihood is expected to follow a Gaussian distribution and would employ the following function when classifying and input:

For the K-Nearest Neighbors models, they were run using several different numbers of neighbors, in an effort to get an idea for what the true optimal number of neighbors is. In the experiment, I ran models using 11, 21, 51, and 101 neighbors. After all of the models were run, a simple cross model average was taken for training and testing data to indicate the quality of performance of the models that were used relative to what is average.

Chapter 6

Results

6.1 Experimental Results

Table 6.1: Training and Testing Results

Model	Training Accuracy	Testing Accuracy
Decision Tree	**88.25%**	**73.40%**
Random Forest	**87.62%**	**74.36%**
Support Vector Machine	**79.25%**	**79.17%**
Naive Bayes	**75.50%**	**76.92%**
Multi-Layer Perceptron	**81.25%**	**79.81%**
K-Nearest Neighbor (11 Neighbor)	**81.25%**	**78.53%**
K-Nearest Neighbor (21 Neighbor)	**80.38%**	**79.81%**
K-Nearest Neighbor (51 Neighbor)	**79.75%**	**81.73%**
K-Nearest Neighbor (101 Neighbor)	**79.88%**	**81.41%**
Overall Average	**81.46%**	**78.34%**

In addition to discovering the testing and training accuracy for each model, it is also useful to understand the confusion matrix for each model. The confusion matrix gives information about how the models are predicting, and gives insights on things like the number of true yes predictions, the number of true no predictions, the number of false yes predictions, and the number of false no predictions.

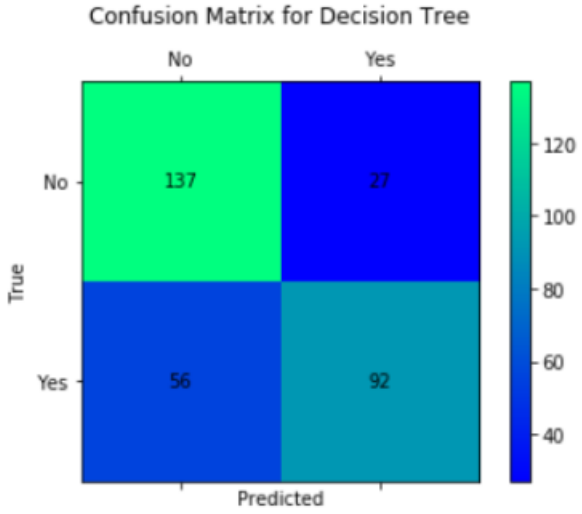


Figure 6.1: Confusion Matrix for Decision Tree

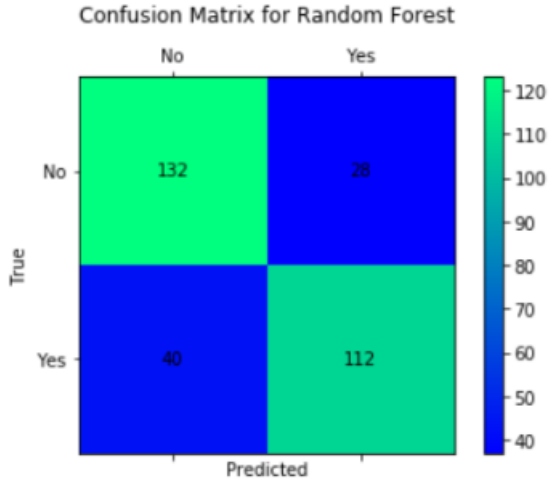


Figure 6.2: Confusion Matrix for Random Forest

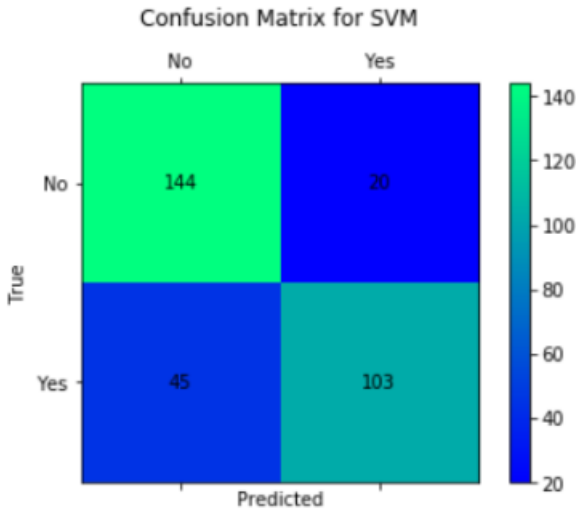


Figure 6.3: Confusion Matrix for SVM

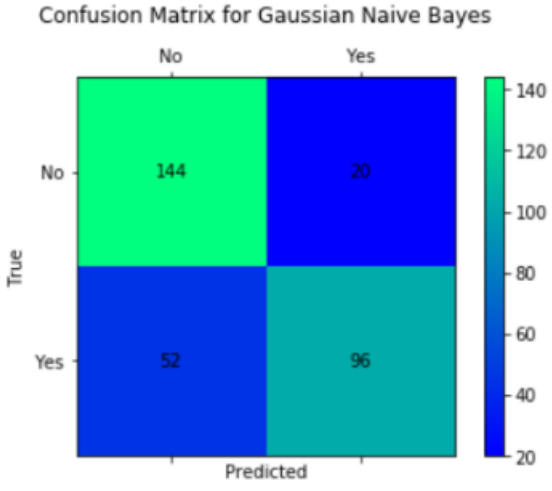


Figure 6.4: Confusion Matrix for Gaussian Naive Bayes

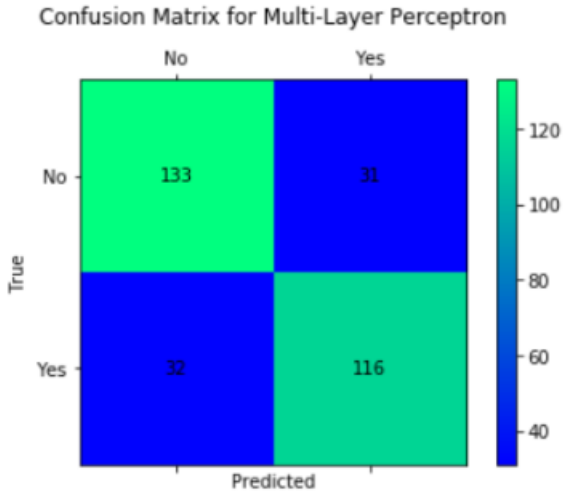


Figure 6.5: Confusion Matrix for MLP

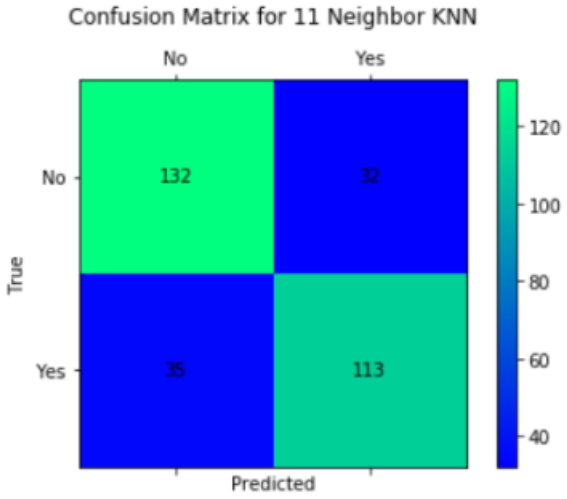


Figure 6.6: Confusion Matrix for 11 Neighbor KNN

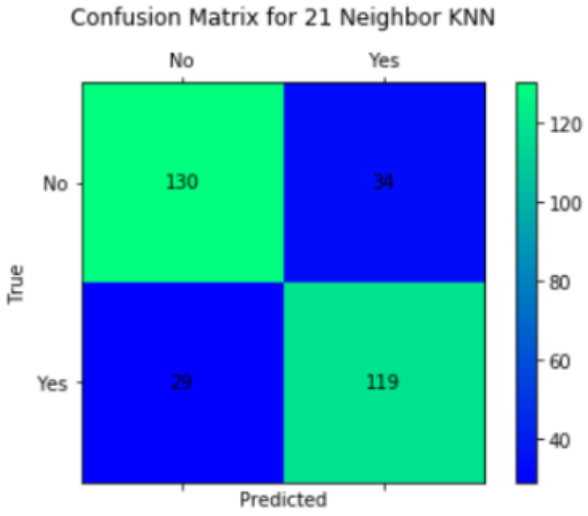


Figure 6.7: Confusion Matrix for 21 Neighbor KNN

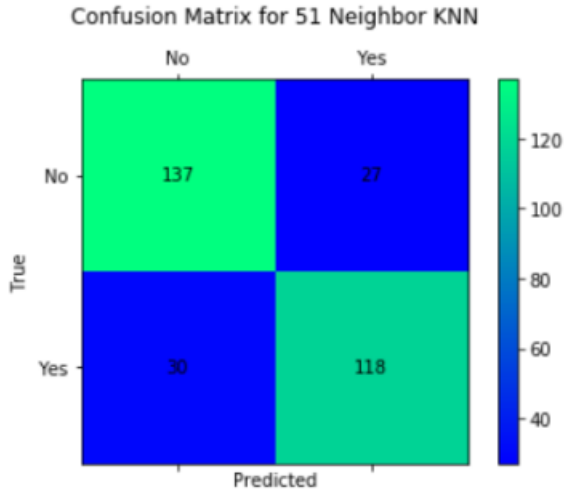


Figure 6.8: Confusion Matrix for 51 Neighbor KNN

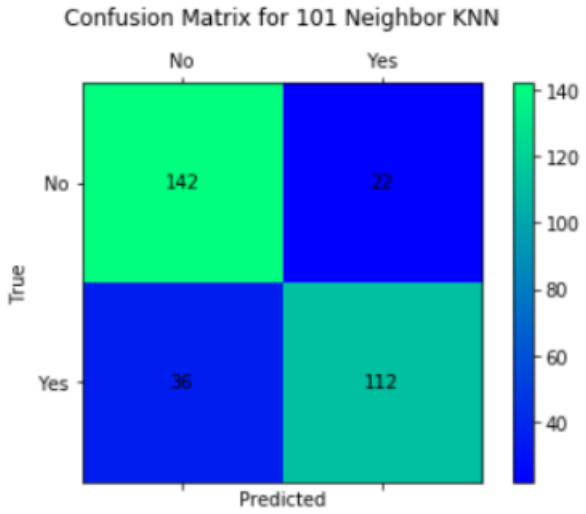


Figure 6.9: Confusion Matrix for 101 Neighbor KNN

In the following table, the summary between model predictions from the confusion matrices can be observed:

Table 6.2: Confusion Matrix Results by Model

Model	True No	True Yes	False No	False Yes
Decision Tree	**137**	**92**	**56**	**27**
Random Forest	**132**	**112**	**40**	**28**
Support Vector Machine	**144**	**103**	**45**	**20**
Naive Bayes	**144**	**96**	**52**	**20**
Multi-Layer Perceptron	**133**	**116**	**32**	**31**
K-Nearest Neighbor (11 Neighbor)	**132**	**113**	**35**	**32**
K-Nearest Neighbor (21 Neighbor)	**130**	**119**	**29**	**34**
K-Nearest Neighbor (51 Neighbor)	**137**	**118**	**30**	**27**
K-Nearest Neighbor (101 Neighbor)	**142**	**112**	**36**	**22**
Overall Average	**136.78**	**109**	**39.44**	**23.33**

From the information that the confusion matrices provide, one can also determine the precision and recall that is associated with each model. These are metrics that are related to positive identifications. Precision aims to determine the proportion of positive identifications that were actually correct while recall aims to determine the proportion of positives that were correctly identified]. High precision implies few false positives, and high recall implies few false negatives They operate by the following formulas:

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

Figure 6.10: Formulas for Precision and Recall

For the models in this thesis the precision and recall is as follows:

Table 6.3: Precision and Recall Results by Model

Model	Precision	Recall
Decision Tree	**.773**	**.622**
Random Forest	**.825**	**.767**
Support Vector Machine	**.837**	**.696**
Naive Bayes	**.828**	**.649**
Multi-Layer Perceptron	**.789**	**.783**
K-Nearest Neighbor (11 Neighbor)	**.779**	**.764**
K-Nearest Neighbor (21 Neighbor)	**.778**	**.804**
K-Nearest Neighbor (51 Neighbor)	**.814**	**.797**
K-Nearest Neighbor (101 Neighbor)	**.835**	**.757**
Overall Average	**.806**	**.738**

6.2 Sample Predictions

The following are sample predictions that fell into one of four prediction options for the 51 Neighbor KNN, which was the best performing model. The four possible predictions are True No, True Yes, False No, and False Yes

Five Examples for True No's:

- Yeah, whatever snap.
- So you're here for me.
- There's a terminology for that.
- Yeah.
- Okay.

Five Examples for True Yes's:

- So and then do it in a way that is for money and legally and paperwork is also in order in the u.s. Again paperwork paperwork paperwork.
- Okay, right if you and then you said get Google Chrome right at the top, it's a chrome for Mac download.
- So so first thing you have to do if it was a patient conversation at be HIPAA compliant and then but the next but let's say assuming that that was okay.
- If you want to try it out if we can get through the HIPAA compliant things like some cardiology group or some Surgery Center or some group.

- Tell Nate you know, he needs to resign and give up whatever because once you do that, he's just going to leverage that and go around and raise money off of the fact that Cerner wants meta phi's and their thing is never going to go anywhere.

Five Examples for False No's:

- I do a lot of telemedicine.
- translator at $150 an hour.
- Yes, maybe I'm a cloud site.
- This would be something like tell a consultation.
- We build a platform.

Five Examples for False Yes's:

- Yeah, you know they should know that they keep on turning off the lights, right?
- She's she's she's still does yoga
- Okay, can you see me now?
- TYes to be envision your sister have a condo right on the beach.
- He's taking so much of your advice throughout the years you actually you know, every every kid has to have a mentor outside of their immediate group so that they realize that hey, you know, what because what you know, I know I take what Dad says with a grain of salt because Dad says, you know,

you know get up early and exercise and then he says, you know, make sure you do this with engineering so they lump it into one, you know stuff.

6.3 Analysis of Results

The models in this thesis all operated with the same category of inputs, which were the system of flags that were created. While they covered several different categories of criteria, there is room for improvement in the form of creating more flags and expanding on the data available for the existing flags. Since this approach of a multi-flag input for summarization is a new and non-traditional way of trying to solve the meeting summarization problem, there are likely more techniques that could have led to better performance.

As for the modelling, the accuracy appeared to be within a reasonably tight range across all models. The training accuracy varied from 75.50% to 88.25% and the testing accuracy varied from 73.40% to 81.73%. It is of note that the model with the highest training accuracy, the Decision Tree, had the lowest testing accuracy. This may be in part due to the fact that Decision Tree models are prone to overfitting the data. As expected, the Random Forest works like a better Decision Tree in testing, through random subsampling of features which decreases variance. The Gaussian Naive Bayes model performed slightly worse than average on testing data, while the Multi-Layer Perceptron performs slightly better. Naive Bayes is a more statistical approach, while the Multi Layer Perceptron employs a neural net and has more opportunities to learn through its hidden layers. Like for many other binary classifiers, the Support Vector Machine also performs well. Since it aims to maximize the distances between the hyperplane and the data instances, the model is able to do a good job at separating the data. The

K Nearest Neighbors also works well, performing better or worse than average depending on the number of neighbors that are used. It appears to perform best in this situation at 51 neighbors, larger or smaller options indicate non-optimal selection where the predictions are being either over or under generalized.

Overall, the average testing accuracy was around 79% percent which indicates the system in place holds some merit. Given that the models are classifying on two classes, which makes them binary classifiers, a truly random prediction would yield an accuracy of 50%. A testing accuracy of 79% implies a performance that is close to 60% better than random, demonstrating that the models are doing their job.

Through looking at some of the sample predictions of the best performing model, there appear to be a few patterns that can be observed in the predictions versus the true values. In the area of true unimportant sentences, a majority seem to be relatively short in nature. Therefore, the model does seem to be able to do a good job at removing the small, irrelevant pieces from the dialogue. The truly important values tend to be captured well when the sentences are long. However, when the strings are short they will often get classified as falsely being unimportant. In terms of strings that are falsely deemed important, they all tended to be longer and involve topics not directly relevant to the meeting. For example, family life, which may be important for the speakers personally, are not important when it comes to summarizing the business elements of a meeting. Adding more flags to capture more categories of words may help to correct some of these sentences that have had their importance labeled incorrectly.

Based on the samples in the predictions, it appears the model currently places the most value on the sentence length score when determining its importance. There is likely some merit to sentence length and importance; however, there are

situations where this is not always going to be the case, leaving room for improvement. The decision tree has a property that allows us to see the importances of each feature, which confirms this belief. The feature importance for the sentence length was greater than .80 out of 1, meaning that at least for the decision tree, the sentence length was the most important feature.

With regards to precision and recall, it is interesting how some of the models, like Naive Bayes and SVM that had precision values higher than average also had recall values lower than average. This indicates that these models, while good at not falsely classifying positives, struggle when it comes to not falsely classifying negatives. Not surprisingly, the best performing model also had precision and recall values greater than average.

Chapter 7

Future Works

Given how this thesis is focused on expanding offerings for an existing application, there remains room for improvement. Developing a more robust methodology can help provide more accurate results and predictions for importance. As with any machine learning based system, expanding the dataset will help lead to more thorough results. With more data comes more variance in structure and content, proving more opportunities for the models to learn what constitutes importance in a sentence.

7.1 Text and NLP Based

Since the system in place is based off flags, there are several other flags that could be constructed to provide better input data for the models to run off of. Additional flags that employ techniques from NLP could be related to identifying business speak, family speak, and variance in vocabulary. The business and family speak flag would operate in a similar fashion to the time and sentiment word flags. To create them, identifying a dataset of words that are related to business would be identified and the sentences would be screen to determine if they contained any words or phrases within the set. The app is focused on web

conferences for businesses, so by that logic, all the sentences that get flagged as containing business words would likely contribute to it being perceived as being important by the models. Conversely, people often engage in talk about their families while on calls, which is not important to understanding the purpose of that meeting. Therefore, by building a dataset of family and related words, the sentences that have them can be identified, giving the model an opportunity to learn about unimportant information as well. For all of the current and future "word identification to flag" type of approaches, there is always an opportunity to expand the dataset used by the flag.

Related to NLP and text based approaches a vocabulary and word variance based score could also be generated. The vocabulary score could look at the complexity of words being used along with if the words used would be considered part of a normal vocabulary. Highly complex or out of normal words would indicate that something very specific to the business is being discussed, and would therefore indicate importance. A word variance score may also be an interesting flag to implement into the system. By looking at the variance within lengths of the word, it would provide insights on the complexity of the sentence. If a sentence appears more complex it could be interpreted as the user attempting to explain something to the other participants in the meeting. If something is non-complex and doesn't need to be explained, chances are that the other people already understand the subject or it simply is not that important.

Outside of the development of more text based flags, other libraries and tools could be used for processing. For processing the text, the spaCy library could be used instead of NLTK, allowing for the development of flags from its features [18]. On that note the Duckling library could also be used to extract entities and other information from text which would provide models more information to use

in their training [19].

7.2 Multimodal

Outside of directly analyzing the text and using NLP style approaches, a multimodal approach could be added to the flagging system.

7.2.1 Visual

Based on research that was discussed in the related works portion, there appears to be promise in using the visual data that is present from the web-conference. The previous study employed eye tracking as a way to help determine if the speaker was talking about something important. Eye tracking of non-speakers can be one way to determine if what is being said is important through visual data. With the recent advances in facial tracking and body tracking, additional visual based flags could be created by looking at emotions in the meeting participants and their body language in relation to who and what is being discussed.

7.2.2 Sound

Since visual data shows promise for summarization, there may be useful information from sound data as well. People tend to try to draw attention to what they are saying when they deem it important, and this may become evident in their volume. The average speaker volume could be taken, and if there are sections where they are speaking louder or softer than normal, that can be used to gauge importance. Also if the noise level of other meeting members are quieter than normal, that can be interpreted as they are paying attention and focused on

what the speaker is saying. Ultimately, there is room for improvement through a wide variety of ideas that can be run in an experimental fashion.

Chapter 8

Contributions

I now present the major contributions of this work:

1. A system of flags for extracting information from text
2. A multi-faceted approach to summarization
3. New methods for meeting summarization
4. Observable patterns within how the models are predicting importance

Chapter 9

Conclusion

Based on the research and experiments that took place in this thesis, it can be concluded there is some merit to creating flags from the information of the sentences spoken during a webconference to determine the importance of the sentence. The information generated in the flagging steps gives the models more to use than traditional, text-only approaches when making their determination. Developing a greater understanding about the feature weights off which the models are basing their decisions will likely lead to better outcomes in the future, but for now, the results of the experiments can serve as a baseline for the summarization project on which Memoria Inc is working.

This thesis serves to demonstrate that a multi-flag approach is a viable and promising option with regards to solving the problem of webconference summarization, since due to the structure of conversation, traditional summarization approaches can not be used effectively. Once research can be done for more flags and approaches to understand the text, it is likely that higher model accuracies can be generated and a more robust and complete summarization for the meeting can be created.

Bibliography

[1] "1.10. Decision Trees," Scikit-Learn, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/tree.html. [Accessed: 27-Feb-2020]

[2] B. Skerritt, "What is a Decision Tree in Machine Learning?," Hackernoon, 10-Oct-2018. [Online]. Available: https://hackernoon.com/what-is-a-decision-tree-in-machine-learning-15ce51dc445d. [Accessed: 27-Feb-2020].

[3] "1.11. Ensemble Methods," Scikit-Learn, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.htmlforest. [Accessed: 27-Feb-2020]

[4] "1.9. Naive Bayes," Scikit-Learn, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html. [Accessed: 27-Feb-2020]

[5] "Naïve Bayes Classifier," UC Business Analytics R Programming Guide. [Online]. Available: https://uc-r.github.io/naive_bayes. [Accessed: 27-Feb-2020].

[6] "1.4. Support Vector Machines," Scikit-Learn, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html. [Accessed: 27-Feb-2020]

[7] O. C. Carrasco, "Support Vector Machines for Classification," Medium, 22-Aug-2019. [Online]. Available: https://towardsdatascience.com/support-vector-machines-for-classification-fc7c1565e3. [Accessed: 27-Feb-2020].

[8] "1.17. Neural network models (supervised)," Scikit-Learn, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/neural_networks_supervised.html. [Accessed: 27-Feb-2020]

[9] "1.6. Nearest Neighbors," Scikit-Learn, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/neighbors.html. [Accessed: 27-Feb-2020]

[10] A. Navlani, "KNN Classification using Scikit-learn," DataCamp , 02-Aug-2018. [Online]. Available: https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn. [Accessed: 27-Feb-2020].

[11] "Natural Language Toolkit," NLTK 3.4.5 documentation, 2019. [Online]. Available: https://www.nltk.org/. [Accessed: 27-Feb-2020]

[12] "scikit-learn Machine Learning in Python," Scikit-Learn, 2019. [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 27-Feb-2020]

[13] "Descriptions of Inquirer Categories and Use of Inquirer Dictionaries," General Inquirer Categories. [Online]. Available: http://www.wjh.harvard.edu/ inquirer/homecat.htm. [Accessed: 27-Feb-2020].

[14]D. S. Miller, "A System for Natural Language Unmarked Clausal Transformations in Text-to-Text Applications," Digital Commons at Cal Poly, Jun-2009. [Online]. Available: https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1149context= [Accessed: 26-Feb-2020].

[15] A. Sinclair, "PREDICTING MUSIC GENRE PREFERENCES BASED ON ONLINE COMMENTS," Digital Commons at Cal Poly, Jun-2014. [Online]. Available: https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=2333context=theses. [Accessed: 26-Feb-2020].

[16] Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke, "Keep Meeting Summaries on Topic: Abstractive Multi-Modal Meeting Summarization," Association for Computational Linguistics, Florence, Italy, 2019

[17] Yang Liu and Shasha Xie, "Impact of automatic sentence segmentation on meeting summarization," 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, 2008, pp. 5009-5012.

[18] "Classification: Precision and Recall — Machine Learning Crash Course," Google, 10-Mar-2020. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall. [Accessed: 03-Mar-2020].

[19] K. Riedhammer, B. Favre and D. Hakkani-Tur, "A keyphrase based approach to interactive meeting summarization," 2008 IEEE Spoken Language Technology Workshop, Goa, 2008, pp. 153-156.

[20] spaCy, "Industrial-strength Natural Language Processing in Python," 2020. [Online]. Available: https://spacy.io/. [Accessed: 27-Feb-2020].

[21] Duckling, "Duckling The linguist that parses text into structured data," Duckling, 2020. [Online]. Available: https://duckling.wit.ai/. [Accessed: 27-Feb-2020].