Predicting Image by Unsupervised Learning with Text

Master Thesis for Data Science: Business and Governance MSc

Academic Year 2016-2018

Student: Yashi Thakkar

ANR: 850035

Supervisor: Dr. A. T. Hendrickson

Second Reader: Dr. Martin Atzmüller

Master's Program: Data Science Business and Governance

Institute: Faculty of Humanities, Tilburg University

Date: September 20, 2018

1 CONTENTS

2 PREFACE

I would like to start with a small phrase: *"We are never really self-made"*. I am thankful to all the people who have been part of my journey and made me a better person every day.

Starting with my family, I would like to thank my parents and my husband Mohit. You have always supported me and encouraged me to go ahead while extending all possible support you could. I cannot thank you enough for all the love and blessings. Mohit, you have been a turning point in my life. You believed in me more than I do in myself and constantly motivated me. I am lucky to have you as my life partner, a companion for life who helps me being a better version of myself. I would also like to thank Mohit's parents for being so loving and supporting all the way along. Vansh, my brother, thanks for all the fun filled memories. I hope you exchange the same line when you write your thesis in future.

I would like to thank Dr. A. T. Hendrickson (Drew) for constantly motivating me. He has been supportive throughout and helped me grow my understanding towards the topic with immense opportunity to try different things. I really enjoyed my discussions with him as he filled it with different ideas. He is a good listener and is open to ideas. This makes it easy to communicate your ideas, concerns, etc. with him. Meanwhile, he will always back up the tasks with small tips to keep you motivated, and his enthusiasm is contagious. I would also like to thank my course coordinator Ms. Marie Postma for guiding me throughout and putting all the pieces together. Not to forget, I would also like to thank Ms. Denise Lindenau, Ms. Laura van Bochove and Ms. Christa Suos for being there for me to discuss various administrative issues and some new facts about Netherlands or the Tilburg University in general.

3 ABSTRACT

This thesis aims at predicting an image based on different texts describing that image and question asked while guessing the image. The most interesting part is that we will predict the image based on the texts only and compare it with the ground truth data, which is the images eliminated by the participant. Our focus will be on text mining with unsupervised learning. The main research question is to evaluate the self-sufficiency of text data to identify a portrait with the help of text mining and similarity measures.

As text-data is unstructured, vector space model transforms each document into a vector with the help of underlying principle that vector space model relies on. We used two types of vector space model combined with two types of decision rules and one similarity measure that is cosine similarity. Vector space model used are count vectorizer that uses the frequency of a word in the document and term frequency-inverse document frequency that uses the term frequency of a word in a document with a combination of how often that word appears in all the documents. Cosine similarity measure generates the distance between two vectors to find the similarity between them. To decide a threshold similarity score for each instance, various decision rules are used. This also helps in gauging the overall effect of similarity measures and decision rules.

All these images are described, and guessed through human intelligence tasks uploaded on a well-known crowdsourcing website, Amazon mechanical turk. These kinds of tasks need human intelligence to solve the problem at hand. Capturing this data in a structured manner creates a foundation for machines to learn on (O'Reilly, 2017). It shall be noted that we are using the choices made by the participants in the Guessing tasks as the ground truth data rather than labels to train our model. Hence, this is an unsupervised learning. Therefore, all the models are first applied on auxiliary data to compare the performance of various models. The best performing model (combination of vector representation, similarity measure, and decision rule) based on the performance is then tested on a left-out set of data. In our research, term frequency and inverse document frequency performed the best with a precision of 49%. In other words, out of 100 images classified as eliminated, 49 images are correct. Furthermore, the recall is also 49%. Meaning, model captures 49 eliminated images correctly out of 100 eliminated images in real. This is higher than the random model, which is created to set a baseline, and to test the performance of other models. This shows how text itself can help reduce number of alternatives at the time of decision making. This seems very useful in the area where tasks related to finding suspects are concerned.

Moreover, retaining few parts of speech can be helpful to increase the speed of mining as they retain a lot of information simultaneously reducing the noise. This did not perform better than the models without

filtering the parts of speech on the auxiliary data, but at least equivalent to those models, that is, approximately 48%. However, we chose the best model (49% precision and recall) out of all the combinations tried on auxiliary data and applied on a left out set of data. The precision and recall score (47%) were close to the results on auxiliary data which is promising and gives a signal to enhance these models with further research for better results.

4 INTRODUCTION

This chapter starts with the broad context of the topic followed by the objective and the problem statement of the thesis.

4.1 CONTEXT

Many people tend to forget names of the people they meet. However, most of the times they do remember the face to an extent. Meaning, face is the most distinctive way of identifying someone. Various kinds of information are derived from a face including visual structural code. In other words, these codes form basic distinguishing features of a face in the viewer's mind with the help of a unique arrangement of individual features (Young, 1986). Hence, when describing a person, people start with outer visual features, which seem to be distinctive.

The study is based on two experiments, "Guess Who task" and "Face description task", where images and text data describing those images, serves as the base of this study. The text data related to these portraits and the other game (Guess Who) is collected via human intelligence tasks available online. As this data is of unknown faces for the participants, the description of these faces includes a very basic perception of participants.

4.2 OBJECTIVE

This study aims at using text mining to find the connection between the questions asked and the images shown in the board game. With the emergence of tagging, the social media is booming with text linked images. Many search engines also base their search based on tags attached to the images. Moreover, content based search for image retrieval has also been proved to enhance the performance of image retrieval, especially in the images where more domain knowledge is required (A. Sagae, 2015). Text data is more descriptive in nature and could explain some details which are otherwise missed in the perception. Moreover, as the text for online images is filled with different description from various users, it becomes a rich source of information than visual stimulus alone. This also gives a perceptive insight into what areas people focus on when they see an image. As the descriptions in our data are also generated by people, they form a good base of understanding what people notice when they see a portrait. As we are focusing on text data of these images, our aim is not only to compare the performance with visual stimuli, but also to compare various models and decision rules used in text mining with each other. Even though there are methods to categorize images with image processing and multi-modality, we want to understand the strength of text which is generated by people and the way people make choices visually. Information related to a subject can be stored in various formats. As these are related to the same object, the information

is pointing in the same direction and has an underlying connection. Multimodal data mining gathers the information from various modes possible, builds a model to decipher the connections between these modes and trains the model to predict a label accordingly. For example, a video, an image, an audio, and a text could be related to same scene or topic and would be complementing each other to complete the information. Meaning, they are acting as a distinct feature for an instance and shall be more powerful to predict the correct label/topic for that instance (Zhen Guo, 2016). On the other hand, our study focuses only on the text data and does not take any information from the image processing. In fact, all the information related to the image, which is provided by the participants in text format is used. Hence, all our features are in the same format and same mode. Rather than using the different mode of data to compare with, we rely on the choices participant made as our ground truth. Moreover, this also gives an overview of how powerful text data is when people describe an image and is it helpful for others to base their decisions on.

Text mining is quite popular to extract information from data which is either structured or unstructured. It has been applied in many areas to categorize texts, summarize texts with a topic, clustering similar kind of documents, etc. (Hussein Hashimi, 2015). On the other hand, facial images are perceived intuitively. If asked to describe a face, many people would give description according to their own point of view. Hence, it is intriguing to compare the way people describe an image and make choices visually. Many text mining research are done to find the similarity between the texts (Wael H. Gomaa, 2013). However, comparing this similarity with visual similarity (ground truth) is unique to this project. This show, if text helps to narrow the gap between the way people think and describe.

Answering a question from a reading comprehension has become an interesting topic for research in the field of text mining. These studies aim at improving models to find answer of a question with the help of reading comprehension capability of the model. It not only focuses on similarity but also the comprehension with respect to context (Pranav Rajpurkar, 2016)

In our study, we also try to find a close match to the question. However, it is not an answer to the question. It could be something related to the context of the question. To predict the similar text related to the question, vectorizer will be used to transform the data and sorted according to their similarity score with the question with the help of cosine similarity. Moreover, a customized decision rule will be used to ascertain the threshold per case. This will be discussed in the 'Method' section in detail. Overall, two types of decision rule are used and compared with each other to understand the impact of decision rules on performance. On a granular level, each image is classified as "to be eliminated" or "not to be eliminated" per instance. This classification is based on combination of two types of similarity representation, and two

types of decision rule. And one model, with an additional layer of pre-processing, to compare the information provided by the most important parts of a sentence.

The main research question whether text data is efficient to capture the visual/perceptive clues will be answered with the help of the performance of above-mentioned combinations. All these possible combinations will be compared to each other on the evaluation metrics such as precision, recall, etc. This would bring more clarity to what kind of combinations captures the text description best and closer to the visual choices made by the participants.

The data is versatile and touches various fields ranging from psychology, cognitive science, text mining, image processing, etc. This study aims at deep diving in the unstructured data of text with the help of text-mining and some interactions with cognitive science. Next section covers related work in text mining and how this work gives a better understanding of various problems in the field of face recognition, images, decision making, etc.

5 RELATED WORK

This section covers various topics our study touches. As the data we are handling, has text data describing faces, it is important to understand how people perceive face, and describe them accordingly. Furthermore, how this perception then helps them ask a question and make choices is also an interesting area to explore. How turning these problems into a text-mining problem could be beneficial to gain maximum information which is processed while making these perceptions/decisions. In the later parts, we discuss work related specifically to text data and how our study is similar or different from them.

5.1 PERCEPTION OF FACE

A lot of complexity is involved while processing a face. However, we can process that quite fast with some dominant features to recognize one face from other. This processing includes various inputs, such as facial features, emotions, expressions, etc. Not only the facial features help us identify one face from other, but it also affects the social perception of the observer. It has been discussed that information like age, emotion, fitness, and familiarity can be obtained with the help of facial features of a person. It has also been mentioned that dynamic or multi-modal information might have a stronger impact on the observer's perception (Montepare, 2008). This suggests that not only physical cues are picked by the observer, but some latent cues are formed while observing a picture.

As the "Guess Who Task" and "Face Description Task" both depend on text description of the images, it is important to understand the complexity involved to extract relevant features which are equally prominent while making the perception or a decision based on the image recognition. As this perception leads to following actions such as recognition, description, etc, the next section shows how decision making is an important part of the game.

5.2 DECISION MAKING

Decision making is a process of choosing something out of the alternatives available at a given moment based on certain criteria. Even though this seems simple, it involves complex cognitive processes underneath. It has always been an interesting topic for multiple fields as it is one of the fundamental cognitive process. Various theories are in place to explain how decision-making works and affects our day to day decisions. One of the theories, discussed by Kahneman (2011) in his book "Thinking Fast and Slow" is that there are two types of systems in place which help the formation of our thoughts. First, which is fast and subconscious, and second, which is slow, logical, conscious and requires effort. For example, first

system would kick in while recognizing a face. On the other hand, when deliberately picking which feature to ask about the images, second system would be more dominant than the first. Hence, it is interesting to decipher the texts based on the combination of these two systems in the process (Kahneman, 2011)

Identifying a face from a set of faces is a process which involves pruning the alternatives by eliminating and dividing optimally. One of the most interesting part in the process of decision making is how people try to optimize the opportunity by asking various questions which seems to categorize the set of alternatives with maximum information gain. With each question or feature, the options are categorized accordingly based on the implicit categorization according to the context (Hendrickson, 2009).

On the other hand, if we try to understand the decision-making algorithms in machine learning, one of the most popular and transparent methods is called decision trees. Decision trees were also used in operational research problems/Economic problems to find the best alternatives amongst a set of possible options. The way decision trees work is interesting to this experiment as this gives an idea how people tend to gain maximum information helps them best split. Ideally, the attributes that maximizes the gain ratio shall be asked first to decrease the number of attempts to reach the correct answer (V. Podgorelec, 2002). A lot of researchers have compared the decision tree and the way humans take decision. There are some similarities. However, there are huge differences, in terms of consistency, speed, and transparency. If one would like to learn from their previous decisions, it is important that these decisions are based on a logical order, and rules that can guide later. (Rolf, 2005). Most of the times, humans base the decision based on their motives. However, they are not statistically sound because system 1 discussed by Kanheman (2011) kicks-in and might hamper the probabilistic model which would otherwise be close to linear models rather than Bayesian network of decision tress, which also considers the conditional probabilities.

5.3 INFORMATION RETRIEVAL

A lot of research has already been done in text mining related to information retrieval. Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers) (Christopher D. Manning, 2008). This technology is widely used by web search engines, email filters, etc. Mostly this is used to process the unstructured data. As most of the data on web consist of text, information retrieval uses the indexing technique to provide some structure to the text queried and text available. Various forms of indexing such as document indexing, word indexing, and concept indexing are used. However, the most commonly used is word indexing (Nadkarnai, 2002). The next steps following the indexing are filtering and searching. All these steps constitute the Information retrieval system. Searching process uses the structure made by first two process and compares the query and possible alternatives to compare with

for most relevant information (Roohparvar, 2015). We also use similar kind of process, the question will act as a queried text in our case, and the text data describing images will act as text available.

This study aims at exploring the effectiveness of Text Similarity models. Especially, for the text describing an image and question asked by the participant trying to guess the image without any access to the text. This will help in understanding how some features are more relevant and end up in being the most important part of a person's identity that they are most likely in the description, and in the question as well. This can be seen with the similarity between question asked by participant playing game "Guess Who" and description provided for the image by various participants playing the game "Face Rating". As most of the text data is made by natural language. It is important to understand which parts of the language are more important than others. Next section shed some light on the work done in this area and how this can be helpful to or project.

5.4 NATURAL LANGUAGE TOOLKIT AND PART OF SPEECH TAGGING

Natural language tool kit is one of the most popular libraries in python which provides functions dedicated to language. One of the tools is Parts of Speech(POS) tagging. POS tagging has been used not only to identify the parts of speech in one language, but it also helps in understanding the multilingual dynamics between various parts of speech (Benjamin Snyder, 2008). Based on the context, parts of speech like adjectives, nouns, noun phrase, etc are more personal to the identity of a person and are filtered to reduce the features. This step becomes a part of pre-processing and can be used with bi-gram or tri-gram models. We use POS tagging in combination with wordnet as one of the tweaks to improve the performance of model.

5.5 TEXT CLASSIFICATION

Even though, cosine similarity ignores the placement of a word in the sentence while calculating the similarity score, it is still widely used for clustering. The distance between two vectors is measured in the form of the angle and hence it reduces the negative effect a Euclidian distance might bring with the length of the vector (Huang, 2008). Various methods are already used to classify or cluster the text with the help of improved cosine similarities such as textual spatial cosine similarity which also takes into the account the position of the word in a document (Crocetti, 2015). However, the limitation stated above does not affect our case as we want similar descriptions and can bear with different length of the similar documents or different positioning of the words in the documents. Moreover, the overall length of the comparing documents is short.

While retrieving the closest match to the query, one of the issues is vocabulary problem. This means that there would be a mismatch in the terms. The terms mentioned in the query might differ from the one indexed in the documents available to match with. For example, someone might ask a question, “Is it a girl?”. However, the image is described with word such as woman or lady. To search most relevant documents efficiently, it is important to solve this problem. We identified this problem and apply natural language processing toolkit to find synonyms of the words in the query. This enhances the vocabulary and reach for the relevance. One of the areas which is discussed by Olalere A. Abass is taking the feedback from the irrelevant documents as well to improve the relevance. This part is not embedded in the models we tried. However, seems a good approach to enhance the performance of the model (Olalere A. Abass, 2017). On the other hand, we tried to enhance the vocabulary with the help of synonyms in one of the models.

Text mining with image data is either processed in a multimodal format or stand alone. Multi modal data mining focuses on combining various type of inputs and the relation between them to enhance the prediction (Zhen Guo, 2016). Another area where image data is combined with the text data is tagging the image according to the context. Supervised learning is used to find the right tags or topics for an image by using a set of image, text, and topics for training. Even though, no data for image processing is used here, the image is used for collecting text data for training (Chee Wee Leong, 2010). Tagging not only for images but for resources is also very popular, and increases the richness of social annotation. Many ongoing research tries to find similar resources with the help of similarity between the tags of the resources on an aggregated semantic level to reduce the noise (Benjamin Markines, 2009). In our study, we are not using image processing, but only the text data for the image. However, the ground truth information in our case is a combination of visual perception and decision made by the participants. Hence, the comparison of the output is multi-modal in nature.

Another area where text-mining is extensively used is answering an open question from the text available. However, the possibility that the text might not have the answer cannot be ruled out. This leads to a pattern where answer triggering models are searched with the help of various sentence semantic methods such as paragraph vector(PV) and convolutional neural networks where PV is a cosine similarity between question vector and the sentence vector (Yi Yang, 2015). For our task we use text mining not to answer the question but to find whether this question’s context matches with the descriptions of the images. We will be also using cosine similarity as our distance metric to put the image in one of the clusters, namely, “to be eliminated” and “not to be eliminated”.

5.6 UNSUPERVISED LEARNING

Clustering and Classification are two ways by which data is mined. Classification problem has categorical output and each instance falls under one of the category. Most classification problems fall under supervised learning where the model to mine the data is trained with the help of the labels (or outputs). For example, naïve bayes classifier, decision tree classifier, support vector machines. All these models consider the features and output to build a model or decision boundary. On the other hand, clustering falls under the category of unsupervised learning which means the model is not supervised by the output in the training set. The instances are classified again in one of the buckets. However, the labels are not used while building the decision boundary and relies solely on the fact that the documents most similar will fall in one cluster with the help of distance-based-clustering algorithms (Allahyari Mehdi, 2017). The concept for unsupervised learning relies on the fact that outputs is not used in one of these buckets to train the model, the model will fall into the category of unsupervised or semi supervised learning. This is important to our project as we will not be using the labels to build our model. It would already know two categories in which it can classify an image. However, it will not learn from the labels to find a decision boundary, but from various approaches of similarity representations, distance measures, and decision rules.

5.7 DISTANCE MEASURES

Distance measures form the base of clusters/classification by measuring distance between data-points not only for text data but for various kind of data (Shirkhorshidi AS, 2015). There are various types of distance measures, such as manhattan distance, euclidean distance, cosine distance, etc. Each of the distance measure have their own pros and cons and it is important to choose the similarity measure according to the case in hand. For example, as mentioned earlier, even though cosine similarity might ignore the positioning of the words in a vector or the length of the vector. However, it does measure the angle between the two vectors and hence is helpful to find the context similarity in our case.

To find the similarity between the texts with the above-mentioned similarity measures, it is important to transform the text into features with the help of vector space model (Cha Yang, 2007). Various methods are used to transform the data into a vector. One is called count vectorizer which is counts the occurrence of a word per document out of number of distinct word in that document and assigns a weight accordingly. On the other hand, TF-IDF is other well-known representation which stands for term frequency and inverse document frequency. Term frequency keeps a count of a term in the document. However, IDF weighs the term frequency according to the presence of a term in various training documents available (Chien-HsingChen, 2017). The main difference here and the former approach of count vectorizer is the

way the terms are weighed according to their frequency. The Inverse document frequency weighs the terms which are less frequent over different documents more than the ones which are more frequent (Gerard Salton, 1988). Meaning, if a term is extremely common over all the documents, it would not be a distinguishing feature to base decision on. TFIDF will take care of such cases. Performance of this model will be then compared with count vectorizer to understand the difference in the performance.

6 METHOD

6.1 DATA SET DESCRIPTION, FEATURE DESCRIPTION AND SOFTWARE

6.1.1 Data set description

The data is collected with the help of two type of experiments namely 'Description Task' and 'Guess Who task'. These experiments were conducted on crowdsourcing Internet marketplace called Amazon Mechanical Turk.

Task one, 'Description Task' involves describing a face shown on the website.

The data consists of two experiments:

i) Face rating experiment or Description Task, and
ii) Guess who experiment.

The Face rating data consists of 196 faces to be rated by 205 raters. Each rater rates five faces. Overall, there are 1138 descriptions for 196 faces.

The second experiment consists of 200 participants. The participant plays a game where 16 faces are shown, and the participant should guess a target face by asking close ended, yes or no question. As there is no target face, the answer to the question is generated randomly. Each participant played four times (also called "board"), and the set of faces in each board differ randomly across the game.

These two types of experiments were conducted twice with the same set of images:

Experiment 1	Description Task	196 faces, 205 raters
Experiment 2	Guess who task	200 participants, 4 board games per participant
Experiment 3	Description Task	196 faces, 500 raters
Experiment 4	Guess Who Task	1000 guess who games

6.1.2 Preprocessing

All the data related to the experiment is stored in a JSON format. Hence, it is extracted according to the task. As our aim is to predict the image, which a participant will eliminate, based on the description of images and question asked by the participant, it is essential to extract this information from both the experiments and merge them accordingly for various purposes.

We created three text files, which will serve as the base tables to extract the data further:

i) Extracting unique ID(s) of the participants playing 'Guess Who'. These unique ID(s) are split into Training(auxiliary) and Test set. For each unique ID in Train set, extracting the board number, question number in that board game, question, and answer. A CSV file with unique ID as the primary key, with four additional columns for board number, question, answer is created. As Unique ID(s) will not be distinct due to this division of question per board, the instances amount to the multiplication of number of questions asked by a participant, and the number of board games played.

ii) Extracting image ID(s) in different stages of a board game. Such as, image ID(s) before a participant asks a question, and image ID(s) eliminated by the participant after the question is answered per instance in each board game. The primary key remains the same as above(i.e. Unique ID(s) of participants playing Guess Who Task). This data acts like a connection between the text data present in Guess Who task and the Face description task. In the data itself, the image Id(s) present before a question is asked and the entire image Id(s) eliminated till then are available. Hence, to find out which ones that are eliminated during this question needs to be extracted and that would be saved in this text file for each instance.

iii) Extracting text description of Images shown in the Guess Who task. As we are mainly interested in the text description of these image ID(s) (extracted in the second text file), the description for each image is extracted and stored together for further analysis. As each image is rated by different raters in the 'Description task', it is essential to collect all the descriptions together per image ID for building a text document per image. With the help of the details extracted in previous step (step ii), the text is merged with images available before the question is asked by the participant, and images eliminated after the current question was answered.

6.1.3 Feature selection and description

I. Image ID(S) and their Description

As discussed by Young (1986), the main distinctive features for identifying a person are physical features specifically a person's face. Hence, all the possible image ID(s) in the Guess Who game are rated in another experiment (face rating experiment) and the description related to these images are extracted and used in this paper to predict the images to be eliminated according to the question asked. Therefore, Image Id(s) and their description are one of the major features. The image ID(s) present while asking the question and before eliminating any of those present are referred as Images Pre-Question. The image ID(s) eliminated after the question is answered are called current eliminated Image ID(s). The pre-question image ID(s) are used to extract the related text for respective images and using the texts per image as one document. The current eliminated image ID(s) are used to evaluate the results. As discussed earlier, there are two types of decision rules used in the project. In one of the decision rules, Peek decision rule, the count of these image id(s) are used as hint for setting the threshold. That is, how many image Id(s) shall be picked according to high similarity scores.

II. Question number and question

Another important feature is the question asked in a board game. As the question asked are the base document and similarity with the question will help in identifying the images which are likely to be eliminated, these features are of utmost importance. For each board game, many questions are asked to reach the target face. Each data point is trained at question asked per board game, and the images present before, and after the question was asked. This helps in linking the right questions, and right images in order.

6.1.4 Outlier detection

Three types of outliers were eliminated:

First, the data points where there are blanks in the question asked. Second, where the question asked is NA as in that instance, final target face has been selected. Third, cases where there is only one image id before a question. In this case, this becomes an outlier for our task because there is only one image id before the question, so it will be predicted correctly always. This might skew the evaluation. Hence, all such cases are removed in Pilot files per se before applying the models and similarity measures.

6.1.5 Software and Code

All the process starting from extraction, cleaning, pre-processing, models, etc are executed using Python 3. All the code with stepwise numbers as prefix are available on github-ythakkar.[1] Various libraries in Python helps in using the models and data formats. Numpy is used to convert format or load txt file into an array. Tableau is used to plot the final results of all the models.

[1] https://github.com/ythakkar/FinalCode

Software/Tools	Purpose
Python3	Overall structuring, storing, and cleaning of data. Various libraries available in python were used to build the model, such as Scipy, sklearn, etc.
Github	This is a version control used to co-ordinate and communicate the code seamlessly.
Tableau	Plotting result stored in a text file generated by various fragments of code

6.2 EXPLORATORY DATA ANALYSIS

In the training set of Guess Who experiment total 1061 uniqueID(s) participated in the game. These unique Id(s) in the training set are derived after randomly splitting all the unique ID(s) as mentioned in detail in the section "Experimental Procedure". Out of which 455 unique ID(s) mentioned their gender as female, 608 specified themselves as male, and 3 mentioned "neither". However, it is evident that these numbers are exceeding the total possible unique_id(s) in the training set. Meaning, there are some of the unique_id(s) which are present in both. After making a set of both the lists, the intersection of these list shall be null if there were no overlap. On the contrary, there were 5 unique_id(s) which were common in these two lists. As these are self-reported by the participants and there was no cross check while the game is in progress, a person can report different details for themselves in different rounds. However, as we are not going to use this field as one of the features for building the model, these cases were not removed. Majority of the participants playing the "Guess Who" were from United States of America. 439 females, 548 males and 3 neither.

6.3 EXPERIMENTAL PROCEDURE

6.3.1 Train and test split

Our aim is to predict which face will be eliminated based on the question asked by the participant playing a "Guess Who" game. This means, the base of our data point lies in the Unique Id of the participants. This would also avoid any data leakage between the training and test set. We split our data set into training and

test set with a ratio of 75% and 25%. This was done by randomly picking up 75% of the Distinct Unique_id(s) after combining the data collected in the two "Guess Who experiments" and writing them in csv file named uid_train. This step was executed with the help of the "Extract_random_ _uniqueid_train_test.ipynb", and contains only unique_id(s) which belong to Training Set.

After splitting the data, unique_id(s) in the Training set are used to extract the features. Even though, Board number and question number are not primary features, they are supportive features to reach a single data point. With the help of board number and question number, question asked is extracted as one text (base). And the text(related) to image id(s) present before the question was posed, serve as other documents which will be compared with the question to see if it matches to the context.

This train data is used as the auxiliary data to see the effectiveness of various document representations and similarity measures. The combination of similarity measure, and decision rule, which performs the best, will be applied to test-data to check the performance of the model for completely unseen data.

6.3.2 Classification

On a higher level, this seems like a binary classification task as the texts which are most similar to the question are either classified as "to be eliminated" or "not to be eliminated" by the participant. However, as the information whether an image is eliminated or not is not used in training the model, this task falls under the category of unsupervised learning.

Texts of image ID(s) are classified as similar or not similar based on the similarity scores. However, to reach this classification the similarity scores are ranked and with the maximum difference in similarity scores of two texts a threshold per case is decided to restrict the number of images to be predicted. This way, the model works on a rule based approach where no absolute similarity score is applied as a threshold for all cases. This is a novel approach, which is referred as "Delta Ranking" in this paper which is explained in detail in the next section, algorithms/models.

To set a benchmark, a chance model is developed which gives random similarity score to each image id for each question. These scores are sorted in descending order. Top n texts(images) are classified as similar based on random threshold per case. The results of this model will be set as a benchmark to compare the performance of more sophisticated models. All the combination of models is discussed in detail in the section Algorithms/ Models.

6.3.3 Algorithms/ Models

All the models used in the experiment are described in detail in this section. First, the random model, and how it is built is discussed. Second, the two decision rules used in this experiment are explained in detail.

Next, the two vector representations are discussed in detail to understand the difference in the way they form a vector from a text. Later, few additional models are discussed which involve small modifications/ filtering in the algorithm. For instance, Reversing the order of selection based on the answer generated for the question asked in the game, or, adding a filter of POS tagging to understand the possible reduction of features/text size based on the part of speech selection. Finally, other areas of the model, such as similarity measure, parameters of the vectors are discussed with a list of possible combination of models used in this paper.

Random Model

First, a random model is used to set a benchmark. All the data created by extraction files was fetched in the file Random_base_model.ipynb. In this model, a random similarity score is assigned to each image Id by creating a seed for each instance based on the size of pre-que images. After assigning the similarity score by random model, the scores are sorted in descending order and a random model then decides the number of images to be predicted. For example, an instance in the nth row of the array has 12 image Id(s) when the question was posed. First a random seed is set based on the number of image id(s) before the question, multiplied by the number of image id(s) before the question for instance n-1. Then a sample of random number between 0 and 1 are generated and makes a similarity score vector for that set of 12 images. This random numbers are now treated as similarity score per image Id according to the respective indices. These are then sorted and image Id(s) with top similarity scores are picked up according to the number of image Id(s) present in current eliminated column.

Decision Rule: Delta Ranking

Delta ranking means, determining the threshold with the help of delta or difference between sorted similarity scores. The threshold lies where the delta is maximum. This is a novel approach applied in this scenario.

For example, the similarity scores for few documents with the question at an instance are as follows:

Similarity Scores	**0.23**	**0.45**	**0.1**	**0.7**	**0.8**	**0.3**
Indices	0	1	2	3	4	5

After sorting the scores in descending order, the array changes into the following:

Similarity Scores	**0.8**	**0.7**	**0.45**	**0.3**	**0.23**	**0.1**
Rank	1	2	3	4	5	6
Original Indices	4	3	1	5	0	2
Delta		0.1	**0.25**	0.15	0.07	0.13

Maximum delta is 0.25, and hence according to the decision rule in place, data points ranked 1st and 2nd will be eliminated. That is, image Id(s) at 3rd and 4th indices (1st and 2nd in ranking in Figure1) will be classified as "to be eliminated" and all others as "not to be eliminated".

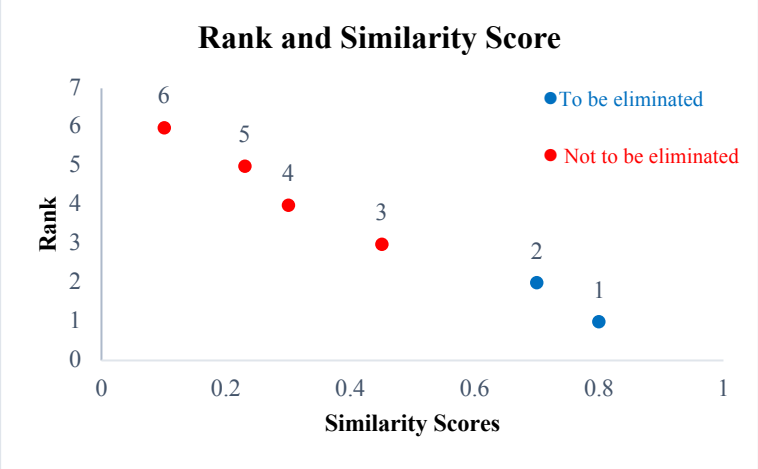

Figure 1: Delta Ranking

With this decision rule Count Vectorizer and TFIDF with and without POS tagging are run to understand the performance of various vector representations. These vector representations with addition layer of POS tagging will be discussed later in this section.

Decision Rule: Peek the Number

This decision rule is to understand the effect of decision rules. The decision rule helps a model to set a threshold per instance. Meaning, delta ranking decision rule also affects the number of images predicted. For example, if there are 11 images before a question is asked and 8 are eliminated after the question is asked. Hence, Peek the Numbers decision rule tries to separate the usage of similarity information for ranking images and for determining the decision point by peeking the decision point from the ground-

truth data. Even though this seems a basic and pragmatic approach. This is a novel approach for this used case. This rule is explained in the form of two cases below:

Case 1: Delta ranking:
If the delta ranking sets a threshold after top 5 similarity scores based on the difference between the consecutive scores, only 5 images will be predicted as "to be eliminated". Out of this 5 if only 3 are correct. The precision score for that instance is 0.6. However, if the model already knows that at least 8 images are eliminated after the question, top 8 images will be predicted as "to be eliminated". This may increase or decrease the precision score.

Case 2: Peek the number
Out of 8 images if 6 images are predicted correctly, precision score increases to 0.75. Meaning, the similarity score assigned to that image might be slightly lower and a wrong image ID might be assigned a higher score in between. On the other hand, it is also possible that the similarity score does not increase or decrease due to more wrong images being pushed up in the ranking. Hence, it is interesting to observe the effect of the decision rule for determining the threshold.

Vector representation: Count Vectorizer
Count Vectorizer is a way presenting the texts in the form of frequency of words in a document. Each image Id before question has a description attached to it. To ascertain the text similarity, the data needs to be transformed into vector space as this represents the text in a form where it is easy to apply algorithms on the text. One of the way in which documents are represented are with the help of count vectorizer. If applied to a document on word level, it stores frequency of word in a document. A word is called a token in this context and assigned a weight according to the frequency it appears in the text (Gerard Salton, 1988). For example, if we have three sentences:

1. This is Water
2. This is an empty bottle
3. Water the plants

Each distinct word in these sentences are collected and each sentence is converted into a vector according to the frequency of all the possible words (i.e 8 words) in that sentence.

	this	**is**	**water**	**An**	**empty**	**bottle**	**the**	**plants**
1	1	1	1	0	0	0	0	0
2	1	1	0	1	1	1	0	0
3	0	0	1	0	0	0	1	1

Vector Representation: Term Frequency-Inverse Document Frequency (TFIDF)

Similar decision rules are tested with TF-idf and results are recorded for both type of vector representations. Parameter tuning such as ngram range and sublinear weighing of the term are used to see the performance for different variants of these vector space representations (Jones, 1972).

In the above example, the terms will be presented with term frequency * inverse document frequency weight.

For the last sentence: Water the plants, there are 3 distinct words. Each word would have a term frequency = 0.33. As these words are also present in other documents, document frequency of each term will be as follows:

$$\text{Water} = 0.3 * \log_{10}(3/2)$$

$$\text{the} = 0.3*\log_{10}(3/2)$$

$$\text{plants} = 0.3*\log_{10}(3/1)$$

Changing the Order of selection based on the Answer

With similar decision rules, and vector representations, a small change in the order of selection will be made based on the answer generated by the game for the question asked by the participant. For example, if a participant asked a question: “Is the person boy?”, and the answer generated for this question is “No”, then the faces with least similarity scores will be selected first rather than the ones with highest similarity score.

Additional Filter: Selected Parts of Speech and Vocabulary Enhancement with Synonyms

Another approach applied to enhance the relevance is using POS Tagging for the question. If the question is converted into a set of filtered synonyms, the possibility to find similar document might increase. In general, Adjectives play a crucial role in the description of a person across various languages (Maass, May 2006). Hence, with the help of NLTK tools, the questions were converted into a part of speech dictionary and adjectives, nouns and noun phrases were retained out of the question for further processing. As these parts of speech theoretically seems to describe an actor the most, synonyms for such words are generated with the help of wordnet. After creating a bag of word per question, these are then compared with the description of the images accordingly. One of the reasons that the descriptions are not exposed to this treatment is that the descriptions are large and already have a lot of words. Moreover, if they are also

changed to the synonyms, this will distort the similarity completely. Hence, only the query is converted to the synonym version with most descriptive parts of speech.

Similarity Measure

Cosine similarity measure is used to assign similarity scores to all the image descriptions with reference to the question asked. After pairwise similarity scores are assigned to each image description, they are sorted and delta ranking decision rule is applied to find the threshold per instance. This decision rule is new in this context and is customized per case. However, as this decision rule also limits the number of images predicted, it might also hamper to see the full performance of the vectorizer or similarity measure. To overcome this problem, the next step was to observe the performance of these models with the same number of predicted images as actually eliminated by the participant. In other words, the data takes a clue for the number of prediction but not which images to predict.

Parameter Tuning

Parameter Tuning refers to various possibilities a model might have to enhance the performance of the model. For instance, Ngrams, sublinear_tf, etc are widely used in the above models to observe the performance of the model. Ngrams refers to a sequence of strings together. However, if the analyzer is set to Word level, ngram refers to a set of sequential words (William B. Cavnar, 1994). For example, wears glasses. Other important parameter specific for TF-idf is sublinear_tf, which weighs the term frequency not on the frequency of the term but by logarithm of the term frequency (Christopher D. Manning, 2008).

List of Models

As there are various combinations of the abovementioned decision rules, vector representation, filtering, etc, a list of various models applied on the Training set are listed below:

Combination	Vector Representation	Similarity Measure	Decision rules	Additional filters
1	Random model	Random	Random	None
2	Count vectorizer	Cosine Similarity	Delta ranking	None
3	Count vectorizer	Cosine Similarity	Delta ranking	Sorting similarity based on answer
4	Count vectorizer	Cosine Similarity	Delta ranking	POS tagging
5	Count vectorizer	Cosine Similarity	Peek the count	None
6	Count vectorizer	Cosine Similarity	Peek the count	Sorting similarity based on answer
7	Count vectorizer	Cosine Similarity	Peek the count	POS tagging
8	TFIDF	Cosine Similarity	Delta ranking	None
9	TFIDF	Cosine Similarity	Delta ranking	Sorting similarity based on answers
10	TFIDF	Cosine Similarity	Delta ranking	POS Tagging
11	TFIDF	Cosine Similarity	Peek the count	None
12	TFIDF	Cosine Similarity	Peek the count	Sorting similarity based on answer
13	TFIDF	Cosine Similarity	Peek the count	POS Tagging

6.4 EVALUATION CRITERIA

Even though this is a classification task at a granular level. However, for each instance, few images are labeled either "eliminated" or "not eliminated". Hence, it deviates from classic classification task and becomes an unsupervised learning. Hence, standard evaluation procedures like k-fold cross validation was not conducted. As we are not using the labels to train the model, application of such an evaluation procedure would not be able to serve the purpose as each instance learns by itself without the help of the label or other instances. Meaning, each instance is a new data.

Overall, we have three evaluation criteria:

i. Precision is used to identify how many images are classified correctly out of the ones which are predicted to be eliminated. Precision is considered for delta ranking decision rule as well as Peek the number decision rule. Precision is the main metric in evaluating the performance for "peek the number" decision rule.

ii. Recall is used to understand if the model can spot as many relevant faces as possible. To test the performance with the delta decision rule, recall becomes more important as it also gives an idea how many more relevant faces are captured out of the possibilities available.

		Actual	
		Eliminated	Not Eliminated
Predicted	Eliminated	True Positive	False Positive
	Not Eliminated	False Negative	True Negative

We are mostly interested in increasing True Positive and decreasing False Positive and False Negative. As we want to focus on Recall, decreasing False Negative would be more important. However, decreasing False negative might increase True Positive or increase False Positive. This trade-off might affect Precision.

iii. F-score is a harmonic mean of Precision and Recall (Steven Bird, 2010). However, the weightage of precision and Recall is modified for this task as we want to focus more on precision than on recall. In its basic form F score gives equal weight to both Precision and Recall. Hence the formula for F-score is twice the sum of precision and recall divided by product of precision and recall (Christopher D. Manning, 2008). However, it is easy to tune this score according to the need of the task at hand. For example, in our case, as we care more about precision, it is important to determine how much do we care more about precision than recall. In this case, we care for preci-

sion twice as much as we care about recall. This also means recall is half as important as precision. The reason for this is to make sure that we focus on increasing True positive as mentioned above. To achieve this, the beta is replaced by 0.5 in the general formula, i.e.

$$F_\beta = \frac{(\beta^2 + 1) * \text{Precision} * \text{Recall}}{(\beta^2 * Precision) + Recall}$$

7 RESULTS

This section discusses the results for various models. First, the result of random model is shared to set a base line. Second, performance of various vector representations, with and without parameter tuning are shared for both the decision rules, i.e. delta ranking, and peek decision rule respectively. Next, the performance of these combinations with additional filter of POS tagging is discussed to compare the impact of this filter on these models.

7.1 RANDOM MODEL

As a baseline, a random model was built to produce similarity scores per image and to cut off at a random. For example, if there are 11 texts and one question. The random model would assign random number between 0 and 1 to each text. These will be considered as a similarity score and as this is a random model, no decision rule will be applied to cut off the number of output. This will be random as well. A random integer between 1 and length of documents available will be generated and similarity scores after sorting will be cut-off till that random integer.

Random Model for Baseline

Simulation 1	Precision	Recall	F0.5 score
Random base model	0.43	0.45	0.40

The focus is building pre-defined classes with the help of unsupervised learning. The two ways of data representation i.e. count vectorizer and TFIDF are chosen to identify the difference between the way words are weighed by these representations for different decision rules.

7.2 DELTA RANKING DECISION RULE WITH VARIOUS VECTOR REPRESENTATIONS

7.2.1 Default vector representations with Parameter Tuning

Next simulation was executed with default TFIDF and CV applied on the questions and descriptions and using Delta Ranking approach to draw the line. It is noted that recall is higher than precision for count vectorizer (CV) and the opposite is true for TFIDF.

To improve on default models, few combinations of parameters were modified. N-gram range helps find the words which are frequently used together. Changing n-grams has no effect on the performance of count vectorizer. On the other hand, it moved the performance of TFIDF minutely. However, that is not much and still the models are not performing better than the random model. We chose not to normalize

the term frequency weights by maximum term frequency in a document as the document in our cases were small description or one-line questions. However, another variation is sublinear term frequency scaling which has been discussed earlier in the method section. This method replaces term frequency with 1 + log (term frequency) and tries to overcome the problem of high weightage to a term due to its high occurrence.

Default TFIDF and CV

Simulation 2	Precision	Recall	F0.5 score
TFIDF	0.41	0.38	0.37
CV	0.42	0.43	0.38
CV ngram_range 2,2	0.42	0.28	0.33
CV ngram_range 2,3	0.42	0.28	0.33
CV ngram range 2,4	0.42	0.28	0.33
CV ngram range 2,5	0.42	0.28	0.33
TFIDF ngram_range 2,2	0.424	0.27	0.331
TFIDF ngram_range 2,3	0.424	0.267	0.329
TFIDF ngram_range 2,4	0.424	0.267	0.33
TFIDF ngram_range 2,5	0.424	0.266	0.329
TFIDF ngram_range 2,2 Sublinear	0.423	0.296	0.339
TFIDF ngram_range 2,3 Sublinear	0.423	0.293	0.338
TFIDF ngram_range 2,4 Sublinear	0.423	0.293	0.338
TFIDF ngram_range 2,5 Sublinear	0.424	0.293	0.339

7.2.2 Changing the sorting order based on Positive or Negative Answer generated per Question

Similarity sorting based on "Yes" or "No" Answer generated by the Game

Simulation 3	Precision	Recall	F0.5 score
TFIDF	0.286	0.28	0.258
CV	0.327	0.332	0.299

As the results were not that promising, we also tried a small modification with the same decision rule where "yes" or "no" answer generated by the game for the question asked by the participant is also taken into the account to sort the similarity scores. In other words, if the answer is no, the similarity measure

will pick least similar texts and respective images for elimination. The option to consider answers performed worse than the model when answers were not in the equation. Hence, we continued to improve the default version of TFIDF and CV without considering answers in the process.

7.2.3 Filtering the text based on POS tagging, and enhancing Vocabulary with Synonyms

One of the options which we wanted to compare was how well this model perform if the features are filtered based on the parts of speech. With the help of various literature, we assumed that most of the information revealing a face description would be either adjective or noun/ noun phrase of the parts of speech. However, to overcome the problem of limited vocabulary for the parts, we also added synonyms of those words to the question, keeping the description of faces as it is.

NLTK Preprocessing with Delta Ranking

Simulation 4	Precision	Recall	F0.5 score
TFIDF	0.418	0.349	0.358
CV	0.421	0.376	0.368
CV ngram_range 2,2	0.425	0.170	0.270
CV ngram_range 2,3	0.425	0.170	0.270
CV ngram range 2,4	0.425	0.171	0.271
TFIDF ngram_range 2,2	0.425	0.171	0.271
TFIDF ngram_range 2,3	0.425	0.171	0.271
TFIDF ngram_range 2,4	0.425	0.171	0.271
TFIDF ngram_range 2,2 Sublinear	0.425	0.171	0.271
TFIDF ngram_range 2,3 Sublinear	0.425	0.171	0.271
TFIDF ngram_range 2,4 Sublinear	0.425	0.171	0.271

One interesting observation here is that the precision for both the vector representations remains the same as in the basic TFIDF and CV in Simulation 2. However, recall drops. One of the reasons for this could be that we are adding synonyms to the questions making the overall vector more varied and consequently adding more noise. This means that this process is not hampering true positive or false positive as much as it is affecting false negative to increase. Parameter tuning helps increase precision, but hurts recall badly. One of the possible reasons for this could be the ngram_range here is completely different because the questions are filled with synonyms. This might be decreasing the reach for relevant documents.

Till now the models are performing lower than the random model and recall is very low. One of the possible cause for such a low performance could be the decision rule we are using to set a threshold for the

number of images to be labeled. Hence, to evaluate the performance of the decision rule we need a reference to compare with. Therefore, we relaxed the unsupervised learning a bit and peeked into the number of images eliminated in that question.

7.3 PEEK THE NUMBER DECISION RULE WITH VARIOUS VECTOR REPRESENTATIONS

This information leakage shall also help bridge the gap between precision and simultaneously increasing recall. The gap between precision and recall will be vanished because sum of true positive and false positive will be now equal to sum true positive and false negative. However, we are interested in increasing true positive by making false positive equal to false negative. This also gives us an idea about the performance of delta ranking decision boundary, whether it was setting too low a threshold or too high.

7.3.1 Default vector representations with Parameter Tuning

Peek the Number Decision Rule

Simulation 5	Precision	Recall	F0.5 score
TFIDF	0.489	0.489	0.489
CV	0.475	0.475	0.475
TFIDF ngram_range 2,4 Sublinear	0.446	0.446	0.446
CV ngram_range 2,4	0.446	0.446	0.446

A hike in performance is observed. If we compare this performance with simulation 2 (delta ranking decision rule), there is an increase in both the metrics of the models. This means that the threshold set after peeking the number tends to include correct images more than including wrong images or excluding right images. It is also interesting that now TFIDF is performing better than CV as opposed to the results in simulation 2. To continue improvement in the performance similar parameter tuning as simulation 2 were done to see similar kind of hike. On the contrary, this parameter tuning decreases the performance of the models. Hence, TFIDF in its default version with peek decision rule seems to work better than the random model.

As we can see parameter tuning loses out the balance between precision and recall. We move forward with peek decision rule and apply it on NLTK processed data.

7.3.2 Filtering the text based on POS tagging, and enhancing Vocabulary with Synonyms

NLTK Processed data with Peek Decision Rule

Simulation 6	Precision	Recall	F0.5 score
TFIDF	0.486	0.486	0.486
CV	0.486	0.486	0.486
TFIDF ngram 2,2	0.429	0.429	0.429
TFIDF ngram 2,3	0.429	0.429	0.429
TFIDF ngram 2,2 Sublinear	0.429	0.429	0.429
TFIDF ngram 2,3 Sublinear	0.429	0.429	0.429
CV ngram 2,2	0.429	0.429	0.429
CV ngram 2,3	0.429	0.429	0.429

As expected, changing the decision rule boosts the performance. Moreover, it is interesting that both the vector representations are doing equally well. However, to see if parameter tuning helps improves performance here, we moved forward with tweaking Ngram range and scaling. Parameter tuning for TFIDF and CV in peek decision rule seems to go in the opposite direction as it decreased the all the metrics compared to its default version in this simulation. Hence, if we choose the best of NLTK processed and non NLTK processed, we can also compare whether filtering affected the performance in a bad way or had no effect. For this reason, we compare simulation 5 and simulation 6 without parameter tuning and observe that this additional layer of pre-processing which reduces the features to a limited type of parts of speech manages to retain most of the information.

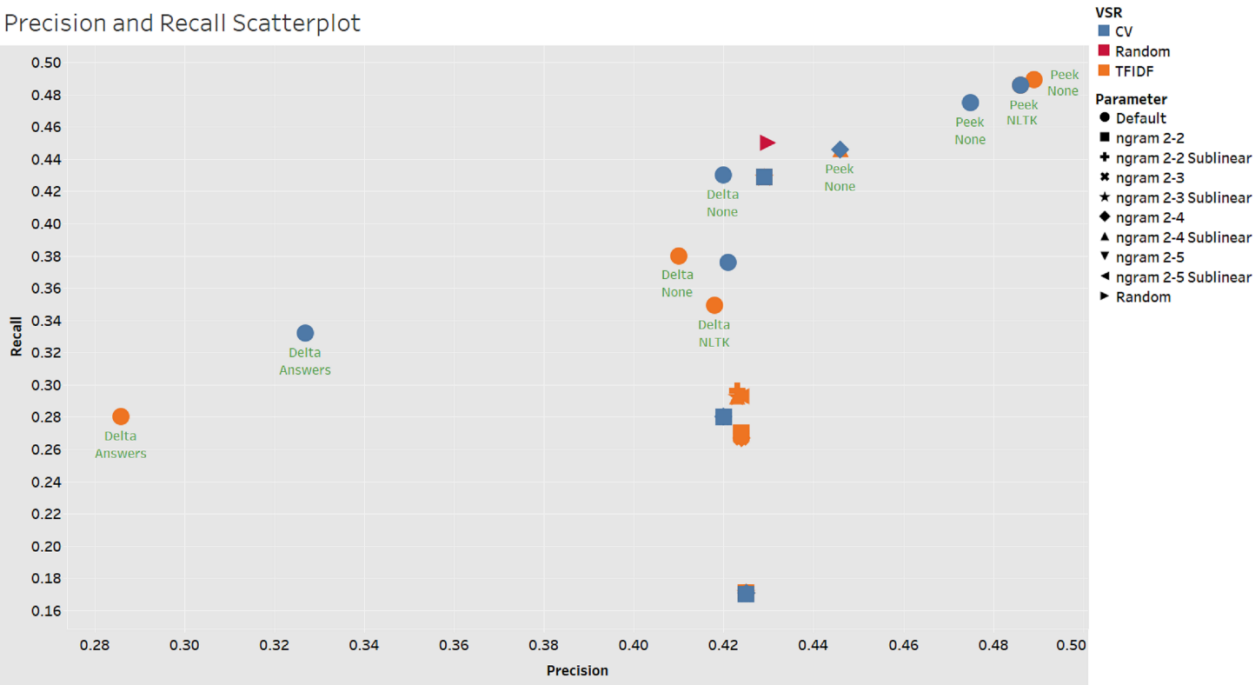

Figure 2: Precision and Recall Scatterplot with all the results (VSR stands for Vector Space Representation) The color represents the Vector Representation Used. The decision rule is embedded as labels in the graph. The shape represents the parameters of the vector representations. The models on the right side of the Random Model(represented b red triangle) are the ones performing above the base line, and the ones on the left side of the Random model are performing below the base line (worse).

Overall, various simulations were tried starting from basic TFIDF and CV model with cosine similarity. Not only various parameters and decision rules were tried, the performance was compared by adding an extra layer of NLTK pre-processing with selected parts of speech and synonyms to overcome the problem of vocabulary of filtered words. This helped us understand that most of the information is stored in few parts of speech such as adjectives, nouns, and cardinal number. All these simulations were conducted on training/auxiliary data. The model (simulation 5), which performed the best on the training set was selected as the final model and then tested on the test set to check the performance of this model on a similar but unseen data (Figure 2). Hence, we applied simulation 5, i.e. TFIDF, default version with peek decision rule on our test set. The model performed at a precision, recall and F0.5 of 0.478 on the test set, which is less than what it performed on the auxiliary data. However, it is still better than the base line set by random model. As there were various simulations with interesting outcomes, the next section will summarize the same with further possibilities in this area.

8 DISCUSSION

As observed in the results section, there are few interesting findings which came up during this study such as the possibility to match up with the visual cues based on text similarity is tough especially when the comparable text is a one-line question. This could be one of the major reasons that the precision did not increase tremendously even after peeking the number of possible members to bucket in each category. One of the other interesting finding were the performance of TFIDF on the whole data and on the NLTK preprocessed data was compared. However, this can be further investigated on granular level to find out whether this performance is due to filters or decision rule. Furthermore, different combinations of NLTK pre-processing could be compared to find the most contributing features.

On vector space model, both the representations performed closely. However, TFIDF performed slightly better in majority of the experiments. To extend this model, further possible areas shall be explored such as combination of TFIDF with supervised learning. However, this task would require additional effort to transform the data on image level and label each image id per question as "eliminated" or "not eliminated". Due to limited time, we could not afford to transform the whole set-up, but find this area worth investigating in future. Another area which seems interesting is combining the answers to change the question from interrogative to affirmative or negative sentence and then compare with the similarity especially after filtering parts of speech.

Furthermore, this data is rich in perspective and further research can also be based on the perspective from other features such as gender or ethnicity and if descriptions about similar ethnicity/gender have some common words, stereotypes or specific facial features which are described the most. Moreover, it is also possible to extend this research based on participant's features leading to similarity in the way similar groups describe.

We only focused on text data in our study. However, further research could also be based on image data or multi modal data trying to combine image and text similarity.

9 CONCLUSION

Based on the results and findings, we conclude that text data is hard to predict which faces will be eliminated based on a question. Especially as these cases are static, it is also possible that text is unable to capture all the information visually received and processed in our brain. However, term frequency and Inverse document frequency performed the best with a precision and recall of 49% which is higher than the random model, which also suggests that in a case of narrowing down a search to a target face in practical setting texts can help in eliminating the alternatives to speed up the process.

Even though, similarity measures in unsupervised learning can perform better than the random model and increase the chance of identifying, they are still less than 50%, and further research can be done to improve upon these models.

Furthermore, we also found out that retaining few parts of speech can be helpful to increase the speed of mining as they retain a lot of information which does not make them perform better than the default model but at least equivalent to those default models. This also suggests that while describing a face or making a decision based on the description of a face, people tend to focus more on the adjectives, nouns, etc. Consequently, in machine learning problems this could be very helpful as it helps to enhance the speed of learning, reducing noise simultaneously. It could also help in understanding, how decision-making is based on weightage of few features more than others, and finding these features can enhance the computation tremendously.

10 REFERENCES

A. Sagae, S. F. (2015). Image Retrieval with Textual Label Similarity Features. *Intelligent Systems in Accounting, Finance and Management*, 101-113.

Allahyari Mehdi, S. P. (2017). A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. *KDD Bigdas.* Halifax, Canada.

Benjamin Markines, C. C. (2009). Evaluating similarity measures for emergent semantics of social tagging. *WWW '09 Proceedings of the 18th international conference on World wide web*, (pp. 641-650). Madrid.

Benjamin Snyder, T. N. (2008). Unsupervised multilingual learning for POS tagging. *EMNLP '08 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (pp. 1041-1050). Honolulu, Hawai.

Cha Yang, J. W. (2007). Text Categorization Based on a Similarity Approach. *In Proceedings of International Conference on Intelligence Systems and Knowledge Engineering (ISKE).*

Chee Wee Leong, R. M. (2010). Text mining for automatic image tagging. *COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, (pp. 647-655). Beijing, China.

Chien-HsingChen. (2017). Improved TFIDF in big news retrieval: An empirical study. *Pattern Recognition Letters*, Vol 93, pp 113-122.

Christopher D. Manning, P. R. (2008). *Introduction to Information Retrieval.* Cambridge University Press.

Crocetti, G. (2015). *Textual Spatial Cosine Similarity.* Proceedings of 12th Annual Research Day, 2014 - Pace University.

Farah, M. J. (1998). What is "special" about face perception? *Psychological Review, 105(3)*, 482-498.

Fernando Perez-Telleza, J. C. (2014). Weblog and Short Text Feature Extraction and Impact on Categorisation. *Journal of Intelligent & Fuzzy Systems, vol.27* , 2529–2544.

Gerard Salton, C. B. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 513-523.

Hakan Altınçay, Z. E. (2014). Ternary encoding based feature extraction for binary text classification. *Applied Intelligence*, 310–326.

Hendrickson, R. L. (2009). Categorical Perception. *Cognitive Science > Wiley Interdisciplinary Reviews: Cognitive Science*.

Huang, A. (2008). Similarity Measures for Text Document Clustering. Department of Computer Science, The University of Waikato, Hamilton, New Zealand.

Hussein Hashimi, A. H. (2015). Selection criteria for Text mining approaches. *Computers in Human behavior*, 729-733.

Jennifer J. Richler, O. S. (2011). Holistic Processing Predicts Face Recognition. *Psychol Sci*, 464-471.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 11-21.

Kahneman, D. (2011). *Thinking, Fast and Slow.* Farrar, Straus and Giroux.

Maass, A. M. (May 2006). Do verbs and adjectives play different roles in different cultures? A cross-linguistic analysis of person representation. *Journal of Personality and Social Psychology*, Vol 90(5), 734-750.

Màrquez, L. P. (2000). A Machine Learning Approach to POS Tagging. *Machine Learning*.

Montepare, L. A. (2008). Social Psychological Face Perception: Why Appearance Matters. *Soc Personal Psychol Compass*, 1497.

Nadkarnai, P. M. (2002). An introduction to information retrieval: applications in genomics. *Pharmacogenomics J.*, 96–102.

Olalere A. Abass, O. F. (2017). Automatic Query Expansion for Information Retrieval: A Survey and Problem Definition. *American Journal of Computer Science and Information Engineering*, Vol.4, 24-30.

O'Reilly, T. (2017). *WTF: What's the Future and Why It's Up to Us.* Random House.

Pranav Rajpurkar, J. Z. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text.

Rolf, C. C. (2005). Beyond Accuracy: How Models of Decision Making Compare to Human Decision Making. Sweden.

Roohparvar, A. R. (2015). Review: Information Retrieval Techniques and Applications. *International Journal of Computer Networks and Communications Security*, Vol.3, No. 9, 373-377.

Ruhe, Y. W. (2007). The Cognitive Process of Decision Making. *Int'l Journal of Cognitive Informatics and Natural Intelligence, 1(2)*, 73-85.

Shirkhorshidi AS, A. S. (2015). A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data. *PLoS ONE10(12): e0144059*.

Steven Bird, E. K. (2010). *Natural Language Processing with Python.* O'Reilly Media.

V. Podgorelec, P. K. (2002). Decision trees: an overview and their use in medicine. *Journal of Medical Systems, Kluwer Academic/Plenum Press*, 445-463.

Wael H. Gomaa, A. A. (2013). A Survey of Text Similarity Approaches. *International Journal of Computer Applications*.

Wei Zhou, N. R. (2006). A tutorial on information retrieval: basic terms and concepts. *Journal of Biomedical Discovery and Collaboration*.

William B. Cavnar, J. M. (1994). N-Gram-Based Text Categorization. *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 161-175.

Yi Yang, W.-t. Y. (2015). WikiQA: A challenge Dataset for Open-Domsin Question Answering. *Proceedings of the 2015 Conference on Emperical Methods in Natural Language Processing* (pp. 2013-2018). Lisbon, Portugal: Association for Computational Linguistics.

Young, V. B. (1986). Understanding face recognition. *British Journal of Psychology*.

Zhen Guo, Z. (. (2016). Multimodal Data Mining in a Multimedia Database Based on Structured Max Margin Learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Volume 10 Issue 3, February 2016, Article No. 23 .

11 APPENDICES

11.1 CODE RELATED TO PRE-PROCESSING

Simulations files can be found at [github-ythakkar]

Following are the file names related to the simulation and pre-processing mentioned above.

Event	File name for Code	Purpose
Pre-processing	01 Extract_random_uniqueid_train_test.ipynb	Splits uniqueID(s) into training and test set randomly
Pre-processing	02 Extract_uniqueid_que_train_full.ipynb 13_Extract_uniqueid_que_test.ipynb	Extracts questions asked by each uniqueID per board on auxiliary set and test set in 02 and 13 respectively.
Pre-processing	03 Extract_face_eliminated_per_question-train.ipynb 14 Extract_face_eliminated_per_question-test.ipynb	Extracts imageId(s) present before and after each question per board game on auxiliary set and test set respectively
Pre-processing	04 Extract face description.ipynb	Collects all the description each image has received in face description task.
Simulation 1	05 Pilot random base model.ipynb	Sets up a random model, which acts as a benchmark
Simulation 2	06 Pilot CV Delta Ranking.ipynb 07 Pilot TFIDF Delta Ranking.ipynb	Calculates similarity between question and image description with the help of Countvectorizer or Term Frequency Inverse Document Frequency (TFIDF) vector representation with Delta Ranking Decision Rule. With different parameters these files generate output according to changes made in the code
Simulation 3	08 Pilot TFIDF Delta Ranking and Answers.ipynb	Calculates similarity between question, answer, and image description with the help of TFIDF/CV vector representation and Delta Ranking Decision Rule and customized sorting based on answers

Simulation 4	11 Pilot POS tag and other NLTK tools Delta Ranking.ipynb	Calculates similarity between question(modified with POS filtering and synonyms) and image description with the help of CV/TFIDF vector representation with **Delta Ranking Decision** Rule. With different parameters these files generate output according to changes made in the code
Simulation 5	09 Pilot CV Peek Decision Rule.ipynb 10 Pilot TFIDF Peek Decision Rule.ipynb	Calculates similarity between question and image description with the help of CV/TFIDF vector representation with **Peek** Decision Rule. With different parameters these files generate output according to changes made in the code
Simulation 6	12 Pilot POS tag and other NLTK tools Peek Decision Rule.ipynb	Calculates similarity between question (modified with POS filtering and synonyms) and image description with the help of CV/TFIDF vector representation with **Peek Decision Rule**. With different parameters these files generate output according to changes made in the code
Final(On test set)	15 Pilot TFIDF Peek Decision Rule on Test Set.ipynb	Best performing Simulation applied on Test set

11.2 SIMULATION RESULTS

Simulation	VSR	Parameter	DR	Other	Precision
1	Random	Random	Random	None	0.43
2	TFIDF	Default	Delta	None	0.41
2	CV	Default	Delta	None	0.42
2	CV	ngram 2-2	Delta	None	0.42
2	CV	ngram 2-3	Delta	None	0.42
2	CV	ngram 2-4	Delta	None	0.42
2	CV	ngram 2-5	Delta	None	0.42
2	TFIDF	ngram 2-2	Delta	None	0.424
2	TFIDF	ngram 2-3	Delta	None	0.424
2	TFIDF	ngram 2-4	Delta	None	0.424
2	TFIDF	ngram 2-5	Delta	None	0.424
2	TFIDF	ngram 2-2 Sublinear	Delta	None	0.423
2	TFIDF	ngram 2-3 Sublinear	Delta	None	0.423
2	TFIDF	ngram 2-4 Sublinear	Delta	None	0.423
2	TFIDF	ngram 2-5 Sublinear	Delta	None	0.424
3	TFIDF	Default	Delta	Answers	0.286
3	CV	Default	Delta	Answers	0.327
4	TFIDF	Default	Delta	NLTK	0.418
4	CV	Default	Delta	NLTK	0.421
4	CV	ngram 2-2	Delta	NLTK	0.425
4	CV	ngram 2-3	Delta	NLTK	0.425
4	CV	ngram 2-4	Delta	NLTK	0.425
4	TFIDF	ngram 2-2	Delta	NLTK	0.425
4	TFIDF	ngram 2-3	Delta	NLTK	0.425
4	TFIDF	ngram 2-4	Delta	NLTK	0.425

4	TFIDF	ngram 2-2 Sublinear	Delta	NLTK	0.425
4	TFIDF	ngram 2-3 Sublinear	Delta	NLTK	0.425
4	TFIDF	ngram 2-4 Sublinear	Delta	NLTK	0.425
5	TFIDF	Default	Peek	None	0.489
5	CV	Default	Peek	None	0.475
5	TFIDF	ngram 2-4 Sublinear	Peek	None	0.446
5	CV	ngram 2-4	Peek	None	0.446
6	TFIDF	Default	Peek	NLTK	0.486
6	CV	Default	Peek	NLTK	0.486
6	TFIDF	ngram 2-2	Peek	NLTK	0.429
6	TFIDF	ngram 2-3	Peek	NLTK	0.429
6	TFIDF	ngram 2-2 Sublinear	Peek	NLTK	0.429
6	TFIDF	ngram 2-3 Sublinear	Peek	NLTK	0.429
6	CV	ngram 2-2	Peek	NLTK	0.429
6	CV	ngram 2-3	Peek	NLTK	0.429
6	TFIDF	ngram 2-3	Peek	NLTK	0.429