Data-Driven Secret Santa

Giorgi Kvernadze*
University of Utah

Abstract

In 2016, the Bank of Georgia and the Georgian Post conducted a nation-wide secret santa for the residents of the country Georgia. For 2017, they wanted to repeat the campaign while using the collected data from the previous year to improve the experience of the participants as well as the overall quality of the campaign. There were three main objectives in the project: 1) Extract information from the user-provided text fields. This was done through a combination of ad hoc linguistic and probabilistic methods. 2) Building a recommendation system that recommends items from local businesses in order to boost local economy. This was accomplished through combining item-item collaborative filtering with the demographic information of the users. And 3) creating a matching algorithm that maximizes the number of participants who both sent and received a gift. We trained a hybrid model with a Random Forest and Logistic Regression. The results were: 1417 extracted keywords, over 450 different local businesses connected with the users and increasing the percentage of people who sent out as well as received a gift by 14% compared to the previous year.

*Affiliation at the time of the project Pulsar AI

1 Introduction

The act of gift giving has been observed throughout the history in many different cultures resulting in anthropologists, sociologists and economists pondering about the underlying reason for such seemingly altruistic acts. Bronislaw Malinowski first famously described the "Kula ring", a ceremonial gift exchange in Papua New Guinea. Both his research and Marcel Mauss' book, The Gift, opened up the debate about the phenomenon of gift exchanges or economies across cultures. Some argue that gift giving can be viewed as an instrument of social or moral obligation [1]. Others claim that it is more than just an expression of obligation. Cumulative efforts to research the purpose of gift giving is often quantified by four potential functions: social exchange, economic exchange, socialization and communication [2].

While the exact drive for gift giving is uncertain, it remains true that gift giving has an important role in the economy. Holidays involving gift exchanges typically increase the retail-sales by a noticeable margin. In the year 2017, the National Retail Federation (NRF) total winter holiday retail sales were reported to be $682 billion, up from $655.8 billion in 2016. The NRF also estimated that the average consumer spends $608 on gifts in 2017 [3]. It is important to note that the psychology of a consumer as a gift giver differs significantly depending on if the consumer is considered as need-based or want-based. Generally, people overspend on holidays and especially when buying gifts for their loved ones.

Rather than analyze the overall cause or economic impact of gift giving, this project focuses on one particular campaign jointly initiated by the Bank of Georgia (BOG) and the Georgian Post, called Secret Santa. Secret santa is a holiday gift exchanging game, usually played between friends, family members or other social groups. Each member of the group is randomly assigned one person to anonymously purchase a gift for, resulting in each person receiving a gift from exactly one person. The Secret Santa created by the BOG differed from traditional secret santas in that participation was country-wide with participants being strangers rather than close social circles. Another popular example of a similar large-scale campaign among strangers is Reddit's online gift exchange with the same name. In 2013, this campaign involved 89,421 people participating from 160 countries [4].

In 2016, the BOG set up a website, secretsanta.ge, where any resident of the country of Georgia could sign up to participate in their game by setting up an account where they provide basic demographic information such as age, gender, location as well as a short message about themselves. The intention of the provided information was so their assigned gift-giver would have enough information about their interests and hobbies to make the decision of a potential gift as easy as possible. Once the deadline for registration closed, participants were matched with each person having exactly one recipient and one secret santa assigned to them.

The 2016 campaign had 58,541 participants in 612 different cities. All of the information that was provided by the participants was retained for the improvement of the next year's campaign. In order to accomplish this, the BOG contracted out the analysis of the collected data to create predictive models that would enhance the overall participant satisfaction. The analysis and the modeling of the data was done by Pulsar AI, an artificial intelligence start-up based in Tbilisi. The digital marketing and campaign management was handled by Redberry. The creative side of the campaign such as TV and outdoor advertisement was done by the advertising company JWT Metro Georgia. The shipment and the handling of the gift items was managed by the Georgian Post.

2 Dataset Description

The dataset spans from 12/02/2016 to 12/16/2016 with a total of 58,541 participants. The variables available for each data point are as follows.

- User_ID
- Age
- Sex
- City_ID
- Interests
- Sending_gift_status
- Receiving_gift_status
- Santa_ID

Demographic Information: Some variables such as Age, Sex and City_ID represent basic demographic information of each data point. Age was converted from the participant's entered date of birth. Sex represented the users' specified gender and the City_ID assigned each participant a specific ID meant to represent the users' provided city of residence.

Interests: The data for the *Interests* was sourced from a text field within the registration page where users were asked to write about themselves in order to assist their gift-giver in purchasing a fitting gift. The exact message read: "Please tell us a little bit about your interests and hobbies. Ex: favorite sport and team, music and band, books, movies, your profession etc." The text box had a limit of 300 characters. An example of a message (translated from Georgian) from a user may be seen below.

> "I listen to all kinds of music but my favorite band is Coldplay, I love books by Remarque, in my free time I love to watch movies, I love to ski, I love colorful socks as well as various accessories".

Result-based variables: Variables such as Sending_gift_status, Receiving_gift_status and Santa_ID were not obtained by user-provided information but instead from the matching process and later analysis of the 2016 campaign. Sending_gift_status and Receiving_gift_status were both binary variables used to represent if the user sent and/or received a gift from the 2016 Secret Santa, respectively.

The variable Santa_ID represents the users' assigned secret santa. A natural question is: what process was used by the BOG to assign participants their secret santas? The matching process was a straightforward chain starting from the first user to the last. In other words, the first user was the secret santa of the second user, the second one was the secret santa for the third and so on. The "first user" was defined as the first person to create an account on the website. The figure below demonstrates the chain used to assign secret santas where the arrow from one user to another represents the assignment of a secret santa. For example, an arrow from User A to User B means that User A is the secret santa of User B .

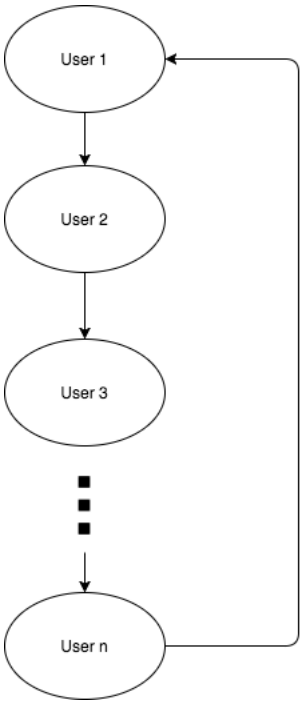

Figure 1: The method used to assign secret santas in 2016

3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial initial step in analysis. It is a way to familiarize oneself to the data through observation of relationships between variables. The intent is to form and then refine hypotheses. As previously mentioned, one of the main goals is to build accurate predictive models using the data that is present to us.

The first step is to calculate summary statistics for provided variables through both univariate and multivariate analysis and visualizations. Sending_gift_status essentially represents the outcome variable so a large portion of analysis revolves around how both individual variables and their relationships affect this label. It was found that only 48% of participants sent out a gift in 2016. Future success of the BOG's Secret Santa is largely dependent on properly understanding what factors correlate to participants actually sending the gift.

Initial examination of the descriptive statistics of the variables immediately revealed a mistake in the Age data. Querying the maximum Age in the dataset showed an impossible value and upon further analysis, we observed that 0.73% of the people reported either impossible or unrealistic ages which skewed the distribution. Since the subset size is not that significant, we will just discard these points for now. The distribution of Ages without the outliers are seen in Figure 2 below.

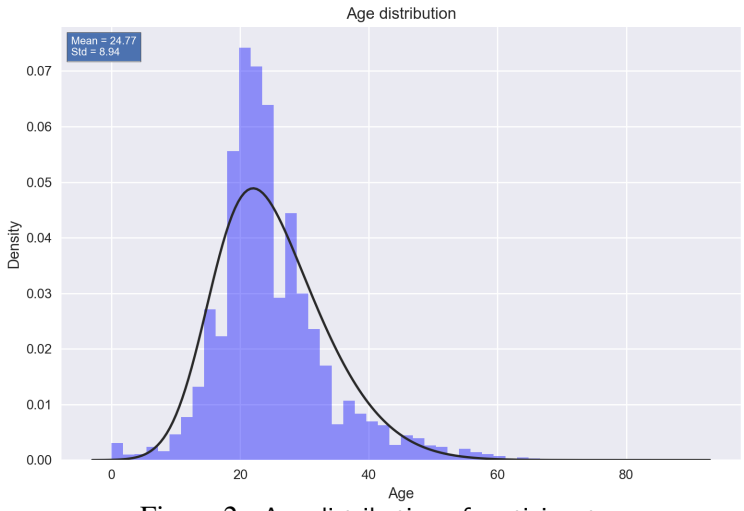

Figure 2: Age distribution of participants

With a mean age of 24.77 years, the distribution shows that the population is mostly younger people. This makes logical sense since web-based campaigns are more likely to appeal to younger populations. The Ages were binned in 8 distinct groups. A contingency table was computed against the variable Sending_gift_status. The visualization of the relationship is shown in Figure 3 below.

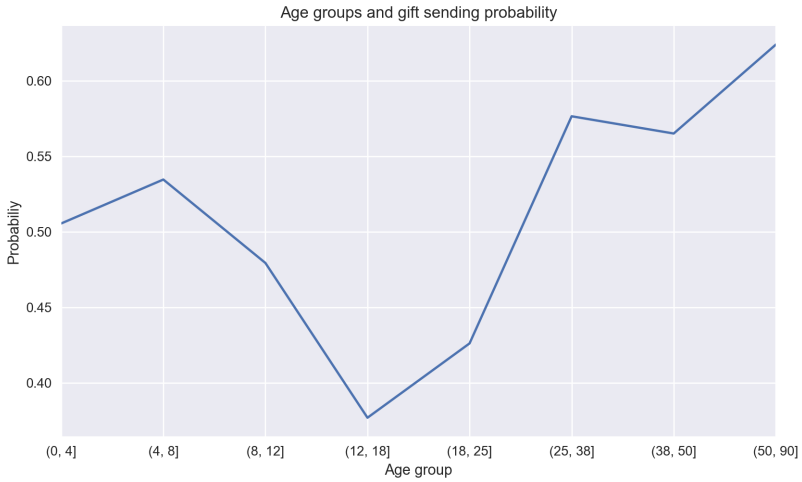

Figure 3: Correlation between Age and probability of sending gift

The graph above reveals an interesting pattern. There is generally a positive correlation between Age and the probability of sending a gift. However, there is slight noise in the data. This can be seen in the ages 8 and below. This can be explained by the fact that these accounts were most likely registered by parents on behalf of their children. Therefore, the probability would likely be based on the parent's age rather than the age provided in the dataset.

With Age considered, Sex of the participants is then analyzed. Comparing data from both Age and Sex, it was found that the Age distributions for males and females are almost identical as seen in Figure 4 below. However, there is a immense difference between the sizes of these populations with approximately 73% of the people in the dataset being female as shown in Figure 5.

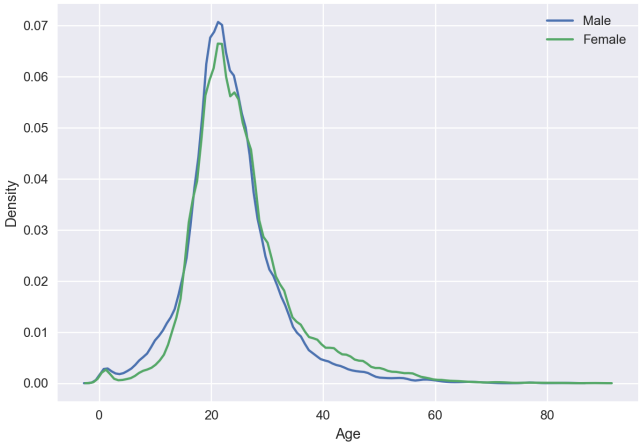

Figure 4: Density plots for Age distributions by Sex

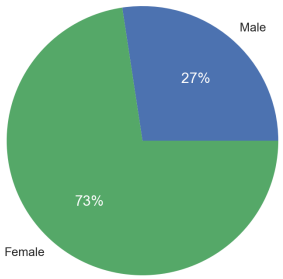

Figure 5: Distribution of overall participants by Sex

Initial analysis revealed that Sex may be one factor affecting the likelihood of participants sending a gift. The data showed that 52% of the females sent the gift, compared to 35% of the males. This can be seen in Figure 6 below.

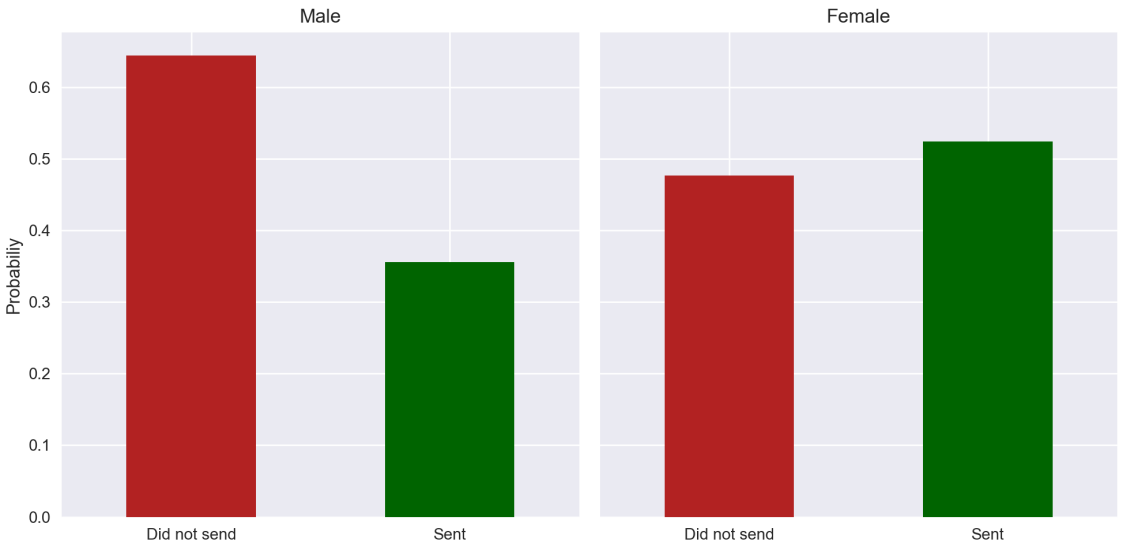

Figure 6: Gender and the difference of probabilities of sending gifts

Unarguably, the most important variable to analyze in our dataset is *Interests*, since it is the most varied and distinctive out of all fields. The variable is essentially a unique descriptor of each person. Figure 7 shows the distribution of the lengths of the provided text in terms of their characters, where Figure 8 shows the correlation between the length of the string variable *Interests* and the probability of the user sending out a gift.

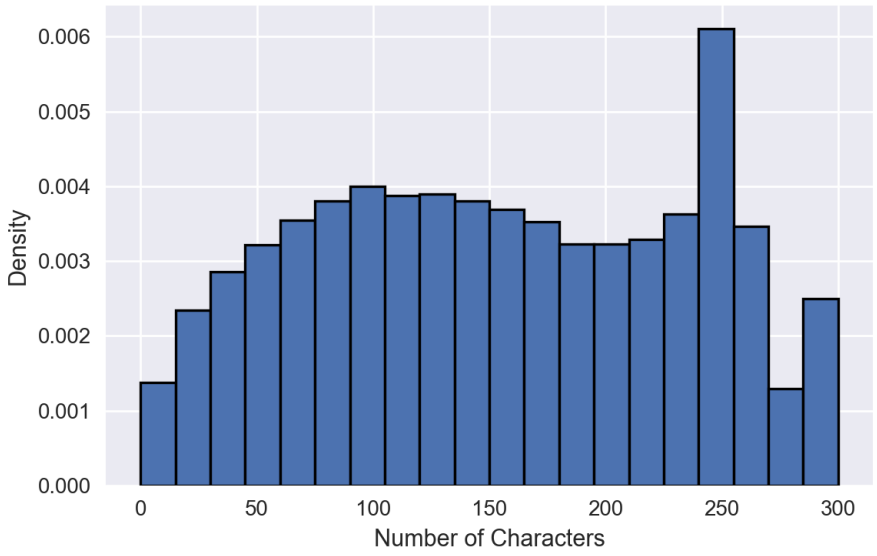

Figure 7: String length distribution

Figure 8: Correlation between character count of Interests and probability of sending gifts

Figure 8 shows a relatively strong positive correlation between the likelihood of sending a gift and the length of the provided text. The more a person writes in their *Interests*, the more likely they are to send the gift. This could logically be explained by a simple observation that a user who writes more is investing more time, which could indicate that a person is more eager to participate fully in the project.

4 Preprocessing Data

A simple procedure is followed for preprocessing the field *Interests*. We remove redundant and extraneous symbols such as special characters or multiple spaces and tabs. Additionally, we coalesce repeating characters, for example: "Goooooooood" becomes "Good", while ":))))))" becomes ":))" and so forth. The idea is to reduce a sequence of identical characters to at most size 2. This is done in order to normalize the words and to make searching easier. Furthermore, we removed URLs and other types of strings that are deemed to be unusable in terms of semantic understanding of the text.

When exploring the variable *Interests*, we noticed that there were three main categories of texts as seen below.

- Proper Georgian: The message is written in Georgian alphabet
- Transliterated Georgian: The message is written with the English alphabet but the content is Georgian
- Proper English

Proper Georgian is easily identifiable; however, distinguishing between transliterated Georgian and English isn't trivial since both of them use the same alphabet and some words in the transliterated Georgian texts might actually be proper English. In order to understand the potential extent of the problem, we first have to measure what the size of these subsets are.

4.1 Transliterated Georgian Classifier

The task is to create a classifier that will classify a message as either proper English or transliterated Georgian. We scraped roughly 12,000 English and 12,000 transliterated Georgian comments from Facebook. We remove all non-alphabetic characters from the comments as well as redundant spaces and tabs. Every word was transformed to its lowercase version. 80% of the scraped comments will be used for building the model, 20% for evaluation. The hypothesis is that each language has a probability distribution of character sequences. In other words, some character sequences have a higher likelihood to appear in Georgian than in English. Our goal is to basically learn this distribution and create a scoring function that will determine the class. For each language, we create a character trigram frequency dictionary. Every character trigram that occurs in the data has a frequency count associated with it. We stored these dictionaries separately for each language. In order to predict the class of the message, we first break the written message up into character trigrams and search what values these trigrams have in each language's trigram frequency dictionary. The score is determined by the sum of the log frequencies. The log function is applied as a smoothing technique. The classification is then based on the comparison of scores from each language. The process is as follows: let G and E be the trigram frequency dictionaries of Georgian and English, respectively. Given a message M and its character trigrams, T, we do the following:

Algorithm 4.1 Classify text as English or transliterated Georgian

Require: G and E, trigram frequency dictionaries of Georgian and English
  $T \leftarrow$ *list of trigrams of the text to be classified*
  for each trigram in T **do**
    $Gscore \mathrel{+}= \log(G[trigram])$
    $Escore \mathrel{+}= \log(E[trigram])$
  end for
  $Total = Gscore + Escore$
  return $\max(Gscore/Total, Escore/Total)$

Table 1: Evaluation of the transliteration classifier

	Precision	Recall	F1-score	Accuracy
EN	0.99	0.98	0.98	0.99
GE	0.95	0.98	0.96	0.97
Avg	0.98	0.98	0.98	0.98

4.1.1 Evaluation of the classifer

The evaluation of the classifier is provided in the table 1. Judging from the results we can use this classifier on the data with great confidence.

Running the model against the data, we found that 16% of the users wrote their texts in transliterated Georgian. In order to transform these texts to proper Georgian, we used a transliteration tool developed by Pulsar AI. The tool applies a mix of rule-based and machine learning techniques to predict the likely mapping of words from one alphabet to the other.

5 Extracting Keywords

One of the most important tasks in the project was to understand, extract and aggregate what people were writing in the *Interests*. This proved to be quite challenging, for the simple reason that natural-language processing tools for the Georgian language are close to non-existent. For that reason, we had to resort to ad hoc and heuristic methods.

Another factor that added to the challenge was the fact that we were dealing with very short and mostly unstructured documents. These conditions hinder the performance of conventional methods for identifying important keywords in sets of documents [5]. For example, the widely used measure, TF-IDF, would be suboptimal since it would give low weights to the words that appear across a great deal of documents. Considering the nature of the data, it is expected to have the same keywords come up in the bulk of the documents. Furthermore, since the text is very short (at most 300 characters long), the term frequency will be very low for a given keyword, simply because most users will mention a keyword once.

Defining Keywords: Any word that identifies a users' interest, hobby or desire for a certain object is a keyword for our purposes. There are 4 main types of keywords. Below is the list of the categories with some concrete examples of each.

- Interest: Music, writing, hiking, cooking, video games, photography etc.
- Proper noun: Pink Floyd, Harry Potter, Golden State Warriors, Agatha Christie etc.
- Object: T-shirt, bracelet, ball, guitar, diary, whiskey etc.
- Attribute: Red, exciting, soft, psychedelic, shiny, decorative etc.

Example from the dataset:

> "I listen to all kinds of ***music*** but my favorite band is ***Coldplay***, I love ***books*** by ***Remarque***, in my free time I love to watch ***movies***, I love to ***ski***, I love ***colorful socks*** as well as various ***accessories***".

5.1 Proper Noun Extraction

The process of identifying proper noun keywords was different from the rest of the categories, since it is more likely that a user will write these keywords in English. The process was initiated by identifying all the messages which included both the English and Georgian alphabet characters, but where the message was dominated by the Georgian alphabet. This was done through character analysis of the texts. We used a series of regular expressions to extract possible keywords from these set of messages.

Before any processing, the total number of keywords found were 9540, but these included duplicates as well as words that can be considered as stop words. We ranked them in terms of their frequencies in order to distinguish relevant and irrelevant keywords. Some examples of irrelevant keywords are: ”secret santa”, ”happy new year”, ”dear santa” etc.

Subsequently, we managed to cut down the list to 3569 keywords. However, this list still contained duplicates caused by people misspelling. For example, cold play, coldpay, coldply, and goldplay all refer to the same intended keyword - Coldplay. We had to find a way to group these keywords to fully capture their true frequencies. Furthermore, identifying these groups would make it easier to extract misspelled keywords in future/unseen cases for the next campaign.

In order to solve this problem, we ran a hierarchical agglomerative clustering (HAC), which is a bottom up approach to clustering. Each point (or keyword in our case) starts out with its own cluster, gradually these clusters are merged until there is only one cluster remaining. Since the goal is to identify $k > 1$ clusters in n points where $k < n$, a visualization method follows this process to determine the optimal cut-off point in the cluster merging process.

HACs are usually visualized with a dendrogram, where every merge of a cluster is shown as a horizontal line. The y-axis shows the distance between the clusters. If two clusters are merged and the vertical line is relatively high, this means that the merging is most probably fruitless. However, it should be noted that what defines ’high’ is subjective and should be determined by the practitioner. There are two main parameters in HAC:

The distance metric: a function that measures distance between two individual points. Usually the values are constrained between 0 and 1.

Linkage: a function to measure distances between two clusters (sets of points).

Since we are dealing with strings, we used the Levenshtein edit distance, which is a popular distance metric for strings. It measures the number of insertions, deletions or substitutions that are needed to transform one string to another. One of the most common application for this measure is spell checking. The linkage used was the average distance between all pairs of elements in each cluster, which is defined as follows:

$$\frac{1}{|A||B|} \sum_{a} \sum_{b} d(a, b) \text{ where A and B are clusters} \tag{5.1}$$

A cut-off distance of 0.4 was chosen through empirical results. The graph below shows the dendrogram visualization of our keyword clusters. The results were a total of 1379 unique proper noun keywords.

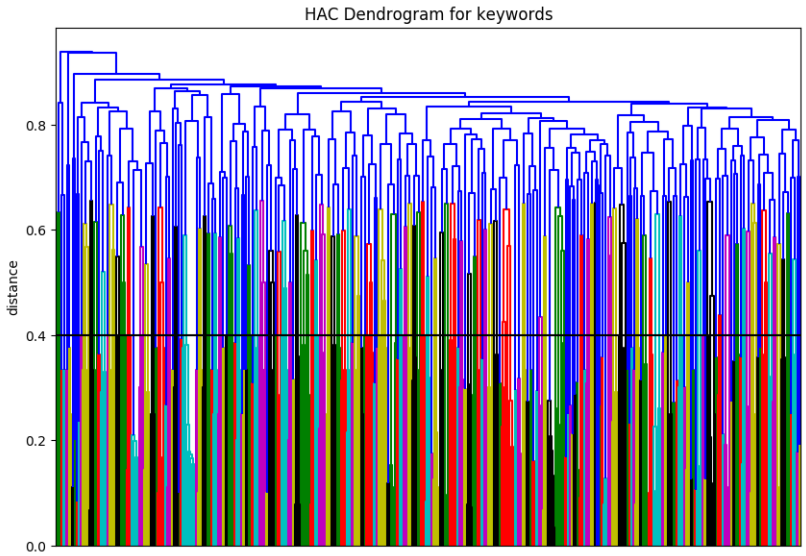

Figure 9: HAC Dendrogram for keywords

5.1.1 Categorization of Proper Nouns

We wanted to assign categories to the extracted proper nouns to better interpret their presence in the data. We define 4 categories: music, movies, books and sports. For each category, we identify keywords that are associated with it. For example, the category of movies has keywords such as ”watching”, ”film”, ”movie”, ”drama” and ”thriller” associated with it. To select the categories, we take each extracted proper noun and determine the co-occurrence counts with category keywords and the proximity. The category with the highest count and lowest proximity is assigned to the proper noun.

5.2 Identifiers for Keyword Extraction

Our approach was to identify common phrases or words that would likely precede or follow keywords. These phrases and words were used as an anchor for finding and extracting the keywords. They shall be defined as identifiers. If a sentence has reasonably proper grammar, there are a finite number of ways to express a desire for an object, or to state an interest in a specific subject. Below we provide a small sample of identifiers that were used (translated from Georgian):

- I like
- I’m interested in
- I want
- I’m crazy about
- my hobby is
- I love

There were total of 29 such identifiers which cover 84% of the data, meaning that 84% of the users mention these phrases at least once in their provided text. We used these identifiers to collect neighboring words. These words were then put in a dictionary and sorted by frequency. All stop words and irrelevant words were removed using rule-based and lexicon methods. It should be noted that this method requires a non-trivial amount of manual work, however it is fairly effective. When there are little to no options for automation, sometimes these ad hoc type processes are the most valid thing to apply.

Another approach we tried was searching collocations that contain one of the identifiers. Collocations are defined as expressions of multiple words that commonly co-occur. Some examples of this are New York,

United States, real estate, etc. For our problem, we can treat an identifier + keyword as a collocation since it is expected to have many keywords expressed in the data with the same identifier/keyword combination. PMI measure was applied to extract the identifier + keyword collocations from the text.

$$pmi(x;y) \equiv \log \frac{p(x,y)}{p(x)p(y)} \tag{5.2}$$

The combined results of all of these methods were a total of 2019 keywords. We dropped a significant amount of them for having too low frequencies. We also grouped keywords that had the same semantic meaning. Finally, we ended up with 1417 total keywords of which 826 are unique.

5.3 Generalized Model for Keyword Extraction

Now that we have a reasonable amount of keywords at our disposal, we can use them to build a model to automate the process for keyword extraction. In order to accomplish this, we create word-embeddings for the words that we are interested in by observing the contexts of the words. We define a context window of size 3 and proceed to collect all the contexts from the dataset that contain one of our keywords, these will be our positive examples. We also collect all the contexts that do not contain a keyword, these will be our negative examples. Since there are much more non-keywords in the data than keywords, there is going to be a significant class imbalance. Therefore, we apply random sub-sampling to balance out the classes to improve the results of our classifier.

The collected contexts are transformed to vector representations by the bag-of-words model. Each context is represented by an N dimensional vector, where N is the number of unique keywords in the entirety of the contexts.

At this point we have successfully reduced keyword extraction to a binary classification problem. Depending on a context a word is either a keyword or not. For training and testing, we combine our list of keywords with random words from the data that are not keywords. We then split this combined list as 80% training and 20% testing. The context collection with transformations and model fitting is done on the training data. A Support Vector Machine (SVM) is trained on this generated dataset.

We used two methods for evaluation of the generated classifiers. Predicting if a word is a keyword based on just a single context and predicting using a batch of contexts. The reason behind this is that, in application, a more realistic scenario is that we aggregate a set of contexts for a word and then we classify it as either a keyword or a non-keyword. However, it is still useful to know how well would our model perform in real-time conditions where the data is coming in as a stream and there is no option to store and evaluate/predict later in time.

Table 2: Evaluation of automatic keyword extractor

	Precision	Recall	F1-score	Accuracy
Single context	0.704	0.705	0.704	0.704
Batch context	0.914	0.806	0.86	0.87

6 Recommendation System

One of the goals of the project was to create a system that would recommend possible gift items for a given user. The idea was to recommend items from local companies in order to boost the regional economy and also to make a more convenient and comfortable experience for the users. Giving a meaningful and relevant gift is difficult when the receiver is one's family member, let alone a complete stranger.

A separate business platform was created for the local vendors. If they wanted to participate in the campaign they had to create an account and provide information about their business. The information was mainly concentrated about the products they have available for users. Furthermore, they were given the keywords that we have extracted from the dataset. Their task was to match their businesses with relevant keywords.

The objective of our recommendation system was to extract keywords from an unseen users' text and then predict possible keywords that are similar to the ones provided. Since the whole process was oriented on the keywords, our approach was to do item-item collaborative filtering, where items are the keywords. Our ratings for items are binary in that either a keyword appears in the text or not.

A data matrix D was created with dimensions $M \times N$, M being the number of users and N being the number of unique keywords that were extracted from the data. Each data point is a vector of binary inputs. The representation of the matrix is provided below:

$$\begin{array}{lc} \text{Matrix D} & \begin{array}{cccc} keyword_1 & keyword_2 & \dots & keyword_N \end{array} \\ \begin{array}{l} user_1 \\ user_2 \\ \vdots \\ user_M \end{array} & \begin{pmatrix} I_{11} & I_{12} & \dots & I_{1N} \\ I_{21} & I_{22} & \dots & I_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ I_{M1} & I_{M2} & \dots & I_{MN} \end{pmatrix} \end{array}$$

Where I is an indicator function defined as:

$$I_{u,k} = \begin{cases} 1 & \text{if the user } u \text{ contains the keyword } k \text{ in their text} \\ 0 & \text{otherwise} \end{cases}$$

Each column of this matrix is essentially a vector representation of a keyword. We use these vectors to construct a gram matrix with the cosine similarity.

$$\begin{array}{lc} \text{Matrix S} & \begin{array}{cccc} keyword_1 & keyword_2 & \dots & keyword_N \end{array} \\ \begin{array}{l} keyword_1 \\ keyword_2 \\ \vdots \\ keyword_N \end{array} & \begin{pmatrix} sim(k_1,k_1) & sim(k_1,k_2) & \dots & sim(k_1,k_N) \\ sim(k_2,k_1) & sim(k_2,k_2) & \dots & sim(k_2,k_N) \\ \vdots & \vdots & \ddots & \vdots \\ sim(k_N,k_1) & sim(k_N,k_2) & \dots & sim(k_N,k_N) \end{pmatrix} \end{array}$$

$$sim(a,b) = \cos(\vec{a},\vec{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{6.1}$$

Once we have a our similarity matrix, we can determine the predictive score of a user-keyword pair using a weighted sum of the similarities. Given the set indices of keywords that were mentioned in the target user's text U_k the score $s(U_k, k^{'})$ is determined by the following function:

$$s(U_k, k^{'}) = \frac{\sum_{i \in U_k} S_{i,k'}}{\sum_{j \in N} S_{j,k'}} \tag{6.2}$$

where $k^{'}$ is the index of the target keyword and S is the similarity matrix.

6.1 Using Demographic Information for Recommendations

As mentioned in the dataset description, we have the basic demographic information about the users. Using this information will most likely improve the accuracy of the recommendation system. It will help with

disambiguation of certain keywords; for example, two users mentioned the keyword 'car', one of them is 10 years old and the other is 40. It is valid to assume that the younger user probably would like a toy car as their gift, but this is unlikely to be true for the older one. Using demographic information will help weed out irrelevant suggestions. We define the following demographic age groups for both males and females: (0, 12]; (12, 18]; (18, 25]; (25, 38]; (38, 50] and (50, 90].

We collect the frequencies of the keywords for each demographic group, and apply the log function to them to smooth out the values. Each group member will have a vector of frequencies of size N, where N is the amount of unique keywords extracted from the data. We use this vector to adjust the weights on our predicted keywords. Given an active user, we first generate a sorted list of all the recommended keywords where the weights define our confidence of recommendation. The top item is the most recommended keyword. We then identify the demographic group of the user and locate the respective keyword-frequency vector. An element-wise product is done on these vectors, which results in a list of keyword-weight association. We sort this list and return the top n of them as our recommendations.

6.2 Evaluation of Recommendation System

We used two baseline models to compare our recommendations system against.

Random: For each instance of a recommendation, we sample n keywords from the entire set of unique keywords with uniform probability.

Top frequency: We pick top n keywords in terms of their frequency in the entire dataset.

For evaluation, we split the dataset into 80/20. We constructed recommendation system on 80% of the data and proceeded to test the results on the remaining 20%. For the points in the test portion we randomly hide one mentioned keyword in the active user then try to predict and see if the hidden keyword is among the top n recommended keywords. Below is a comparison of all of the model performances. It is clear that a combination of item-item collaborative filtering model with cosine similarity and the demographic information reweighing is the highest performing system.

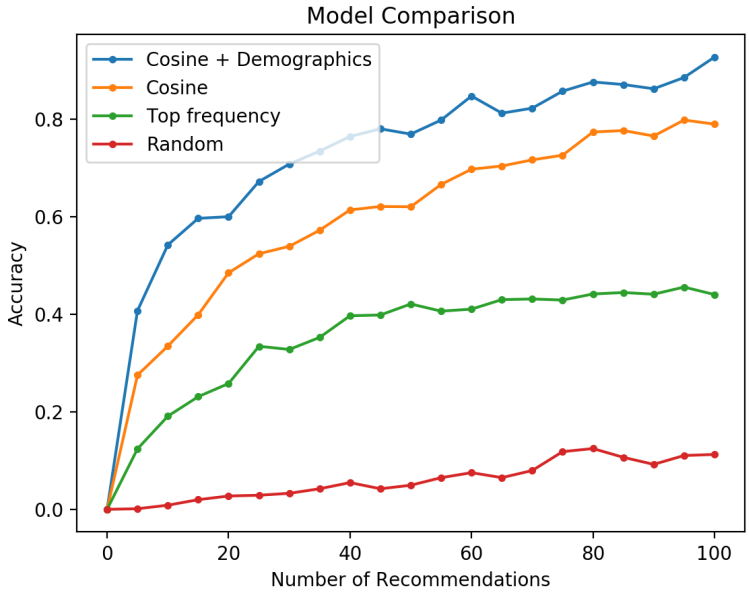

Figure 10: Comparison of recommender systems

7 Matching Algorithm

One of the main objectives of the project was to come up with a model which optimally matches users. Matching refers to the assignment of secret santa - giftee relationship between the participants. Our goal was to maximize the amount of people who both sent and received their gift. A large constraint on the matching system was that there was a time limit on matching. The participants would be coming in as a stream of batches and for every batch we had to perform matching.

7.1 Noise

It came to our attention that there was some contamination in the dataset. In 2016, participants had the ability to go back and change their provided text later in time. Unfortunately, we did not have a record of who changed their inputs. It is extremely important to identify these examples and remove them. We want to avoid our classifiers picking up on these edited texts as important features. Since, the users who did change their texts were overwhelmingly the ones who sent out the gift and did not receive a gift back in return. The changes in their texts expressed disappointment about not receiving the gift. Phrases like ”where are you”, ”did you forget about me”, ”grinch” and symbols like ”:(” , ”;(”, were widely associated with this population. We managed to identify key phrases that express disappointment and show evidence that the text was edited. We proceeded to remove all of these data points from consideration, which was approximately 7% of the entire data set.

7.2 Predicting the Probability of Sending Gift

Our approach was to create a model that would predict the probability of any given user sending the gift, regardless of their assigned giftee. We considered both the textual features extracted from the users' provided text, as well as the demographic information. We define two 'views' of the dataset: the first is purely textual data, taken from the variable *Interests* provided by the user. The second one is constructed with hand-engineered features, full list of the used features is provided below.

Age and Gender: In EDA we recognized strong correlations between gender, age and the probability of sending a gift.

Length: The length of the users' text is essentially a measure of how much effort they put into their account creation. Intuitively, if a person invests more time into it, they would be more likely to honor their promise to send a gift to their assigned person.

Address Length: Upon observing the data, we noticed that there was a significant variation in the character lengths of addresses. There seemed to be a correlation between character length and probability of sending the gift. For example, 14% of the people in the dataset provided an address with length less than 15 characters and 64% of them did not send the gift.

Punctuation: The number of punctuation characters used captures how structured the text is, as well as how much effort was put into writing it. 15% of the users in the dataset did not use a single punctuation character, 73% of them also did not send a gift.

Registration time: This variable indicates how planned was the decision to participate in the campaign. The question is: did the person create an account impulsively, just to see what the website was, or did they actually intend to participate?

Festivity: We defined a lexicon of words and phrases that express festivity. Some examples include ”dear santa”, ”happy”, ”merry”, ”wish you” etc. We then measured both the ratio of these words in the whole text as well as frequencies.This is an attempt to roughly approximate how excited or enthusiastic the person is about participating in the game.

Number of extracted keywords: This feature describes how closely did the participant follow the campaigns guidelines about describing their interests. Which again shows the amount of effort and thought behind the participant's account creation.

Transliterated: A boolean feature representing if the user wrote their text in transliterated Georgian or proper Georgian.

We proceed to train different models on each view of the dataset. For the textual data we first vectorize the inputs with TF-IDF measurement. As we would expect, we end up with a very sparse and high dimensional data. In order to reduce the dimensionality we run Singular Value Decomposition on the dataset. We achieve maximum accuracy of 66%. For the hand-engineered feature dataset we fit a Random Forest (RF). Perform cross-validation for parameter tuning, the maximum accuracy is 68%.

By observing the disagreements between the two models on each dataset, we saw that there was potential to combine these models into one and as a result increase the overall accuracy. The method for combination is as follows, we split the entire dataset into 90/10. The 10% is untouched until the individual models and the combined model is tuned, this will be our final evaluation set. We further split the training part into 70/30, we train a linear model (Logistic Regression) on the 70%, then proceed to predict the probabilities of the labels for the rest of the 30%. This becomes a new meta-feature for the RF. We train the RF on this modified dataset and finally evaluate on the held out test set. Every step involved cross validation for parameter tuning and generalization error estimation. The result was a 2.5% increase in the accuracy. The results for all of the models are seen in the table below.

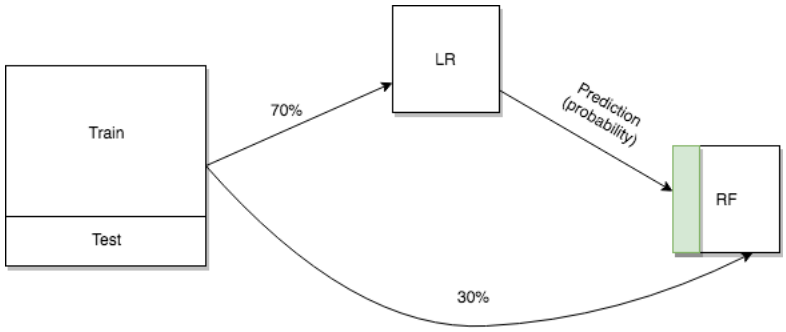

Figure 11: Training process

Table 3: Models for predicting sending gift probability

	Precision	Recall	F1-score	Accuracy
RF	0.6843	0.6827	0.6829	0.6827
LR	0.6654	0.6644	0.6646	0.6644
RF+LR	0.7009	0.7001	0.7002	0.7001

7.3 Matching Process

As mentioned before, the user data was available in batches. We predict the probability of sending the gift for each user in the given batch using the model that was generated. Then, we sort the users in descending order by their probabilities and assign each one from top to bottom. The user with the highest probability

will be the secret santa of the user with the second highest probability and so on. One potential issue to be noted is that the first user in each list will not have a secret santa and the last one will not have a giftee assigned to them. In order to connect these two users, we use the next batch. The next batch will be reversed in descending order but the matching will still be top down. However, now the last user of this batch will be assigned as the secret santa of the first user of the previous batch and the last user of the previous batch will be the secret santa of the first user of the current batch. This process is depicted in Figure 12 below.

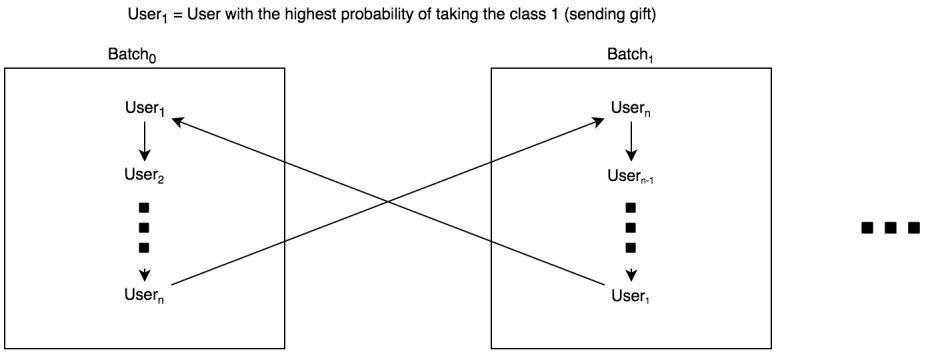

Figure 12: Arrow from $user_i$ to $user_j$ means that the $user_i$ is the secret santa of $user_j$

8 Results/Discussion

The attempt to improve the BOG's Secret Santa campaign required multiple layers of processes. These ranged from preprocessing textual data and extracting keywords to building predictive models that recommend gift items or determine an optimal matching logic. Extracting keywords turned out to be the most challenging task since we had very limited tools and short, unlabeled texts. By using the extracted keywords, we managed to build a generalized model that can be applied in the future. However, the generalized model should still be retrained on new datasets before deployment. With the help of the new matching process we managed to cut the percentage of people who sent out the gift and did not receive a gift down by 14%, compared to the previous year. Important to note that this was the main objective for the matching algorithm. Through the recommendation system, the users were connected to more than 450 different local businesses. This generated a significant revenue for the companies, as well as raising their brand awareness. Their combined web-traffic through the campaign was more than 390k hits.

In future efforts, there can be many more experiments conducted on the different ways to build these predictive models. There is still plenty of unexplored methods in the space of possibilities. A completely different method can be introduced for matching. For example, instead of just looking at the users' information, one can observe the whole data and the users as a graph or collection of graphs. The problem then can be formulated as graph optimization. Although, the issue with having to complete matching in a streaming fashion will still remain. We used binary ratings for the recommendation system, this could be changed into numerical ratings, by aggregating the keywords into defined categories. By using this representation other types of methods could be applied, like matrix factorization, which usually shows to be quite promising when it comes to predicting likely relevant items for a user [6].

The keyword extraction model could be improved by adding multiple layers to the decision making, for example considering punctuation, a word that is between two commas will most likely be a keyword. An extensive stop word list could be generated to filter out words that fall in the likely contexts of a keyword but aren't actually a keyword: "I like him", "I love everything" and etc. Furthermore, the context based keyword extraction could be improved by applying more sophisticated models like RNNs or CNNs, which usually perform very well for text related problems [7, 8].

No matter how sophisticated the governing models are, there will always be a subset of users signing up without sending a gift. Therefore, future efforts to improve overall participant satisfaction may consider focusing on remedying the situation for participants who sent but did not receive a gift. One potential method to consider would be to give these participants the option to be included in a published pool of users where anyone could then sign up to send a gift to the disappointed participant. Another method to consider is to allow users to request a rematch if they are unhappy with their assignment. Furthermore, setting a lower bound on the number of characters that the users write in their *Interests* might be helpful, since it may weed out people who are not willing to invest time and give effort. These users have a smaller likelihood of actually fulfilling their promise of sending the gift, as suggested from the data.

Overall, analyzing the data provided from 2016 did prove beneficial in improving 2017's campaign. A large portion of 2017 had to be dedicated to creating the initial tools to analyze the user-inputs. In future years, these tools already being present may allow for more time to be spent on analysis of the data and their correlations rather than on initial preperations. Future campaigns may then use both these tools and additional data to continue improving the full process, from proper communication to the users to improving the experience of disappointed participants.

Acknowledgements I'd like to thank my mentor Jeff Phillips who has helped me immensely throughout the whole process. He was always willing to meet with me to discuss various specific aspects of the project. His input on the problems I was facing gave me the proper guidance that was needed to meet and in some cases exceed the objectives of the project. Another faculty member at the University of Utah that deserves a mention is Vivek Srikumar who gave me great advice on some specific NLP challenges that were present. Also a huge thank you the whole Pulsar AI team. Thank you to Ana Kolkhidashvili for her crucial input on the understanding of the linguistics of the Georgian language, and thank you to Giorgi Guliashvili, an excellent engineer who contributed to the proper noun keyword extraction process and the preparation of the models for deployment. And finally a special thanks to Stephanie Holzkamp for helping me with editing and revision.

References

[1] Sherry, J. F. Gift Giving in Anthropological Perspective. In *The Journal of Consumer Research, 10(2), 157168.*, 1983.

[2] Belk, Russell W. Gift-Giving Behavior (Faculty Working Paper No. 449) (p. 50). In *TCollege of Commerce and Business Administration, University of Illinois at Urbana-Champaign*, 1977.

[3] National Retail Federation Holiday Retail Sales Increased 5.5 Percent in 2017, Exceeding NRF Forecast and Showing Strongest Gain Since Great Recession (Press Release). *Washington DC: National Retail Federation.*, 2018. `https://nrf.com/media/press-releases/holiday-retail-sales-increased-55-percent-2017-exceeding-nrf-forecast-and`

[4] Eight Things to Know About Redditgifts `https://www.redditgifts.com/about/`

[5] Marujo, L., Ling, W., Trancoso, I., Dyer, C., Black, A., Gershmann, A., Martins de Matos, D., Neto, J., & Carbonell, J. Automatic Keyword Extraction on Twitter. In *53rd Annual Meeting of the Association for Computational Linguistics (pp. 637643). Beijing, China: Association for Computational Linguistics.*, 2015.

[6] Bokde D., Girase S., Mukhopadhyay D. Role of Matrix Factorization Model in Collaborative Filtering Algorithm. In *A Survey. International Journal of Advance Foundation and Research in Computer (IJAFRC) Volume 1, Issue 6.*, 2014.

[7] Wang X., Jiang W., Luo Z. Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts. In *COLING*, 2016.

[8] Yin W., Kann K., Yu M., Schutze H. Comparative Study of CNN and RNN for Natural Language Processing. `https://arxiv.org/pdf/1702.01923.pdf` 2017.