An Automated Approach for Information and Referral of Social Services using Machine Learning

Madhav Sharan[2], Nithin Krishna Ottilingam[1,2], Chris A. Mattmann[1,2], Karanjeet Singh[2], Maribel Marin[1], Amy Latzer[1], Colin Foon[1], Umesh Handore[1]

[1] Research Division
211 Los Angeles County
San Gabriel, CA 91778 USA

[2]Computer Science Department
University of Southern California
Los Angeles, CA 90089 USA

E-mail: cmattmann@211la.org

Abstract

The Information and Referral Federation of Los Angeles County (211 LA County) is a nationally recognized service center that makes referrals to those in need of social service resources available at sites throughout Los Angeles County and nationally for those in need and for at risk populations. Referrals are currently made using an on line web-based referral system backed by a rich highly curated dataset collected over years and informed by a national taxonomy of social services. In support of resource referrals both for our on line system, and for a new website presence, our research team has investigated and realized an automated resource referral system that learns from a caller's demographic information and historical referral data collected by human experts to recommend sites at the time of an active call. This system leverages a state of art multi label neural network classifier, tuned by grid search for obtaining the best hyper parameters for this system. The automated approach we have created allows 211 LA County to interactively provide a meaningful referral to those in need. In this paper we describe our evaluation strategy and accuracy of our system on a one year dataset containing over 450 thousand calls.

1. Introduction

The Information and Referral Federation of Los Angeles County (211 LA County) maintains and curates a decadal information system based on the most widely used taxonomy of social services [1], recognized as the industry standard by 211s and thousands of information and referral providers which operate in the United States and Canada. Anyone in need of any help can reach out to them through a phone call and trained Community Resource Advisors (CRA) are available to help. During a call a CRA rapidly searches through available data to help callers find services they need. While these calls are optimized through the use of information technology such as a state of the art information retrieval-based web search system for community resources, resource referrals are not always repeatable, and there is often hegemony of site referrals based on experience rather than data-driven decision making.

The CRA's job generally is performed in a high paced environment with the added pressure of performance standards for productivity and quality assurance. This pressure is compounded when interacting with an at-risk population. CRAs at 211 LA County are constantly faced with calls about domestic abuse, substance abuse, suicide threats, hunger and poverty [6]. In addition to these constraints, CRA also handle a heavy call load. During peak hours 211 LA receives about 650 calls per day on an average. A CRAs handle 7-8 calls per hour on an average during peak hours. Hence they get little or no downtime between calls. Though median call time is between 7-10 minutes, some calls extend over 30 minutes. Due to the nature of calls that 211 services e.g., emergency; disaster, loss of life, suicide, etc., CRAs need to answer calls with increased diligence.

Topical research has indicated that the average cost of hiring and training a call center agent totals to $8,800 [5]. This creates an immense financial incentive on publicly funded organizations like 211 LA County to do as much as possible to retain highly trained staff [11] . Tools that make a CRA's jobs easier and more effective, contribute greatly to how satisfied they are with their jobs and clients benefit by receiving higher quality service. Contemporary CRAs face a steep learning curve and it takes at least two years for a CRA to become completely proficient. At 211, new employees struggle to gain mastery over the information system quickly and are relatively slower while searching and understanding relevant referrals during an active call. Also in general CRAs are often known to refer certain sites repeatedly for multiple callers since they receive positive feedback from repeat callers or through personal knowledge about sites. These preferences remain personal and are never shared with other CRAs to collectively improve the quality of referrals. Compliance with quality assurance standards per 211's government contracts is dependent upon the accuracy of client assessment, appropriateness of referrals provided and accuracy in data collection regarding the call transaction. These components all depend upon CRA acumen and manual processes so quality is impacted by the CRA's variable call handling expertise and their capacity to manipulate the tools they are provided with.

To enhance CRA repeatability; accuracy of client assessment, appropriateness of referrals, and in data collection, our research unit at 211 has created an automated referral system to address the aforementioned concerns. Our system aims to reduce overall call time, by significantly reducing the time CRAs spend to exhaustively search the system. It aims to flatten the steep learning curve faced by CRAs as they will be able to refer resources quickly even before they obtain complete mastery of the system. Any reduction in call time contributes towards time that CRAs could use to unwind between calls. Since our system trains on data from multiple CRAs, it learns individual preferences and ensures collective intelligence. Additionally, it can plug into 211 LA County's existing call systems with minimal friction.

Our automated system employs machine learning to: (1) learn how CRAs make referrals to sites based upon caller information and based upon 211's human services taxonomy [1]; (2) is web-service based allowing for integration into both the existing 211 customer call center system and the new 211 website; (3) evaluate the accuracy of predictions; and (4) develop a basis for an automated "bot" that can interact with those in need and get them rapidly to sites and services that can provide help. Though our recommendation API and REST service support the bot and its interactions, we restrict the focus of our description in this paper to the 211 data, and the development and evaluation of the recommendation algorithms. We will describe our bot and REST-ful API in greater detail in a future work.

The remainder of this paper is organized as follows. In Section 2 and 3 we introduce our automated referral system and describe our dataset and it's characteristics. In Section 4 we present our approach to create the recommendation model and in Section 5 we present the experiments and results. Finally in, Section 6 we share our experiences and our ideas for future work.

2. Automated Referral System

The existing infrastructure for 211 LA County's call center information management system is shown in the top left-right portions of Figure 1. The bottom portion of the figure shows the new automated referral system and its integration.

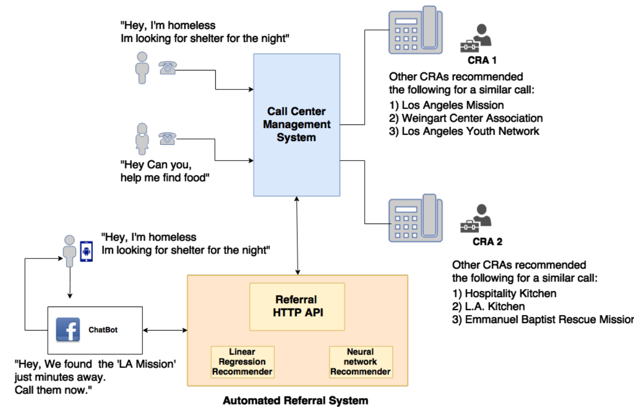

Figure 1: Automated referral system

The bottom area of the architecture demonstrates the recommendation algorithms including both a linear regression baseline algorithm; and our neural network (deep learning) based recommendation algorithm. Both of these components utilize machine learning techniques to learn from past data and predict potential useful service suitable to a caller based on caller's geographical information and answer to survey questions. The algorithms are exposed via a REST-ful application programming interface (API) built using the Python Flask web framework and are shown in the middle-top portion of the system architecture from Figure 1. When a CRA connects to an active caller they inquire and log basic caller demographic information, and 211 LA County's call center management system queries the referral system with logged information, via our new REST API.

The recommended sites are then instantly provided to CRA which can be used to curate a final set of referrals for the caller - or eventually as a mechanism to automatically (in certain calls) provide the entire caller to needed services transaction. Our new automated referral system also powers a new "bot" automated dialog system - shown in the bottom left of the architecture as "Facebook Chatbot". The bot is exposed through the Facebook API and can interact with users collecting their demographic information and answers to survey questions and then can query our referral recommendation REST API to get resources that user might be interested in and direct user to those sites.

In the following section we will explain the data collected that is used to train our new recommendation algorithms for 211.

3. The 211 Dataset

To train our recommendation algorithms we exported one year of 211 LA county call data (2016-17). The data set contains 454,924 call records. Each call is associated

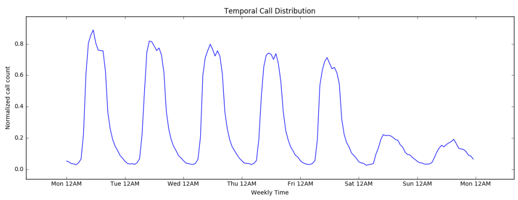

Figure 2: Temporal Call Distribution

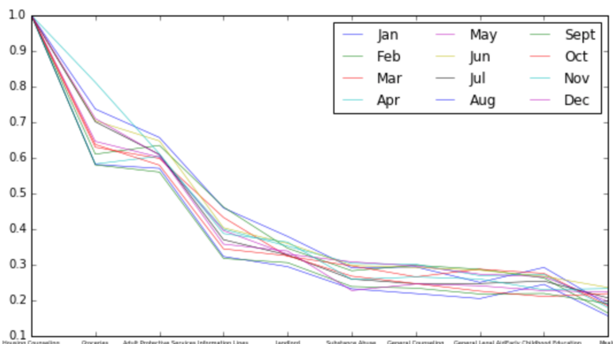

Figure 3: Seasonal Intent Distribution

with a set of sites which the caller was referred to by the CRA. One CRA takes possibly many calls and the dataset contains 1,316,399 referrals made by those CRAs. The median number of referrals per call is 3 indicating that a caller may be referred to multiple services on a single call. For example on a crisis call a caller may be referred to a service for emergency food and shelter; may be referred also to a medical facility for post treatment; and may be referred to site specific counseling services for long term follow up. The CRAs in their introductory conversations with the callers, ask/infer caller demographic information, through a set of pre-curated survey questions. This response information to these questions is logged into the 211 call management system. Our dataset contain 0.25% percent of calls in which the caller hangs up prematurely or the calls drop, preventing the complete collection of demographic information. We eliminate such calls from our dataset for the purpose of this study. The organization does not store any identifying information of caller due to the sensitive nature of calls. Table 1 lists all the caller demographic features collected.

We made following observations on characteristics of various features in the dataset:

- Call volume is normally distributed with mean around 11am as shown in Figure 2. Most calls are received between 10:00 am to 3:00 pm on weekdays which are considered peak hours.

- Call time is a good indicator of the kind of service the caller is looking for e.g. late night callers or callers calling on a weekend are more likely to be looking for ”Emergency Shelters”.

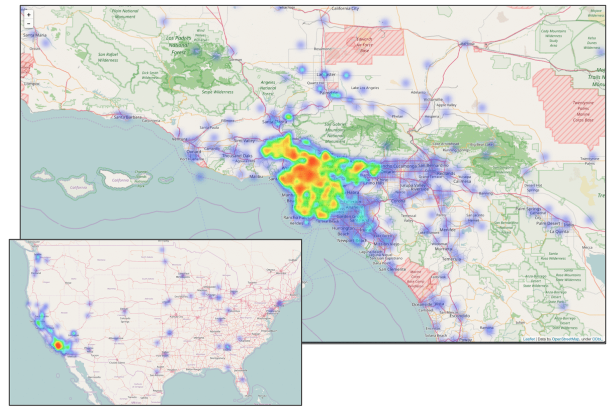

Figure 4: Spatial Call Distribution

- Call nature is mostly periodic across months. Figure 3 shows the normalized distribution of call frequency per Service problem (the reason that the caller is making contact with 211) over each month for the top service problems. We can see that the relative frequency of calls for the top service problems remains consistent across months.

- Neighborhoods around downtown and south Los Angeles call 211 LA comparatively more than other neighborhoods as shown in Figure 4. These are the low income neighborhoods with median income lesser than the county median [9]. High income neighborhoods like Malibu, Beverly Hills and Rancho Palos Verdes produce little or no calls. We hypothesize that this trend is because neighborhoods with low median income is where a significant chunk of at-risk population in Los Angeles reside.

- A caller's location reveals a lot about what it is looking for (e.g. a caller from Downtown LA – a neighborhood where the most number of homeless people in LA live [7] – is more likely to be looking for “Emergency Shelter”).

- Caller demographic information can also be used to gauge the type service caller might be interested in (e.g., If the caller has foster children at home, it's likely that she is looking for foster care services, If the caller has served in the military, it's likely she is looking for veteran support services).

We see that with spatial, temporal and caller demographic features can be generalized into an indication of the type of services a caller might be interested in as such our key

Feature	Value Set
Call Date Time	Hour of day & Day of week
Marital Status	Single {Male/Female}/ {Male/Female/Couple} with children/ Other
Contacter	Self/Other
First time caller	Yes/No
Caller Zip, Service Zip	Zipcodes in CA
Gender	Male/Female/Other/NA
Household size	None/1-7/ 8 or more
Foster children at home	Yes/No/NA
Served in the military	Self/Spouse/NA
Number of pregnant women being served by this call	None/1-7/8 or more
Number of Children between 0-5 being served by this call	None/1-7/8 or more
Services Requested	Meals On Wheels/ Homeless Shelters/etc
Site(s) Referred	LA Mission/ Good Sheperd Center/etc

Table 1: Caller Features

hypothesis was that these features can help to predict the correct sites that can be referred to the a caller in an automatic fashion using machine learning.

In the next section we describe the algorithms for the automatic recommendation of resource referrals based on the features of 211's dataset.

4. Recommendation Algorithms

We explored a series of recommendation approaches including examining collaborative filtering algorithms; and machine learning, initially experimenting with a linear regression approach that we used later as a baseline and setting on a deep neural network (DNN) approach described later in the section.

Conventional collaborative filtering approaches for recommendation like matrix factorization (MF) utilize historical user-item preference data to predict other items user might be interested in. For example Netflix [2] recommends movies based on movies that a customer has watched and movies watched by users similar to that customer. 211 call data does not include identification features about the caller like a phone number. This is done to ensure that every caller is treated equally and also due to fact that some phone numbers are from public sources and are used my multiple people. Thus it is not possible to directly distinguish between different callers from the call records and in our case it is not possible to directly model this task of recommending sites to callers as a collaborative filtering problem.

In a conventional classification problem, data is associated to a single class from a set of disjoint classes e.g., an email 'Spam' classifier [10], labels an email as 'Spam' or 'Not-Spam'. An email can never be both 'Spam' and 'Not-Spam' at once so typically in classification the class labels are disjoint. However some in classification contexts data belongs to more than one conceptual class. This problem is referred to as a multilabel classification problem[8]. For example a piece of music might belong to multiple genres like Rap, Hiphop and electronic dance music (EDM). Genre classification of music is an example of a multi label classification problem.

We model the problem of recommending sites to a caller in our 211 approach as a multilabel classification problem. We treat each site as a class which the call could potentially belong to. We assume each call can be categorized into multiple such classes. We first discuss our approach to feature engineering, our linear regression baseline and then discuss the deep learning approach.

4.1 Data preparation

We predict the *Site(s) Referred* using the other features listed in table 1. *Caller Date Time* is represented as *Hour of the day* and *Day of the week* are both numeric features. Every feature other than *Caller Date Time* is a categorical feature. Categorical variables are qualitative, (i.e) the values of such variables have no natural order. We convert each n category, categorical feature into n boolean features (i.e., so-called "dummy variables)"; which has value 1 if the record belonged to that category and 0 if not [4]. This scheme also helps up represent multilabel features like *Service Requested* / *Site(s) Referred*. The input feature vector is a combination of call time features, caller demographic features and services requested. The output feature vector is a set of boolean features which indicate the set of sites referred to a given call. After expansion we have a total of 1,166 input features and 596 target features.

4.2 Linear Regression

For a call c given an input feature vector $x_c \in R^m$; For each site s, we predict y_s the likelihood of the site s being referred in call c using a linear regression model: $y_s = w^T x_c$, where w^T is learned from the training data. We then return the top k sites $s_1, s_2..s_k$ with the highest y_s values as the referred sites.

4.2.1 Results

With L1 Regularization $\lambda = 0.1$ the linear regression model produced $\approx 65\%$ recall while predicting top the 10 sites and $\approx 55\%$ recall while predicting the top 5 sites on the test data, which we use as our baseline.

4.3 Neural Network

Deep neural networks are now the state of art to solve many learning problems [12]. With a large number of training examples, deep neural networks have been shown to outperform conventional machine learning algorithms. Neural networks also offer flexible designing and they support many approaches to model a problem. Our approach is to place caller features, survey questions and taxonomy one after another and feed it to a multi layer feed forward neural network. For each input example there can be multiple sites that a user is referred to. That makes our output layer representing all available sites, each neuron having a value as 1/0 depending if that site is recommended in that example or not. For one training example there can be multiple sites referred and this is very easy to support in this design by activating multiple output neurons at once. This design also ensures that each site is compared against each other so while predicting top k most suitable sites the order of k sites is also taken care of. We used Keras [3] which is a very popular library to design neural networks conveniently for all our experiments.

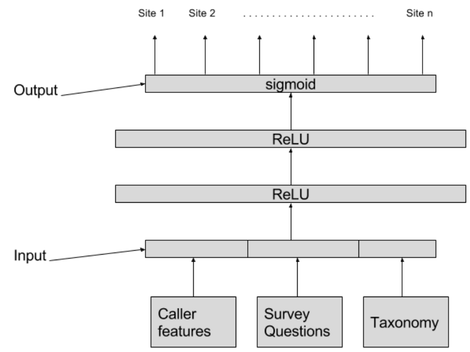

Figure 5: Feed forward multi-label neural network to recommend referral sites

4.3.1 Network optimization

Optimizing the hyper-parameters of neural network manually is a time and resource intensive task. We run a grid search/parameter sweep which exhaustively searches through through the hyper-parameter space and picks the optimal hyper-parameters. We experimented with a multiple layer architecture, loss functions, batch size and regularization constants in a grid search and finally chose one that gave us best results on validation data. We will describe our evaluation process of recommendation system in more details in section 5. Neural net tuning is described below.

In all our experiments we used callback for early stopping to stop training further if error rate on validation data increased continuously in 5 or more epochs and we used standard value for momentum. For all features combined we had input layer of $\approx$ 1500 and for the scope of the problem we had $\approx$ 1500 sites. We skipped sites which were referred less than 35 times in 4 months as we didn't have enough data to learn about them. We used ReLU activation in all hidden layers and sigmoid activation in the output layer. This helped us in training networks faster and sigmoid outputs are useful in prediction of the order of sites. Simple networks like with just one layer of 1024 neurons did not perform well and gave huge errors on validation data. We found that a network with two hidden layers of 4096 neurons worked best in our case. We chose a categorical cross entropy loss function as it is a classification problem and cross entropy are known to work best for classification problems. We compared it with other loss functions e.g., mean squared error but found that categorical cross entropy supersedes them. We selected batch size from range of $[25, 50, 100, 500]$. As we increase batch size training time reduces but we observe that network fails to learn many examples with larger batch size and the error rate is huge. We have nearly 180,000 training examples and we found best results with a batch size of 50. Our final layer representation with input features and output is shown in Fig. 5

4.3.2 Results

The final optimized network was able to give $\approx 93\%$ recall while predicting the top 10 sites for a referral and $\approx 85\%$ recall while predicting the top 5 sites on test data. We will discuss our evaluation mechanism in more details in section 5 along with a comparison against the linear regression approach. During testing we found that if we trained and tested this model only on sites with $\geq$ 850 examples we were left with $\approx$ 65 sites and we get $\approx 97\%$ recall while predicting top the 5 sites and $\approx 90\%$ recall while predicting the top 2 sites on test data.

5. Evaluation

We evaluate our model using 4-fold validation on one year data. K-fold cross validation ensures that our model generalizes well and does not over-fit the data. As we showed in Section 3 our data is periodic every month so we divided our folds such that each fold contains three months of data. We trained our model 4 times, each time reserving one fold for validation and using the other three for training.

Training time for each fold are shown in Table 2. We conducted all our experiments on a Linux machine with 8 CPU cores and 10 GB memory. On an average It took $\approx$ 4 hours to train on 9 months of cleaned data with our optimized settings. We started with 50 epochs and then observed that the model started converging after 30 epochs.

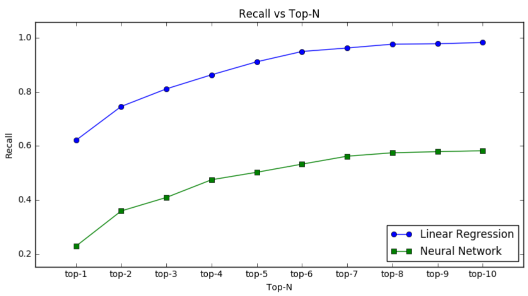

Figure 6: Recall vs top-n

The accuracy of our model on a per call basis is determined by the following metrics:

$$precision = \frac{size(Referred \cap Predicted)}{size(Predicted)}$$

$$recall = \frac{size(Referred \cap Predicted)}{size(Referred)}$$

To evaluate accuracy of the model we calculated recall and precision based on top-n predicted sites for each test example. As we see in Figure 6 recall grows as we increase top-n. The recall produced by our model was near 100% when we predicted top 10 sites. From the data and experiments we see that a CRA refers 3 sites on an average. Recommending too many sites to the CRA would be cumbersome as she would then have to determine which of the recommended sites are most relevant to the active caller. Thus we choose 5 as the ideal value for top-n.

As described in Table 3 we see that our approach gave us 90% recall on an average on all folds and 65% precision on all folds. We found out that popular sites, which were recommended many times by CRAs, were predicted correctly 95% of the time. We observed that network did not perform well for referrals and sites for which there were less than 100 examples in training data.

Testing Quarter	Training Time in hours	
	Neural Network	Linear Regression
Quarter 1	3.75	0.74
Quarter 2	4.49	0.79
Quarter 3	3.03	0.73
Quarter 4	3.07	0.73

Table 2: Training time

6. Future work

Though the early results of our work are promising more research, evaluation and implementation are planned. In the near future we are looking for more interfaces to connect with our recommendation algorithms and API in order to make it more interactive and to increase its utility throughout the organization. The 211 Facebook Bot will help in this area. In addition we will explore reinforcement learning by taking feedback from CRAs and guiding network to perform better. We also want to integrate our recommendation API directly with the 211LA website so it can be used by the broader community to find services for their needs at their own convenience.

Testing Quarter	Neural Network		Linear Regression	
	Recall	Precision	Recall	Precision
Quarter 1	0.90	0.62	0.54	0.36
Quarter 2	0.92	0.63	0.56	0.35
Quarter 3	0.93	0.65	0.56	0.31
Quarter 4	0.93	0.65	0.55	0.33

Table 3: 4-Fold accuracy — sites predicted = 5

7. Acknowledgement

The authors are thankful to the Development team at 211 Los Angeles County, especially Colin Foon for compiling the necessary data dumps to facilitate this study. This effort was supported in part by the Information and Referral Federation of Los Angeles County (211 LA County) and additionally in part by the DARPA Memex/XDATA/D3M programs and NSF award numbers ICER-1639753, PLR-1348450 and PLR-144562 funded a portion of the work.

References

[1] 211 taxonomy. `https://211taxonomy.org/`, 2017.
[2] C. C. Aggarwal. An introduction to recommender systems.
[3] F. Chollet. keras. `https://github.com/fchollet/keras`, 2015.
[4] S. Garavaglia and A. Sharma. A smart guide to dummy variables: Four applications and a macro, 2000.
[5] K. E. Jackson. Overturn the high cost of employee turnover, 2009.
[6] K. Rigden. Stress and the 911 dispatcher, 2009.
[7] J. SCHLEUSS and L. I. ELEBEE. Who are l.a. county's homeless?
[8] M. S. Sorower. A literature survey on algorithms for multi-label learning.
[9] L. Times. Median income ranking - mapping la.
[10] K. Tretyakov. Machine learning techniques in spam filtering, 2004.
[11] R. Vidal and M. McConnell. The true cost of attrition.
[12] H. Wang and B. Raj. On the origin of deep learning, 2017.