IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 1, 2019, accepted November 9, 2019, date of publication November 21, 2019, date of current version December 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2954884

Implicit Life Event Discovery From Call Transcripts Using Temporal Input Transformation Network

NIMA EBADI[1,2], BRANDON LWOWSKI[1,3], MEHRAD JALOLI[1,2], AND PAUL RAD[1,2,3]

[1]Secure AI and Autonomy Laboratory, The University of Texas at San Antonio, San Antonio, TX 78249, USA
[2]Electrical and Computer Engineering Department, The University of Texas at San Antonio, San Antonio, TX 78249, USA
[3]Information Systems and Security Department, The University of Texas at San Antonio, San Antonio, TX 78249, USA

Corresponding author: Paul Rad (peyman.najafirad@ utsa.edu)

ABSTRACT Customer-agent conversations (i.e. *call transcripts*) are invaluable source for companies as they convey direct information from their customers implicit and explicit behaviour. Identifying customer-related events is an important task in customer services which is possible from the call transcripts. However, call centers produces a vast amount of transcripts which makes the manual or semi-manual processing of such raw datasets quite challenging. Furthermore, customer-agent call transcripts tend not to explicitly denote events that might be beneficial to customer services. Albeit being highly researched across multiple domains in the literature, event detection, especially implicit life event detection have not been well examined from call transcripts due to a lack of proper large-scale dataset. In this research, we propose a novel deep learning approach based on latent topic modeling and deep recurrent neural networks with memory units to automatically detect implicit events from a customer's history of call transcripts. These implicit events are detected prior to the report date of that event thereby not containing any explicit topic/feature. We provide a case study on a real-life, large-scale data of more than 800K call transcripts from a large financial services company in the U.S. to examine the practical features and challenges of this problem. The evaluation results demonstrate the potential applicability of our implicit life event detection as it achieves a macro-recall score of 53 (macro-f1 of 47.5) on a highly imbalanced test set, negative samples are 95% of the data. Our model beats the the state-of-the-art text classification benchmarks by macro-f1 score of 5.6 and macro-recall of 8.8 on average, and performs better than the ensemble of all single-document and sequential classification benchmarks albeit being significantly less complex. The comparison results show the importance as well as our model's capability of capturing the mutual information of a sequence of call transcripts in detecting the implicit life events.

INDEX TERMS Implicit event discovery, call transcripts, deep learning, recurrent neural network, machine learning, natural language processing, text classification, topic modeling, event detection.

I. INTRODUCTION

When calling a customer or member service phone number, one typically hears the following, ''this call may be recorded for quality and training purposes''. These companies record the conversations between member representatives and customers, and pass them through a software that transcribes the conversation to a set of text-based utterances in chronological order [1], [2]. No matter the company, these call transcripts contain valuable information regarding the customer's explicit and implicit attitude [3] ranging from their opinion [4], level of satisfaction or sentiment of a product [5] to their intent [6] and deception goals [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Michele Magno.

In specific, call transcripts encompass topics about the customers' life events that influence their attitudes towards the company [8]–[10] which makes such sources of information significantly valuable for marketing and customer services applications [11]. Examples of life events include college graduation, marriage, pregnancy, divorce, retirement and new employment to name but a few. Once these topics are explicitly mentioned by a customer, either through a phone call conversation or any other communication channels, the company's representative reports the customer's life event and the date of occurrence. However, these life events are not always explicitly mentioned in a conversation with the company's agent. Furthermore, predicting whether an event may occur in a customer's life enables prescriptive analytics which is significantly important for call centers [12]. In target marketing and advertisement, if the knowledge of an implicit life event can be detected, especially prior to its occurrence, the team can begin campaigns that would attract the customer to the product or can leverage promotional events related to the life event for experiential marketing of a service, product or brand [13]. Therefore, it stands to the reason the marketing analysts investigate through call transcript databases to detect implicit life events which either have not happened yet or not explicitly mentioned by the customers.

With the expansion of call centers and development of Automatic Speech Recognition (ASR) methods, call transcript data is being generated in immense quantities [14] making human analysts ensnared by information overload. Call center marketing reviewers are able to manually analyze only a limited random portion of the data. Such services are costly and do not scale to the amount of the data generated by call centers [15]. In the light of this,the appropriate approach to analyze call transcript data is large-scale text mining.

In this research, we study the potential application of call transcripts in detecting customers' implicit life events, i.e. to detect a life event which the customer has not explicitly stated that event has occurred. We propose a novel methodology using recurrent neural networks with memory units (we call it *recurrent memory cells*) and customers' history of transcript data, represented as a sequence of transcripts, to detect implicit life events. Specifically, our model exploit temporal latent topics from a series of call transcripts to predict the likelihood of a set of life events. Furthermore, we provide a case study on a large-scale, real-life dataset of call transcripts collected at the customer services center of an American financial company, thereby revealing invaluable information regarding the potentials and challenges of this task. We investigate through various event extraction, text classification and topic modeling benchmarks analyzing their performances, weaknesses and strengths on the large-scale dataset and compare their performances to our model's. The results demonstrate the state-of-the-art performance of our model in predicting the implicit life-events.

The main contributions of this study are summarized as follows:

- We propose a novel approach based on latent topic modeling and deep recurrent neural networks with memory units which effectively captures the sequential clues that exist in a sequence of a customer's call transcripts, and performs significantly better than benchmarks in large-scale detection of life events.
- The work presented here investigates the under-examined problem of life event detection from customers' call transcripts. More specifically, our research focuses on implicit life event detection from an individual customer's call transcripts prior to their explicit mention of occurrence. To the best of our knowledge, there is no/few research regarding such problem in the literature, mainly due to the lack of proper dataset.
- We provide a case study on a large-scale, real-life dataset of call transcripts collected at the customer services center of an American financial company in order to examine the practical features of our model in dealing with large-scale, imbalanced dataset with real transcriptions.

The remainder of this paper is organized as follows. Section 2 elaborate upon the related works and research done regarding event detection across various domains, and the challenges of call-transcript-based event detection. Section 3 explains our proposed methodology including data pre-processing as well as document representation of a call transcript and sequential analysis of a customer's history of recent conversations which forms a sequence of transcripts. In section 4, a case study is presented to evaluate the performance of our methodology in comparison with state-of-the-art text classification benchmarks.

II. RELATED WORK

In this section related works in the literature are investigated from different points of view, e.g., various domains in which event detection has been exploited, different kinds of events as well as different datasets used in previous works.

A. EVENT DETECTION ACROSS ALL DOMAINS

With the increasing number of real-world events that are originated and discussed over different sources such as news, blogs, social media and call transcripts, extracting valuable information from these sources in a timely manner is important since this information is useful for government, health organizations, companies and political sciences. Following the long history of well-studied event detection, lots of various techniques have been proposed to extract/detect events from contexts, articles and transcripts, a significant amount of which have investigated datasets from social media. For example, Lwowski *et al.* [16] uses emotions presented in Tweets for a given city merely using text-based contents. Similar projects are presented across various social media platforms and online information resources such as Facebook, Google Plus [17], Twitter [18] as well as news articles [19] to name but a few. These approaches, however, are mostly focused on public, economic or political events. On the other hand, Li *et al.* [20] proposed a method that is capable of automatically extracting the major life events of ordinary users from their published tweets as well as generating fine-grained descriptions. In many of the prior works on event detection, ACE2005 dataset has been used as the benchmark due to the precise definition of the task, promising capability in capturing contextual information and being an appropriate dataset for evaluation [21], [22].

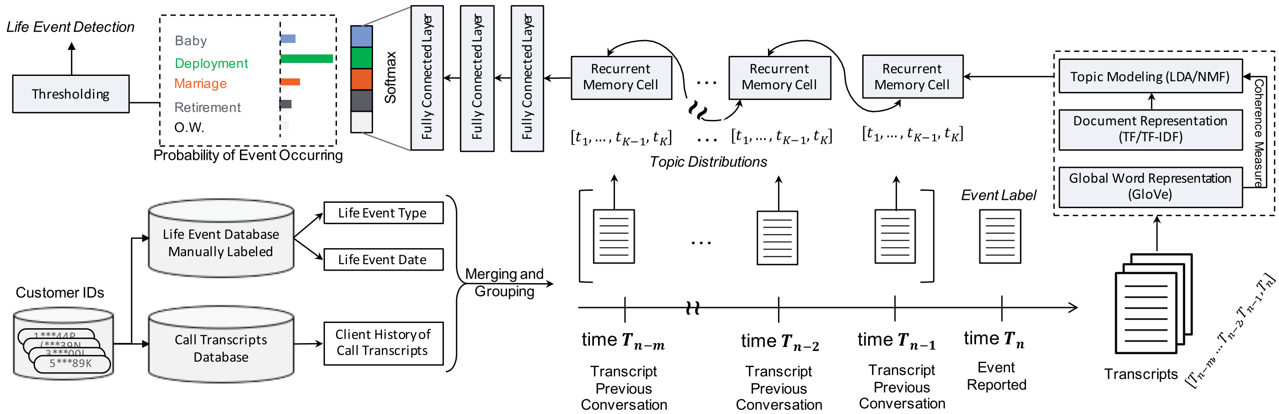

FIGURE 1. Visualization of the model presented for implicit life event event detection from call transcripts, Temporal Input Transformation Network (TITN). First, input sequences of transcripts are extracted from life event and call transcript databases. Moreover, latent variable of the input transcripts are modeled using the call transcript database and inputs are transformed into their latent topic distribution. Next, temporal analysis of the sequences of call transcripts latent variables is performed through the bidirectional neural memory cells which are stacked recursively. Finally, the output is passed through several fully connected layers to estimate the likelihood of various life events.

B. LIFE EVENT DETECTION FROM CALL TRANSCRIPTS

Call transcripts of the call centers generated from client-agent conversations could also be an appropriate datasets for event detection/extraction. These vast amounts of recorded speeches are usually stored but typically not processed for further reasons while considering direct client feedback, complaints and personal opinions, call transcripts are invaluable source of information from customers to the companies.

Zhong and Li [23] used actual customer phone call transcripts of a telecommunication service provider as dataset and developed NLP and Supervised Convolutional Neural Network (CNN) algorithms for customer churn prediction.

Nonetheless, there are some difficulties in processing Call Transcripts making them challenging to be used for extracting significant life events purposes, which addresses the reason of lacking previous work on these kind of datasets. In the other words, life event detection from large-scale dataset of call transcript is by itself a challenging task due to the lack of proper dataset. Real-life call transcript datasets are large-scale, noisy, and with inaccurate annotation. Thus, any life event detection model should be able to deal with issues such as scalability, noisy transcription, inconsistency in data annotation to name but a few. Obviously these challenges are signified once attempting to detect an individual customer's implicit life event from transcripts prior to the occurrence of that event. Followings are some of the main challenges confronting implicit life event detection:

Vague Relationship Between Customer Calls and Major Events: the relationship between customers' calls and network events is blurred because customers respond to an event in different ways. So classifying these responses would be challenging [9].

Inconsistent Labeling Across Different Agents: Client-agent conversations are not usually labeled in a unique manner, in the other words, labeling is personalized across each agent or call center.

Noisiness of speech-to-text transcription: Many important events cannot be detected by looking at calls in one category. On the other hand, aggregating calls from different centers would be challenging as well. Thus, we need a classifier that is robust against noise and different response time.

C. IMPLICIT VS. EXPLICIT

To the best of our knowledge, most of the research already performed in the field of sentiment analysis and event detection/extraction are based on explicit expression of emotions and sentiments. Sometimes, non-word expressions, hesitation, conservatism, etc. are a part of human communication. However, in most cases, emotions and issues (e.g. happiness or sadness) are not stated by using words with a direct meaning, but indirectly, by presenting situations that based on commonsense knowledge can be interpreted in an effective manner [24], [25]. Here in this paper, we try to extract implicitly expressed events from a real-life call transcript dataset from a large company and compare it with various text classification/event detection benchmarks. Extracting implicit issues and emotions of a customer from a client-customer conversation can be a very good source of background information for a company in target marketing. Using this information, a company can have a better prediction of the customer's need and is able to define better targets in their future customer services.

III. METHODOLOGY

This project formulates the problem of implicit life event detection as a multi-label text classification problem [26] where every customer's history of call transcripts, prior to the explicit report date (thereby implicit), is treated as a sequence of documents whose label(s) are the reported life event. However, text classification benchmarks look at a single document, in this case transcribed conversation, to detect whether the implicit life event cues are present or not. This approach has a major downfall that is there could be valuable information present mutually in multiple conversations between customer and agent that could be beneficial in correctly predict the life event. With the use of deep recurrent neural networks, the accuracy of predictions can be greatly increased by treating the history of customer transcripts as a sequence. Recurrent neural networks are proposed in the literature to capture the sequential dependencies within the various steps of a sequence, and perform a specific classification or modeling based on these contextual dependencies [27], [28]. Following the guide-line provided by [29], this study leverages such architectures to detect implicit life event which either has happened but not explicitly reported yet or is going to happen in near future.

In this section, we present our novel approach for the problem of implicit life event detection which is based on deep recurrent neural networks. It contains a recurrent memory cells which capture the joint information between latent variables and cues present in a sequence of recent call conversations history from an individual customer that can denote implicit life events related to them. In the following sections, we elaborate upon various stages and modules of our methodology which includes data pre-processing, document representation, latent variable and feature extraction along with deep recurrent neural network.

A. OVERVIEW OF THE MODEL

As shown in 3, our methodology contains two databases of life events and call transcripts. The former is the reported life events for every customer along with the report date and the customer ID assigned to them. The later contains every customer's call transcript along with their customer ID and the date. By merging and concatenating these two databases using the customer IDs, we extract a history of call transcripts and life events for every customer. This technique generates a dataset of transcript sequences which are annotated with customers' life events. In order for our model to learn the implicit indicators of life events, we eliminate the transcripts whose dates are the same or after the life event report date, and merely focus on the conversations that occurred prior to the explicit mention of life events. After pre-processing and cleaning the transcripts of the merged data, we utilize it for training and testing purposes.

As mentioned in the beginning of this section, we formulate the problem of implicit life event detection as a multi-label text classification problem. In this regard, firstly, we extract the bag-of-words representation of every call transcript. We split the data into training and testing set, initially, and compute the global parameters for the bag-of-words representation algorithms from the training set only. Next, we model the latent variables of the call transcripts using these representations and transform the input data into its latent space, i.e., every transcript is represented in the form of its latent topic distributions. Afterwards, these sequences of latent topic distributions are fed into a customized deep neural network that we designed for this research based on deep recurrent neural and memory networks. This recurrent neural network captures the cues of a sequence of call transcripts latent topic which indicate an implicit life event and after passing through a several fully connected layers outputs the likelihood of various life event types. Using an appropriate threshold, our model can detect events that potentially occur in the customer's life although they may have not explicitly expressed them in their conversations.

B. BAG-OF-WORDS REPRESENTATION OF CALL TRANSCRIPTS

As for the representation of call transcripts, we use bag-of-words scheme based on TF/TF-IDF weighting. In this regard, we initially, parse the transcripts to the word level using phrase-based tokenization scheme of [30] to distinguish phrases from single words. Next, we remove the stop words using NLTK 2.0 package [31].[1]

We extract the TF/TF-IDF vector representation of all transcripts by fitting the global parameters on the training set. Then for every transcript, the following equations are used for TF and TF-IDF weighting scheme:

$$\mathrm{tf}_{w,d} = 0.5 + 0.5\frac{f_{w,d}}{\max\{f_{w',d} : w' \in d\}} \tag{1}$$

$$\mathrm{idf}_{w,D} = \log\frac{|D|}{|\{d \in D : w \in d\}|} \tag{2}$$

$$\text{tf-idf}_{d} = \mathrm{tf}_{w,d} \cdot \mathrm{idf}_{w,D} \tag{3}$$

where $\mathrm{tf}_{w,d}$ and tf-idf_d respectively represent the TF and TF-IDF weight of a word in the call transcript d; as for TF we use double normalization 0.5 scheme [32]. $f_{w,d}$ is the number of times that word appears in the article, D represents the whole corpus of transcripts and $|D|$ is its size, $f_{w,D}$ is the number of transcripts in which the word appears [33].

Afterward, we utilize these representations to model latent variables/topics of the call transcripts, and transform the input into its latent space whose temporal information can be leveraged for implicit life event detection.

C. INPUT TRANSFORMATION VIA LATENT VARIABLE MODELING

Many studies in the area of deep learning have shown that transforming the raw input data into the latent space using latent variable models, and then feeding the latent variables into the learning model improves the performance and robustness [34]–[36]. Moreover, such transformation decreases the dimension of the input data (while maintaining the key patterns) as well as the complexity of the deep learning model; therefore, the overall model requires fewer training samples [37], [38]. Inspired by these studies, we extract latent variables of call transcripts in the form of topic distributions using topic modeling algorithms which have shown promising performance in the literature. As such, topic distributions

[1] We customize the set of stop words iterative based on the performance of our latent topic modeling approach which is explain in section III-C

are modeled initially from the whole training dataset of call transcripts. Next topic distributions are fed fed to the deep learning model as call transcript representations instead of the bag-of-words ones. We devise our topic modeling module (as the latent variable model for call transcripts) based on Box's Loop [38], [39] identifying the hyperparameters and set of proper stop words which yield satisfying performance. In this regard, we use two popular methods of topic modeling: *Linear Dirichlet Allocation* (*LDA*) and *Nonnegative Matrix Factorization* (*NMF*).

1) LINEAR DIRICHLET ALLOCATION (LDA)

Latent Dirichlet Allocation is an unsupervised machine learning algorithm that is used to extract and group text based on topics that are present. The LDA model is fed vectorized text and given a number of topics to discover and the output are defined clusters equal to the number of topics with words that belong in that topic. Once the LDA model is trained, a single document can be fed into it and a probability distribution is outputted. This probability distribution vectors length is equal to the number of topics chosen during training. Each position of the vector represents the probability that topic is present in the document.

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \quad (4)$$

where $\beta_{1:K}$ are the topics, where each β_k is a distribution over the vocabulary, θ_d are the topic proportions for document d, $\theta_{d,k}$ is the topic proportion for topic k in document d, z_d are the topic assignments for document d, $z_{d,n}$ is the topic assignment for word n in document d, w_d are the observed words for document d. [40], [41]

2) NONNEGATIVE MATRIX FACTORIZATION (NMF)

Similar to LDA, nonnegative matrix factorization is a dimensionality reduction method attempting to reduce the size of a data which is originally represented as a nonnegative matrix. However, unlike LDA, NMF is a linear algebric optimization algorithm which simply approximate/factorizes a give nonnegative matrix to the product of two nonnegative matrices [42], [43]. In the context of call transcripts topic modeling, it transforms the bag-of-words representations of call transcripts corpus into out product of two matrices of transcript-topic matrix, every row of which correspond to a transcript document that is represented as a distribution of topics, and topic-word matrix, every row of which correspond to a topic that is represented as a distribution of words in the vocabulary. The following is the mathematical formulation of NMF:

$$W_{1:D} = \theta_{1:D} \times \beta_{1:K} \quad (5)$$

where $W_{1:D}$ is the bag-of-words representation of call transcripts which is a $D \times V$ matrix, each row represents every call transcript document. $\theta_{1:D}$ refers to the topic distribution

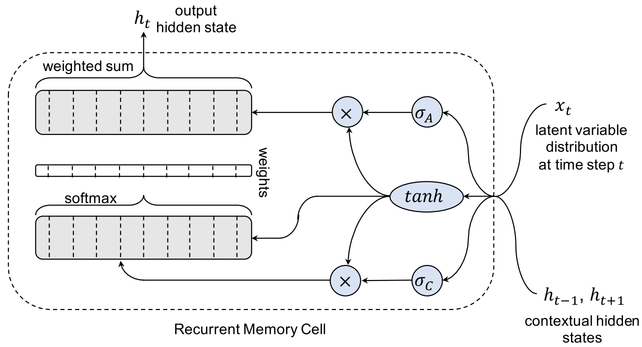

FIGURE 2. The structure of our proposed recurrent memory cell. Every cell is fed by latent variable (topics) distribution at every time step as well as the hidden states from the previous and next steps (since the recurrent structure is bidirectional) which we refer to as contextual hidden states. Contextual hidden state gets concatenated with the distribution and through two input gates σ_A and σ_C get stored in the tensor memories of every cell. The output is computed using an attention over these memories.

of every corresponding transcript and $\beta_{1:K}$ is the distribution of words for every topic indexed 1 to K.

To ideally model the latent variable of topics from the training set, we leverage a similar scheme to Röder *et al.* [44] to find the optimum topic modeling scheme and its hyperparameteres (especially the number of topics) are selected based on the coherence measure below:

$$C_k = \frac{2}{N_t \cdot (N_t - 1)} \sum_{i=1}^{N_t - 1} \sum_{j=i+1}^{N_t} PMI(w_i, w_j) \quad (6)$$

For every set of N_t top words of every topic it calculates an averaged sum of confirmation measure over all word pairs Where N_t is the number of top words of a topic set PMI is the pairwise mutual information between every two words in each topic which are calculated based on context vector for every word and vector similarity measures such as cosine similarity [45]. In this regard, we use global vector representation (GloVe) for every topic top words context vector representation (i.e. word2vec) which were trained on two billion tweets [46].

After designing the optimum latent variable model for call transcript topics, we require an approach to capture the sequential information in a sequence of call transcript history which we elaborate upon in the next subsection.

D. TEMPORAL ANALYSIS VIA RECURRENT NETWORK OF MEMORY CELLS

As mentioned in section III-A, there is mutual information in a sequence of call transcripts that may implicitly indicate possibility of existence of a specific life event. In order to capture the sequential dependencies of call transcript sequences, we customize an approach based on deep recurrent neural network known for their vigorous performance [47]. Long-term memory is necessary to determine the implicit existence of a life event as these sequences can be 20 steps of call transcripts each of which contains detailed information indicating cues about the event. However, a vanilla RNN architecture does not maintain memory as data is passed into the next state of the sequence. If an important piece of information is mentioned several conversations ago, the vanilla RNN would have a hard time maintaining that knowledge as it moves through the sequence.

Long short-term memory (LSTM) and gated recurrent unit (GRU) are variants of recurrent neural networks that are designed to remember past information in a long sequence [48]. Inspired by these variants, we design our customized recurrent neural network using similar architectures. The architecture encompasses a memory that allows the model to keep memory of topics present in conversations from the past and use it to make predictions on the next step. Unlike LSTM, in the context of this research, our methodology does not need to forget past information because the sequences are not very long, and every step is simply the distribution of latent variables. As such we want to store all the information without losing the details and since we are using latent topics the complexity of our model enables using large-scale memory units (larger in comparison to LSTM and GRU). Therefore, inspired by Weston *et al.* [49] and Sukhbaatar *et al.* [50], we devise our recurrent memory cells to capture the temporal cues of the transformed input into its latent variables (hence we call the overall model Temporal Input Transformation Network). We feed the distribution of latent topic variables of every step of call transcript into the recurrent neural network (i.e. on the sequence level). Every recurrent memory cell of our model includes two memory units A and C and performs the following three operations: input generalization, memory update and output. The mathematical formulation of each step is presented as follows:

1) INPUT GENERALIZATION

Similar to LSTM, the input topic distribution gets combined with the previous and next memory cells' outputs (bidirectional) using a linear activation function *tanh*:

$$\tilde{h}[t] = \tanh\left(V_h\theta[t] + U_{h\_}h[t-1] + U_{h_+}h[t+1] + b_h\right) \quad (7)$$

where $\theta[t]$ corresponds to the input vector at time t, V_h to the connection matrices applied to the model's input, $U_h-/+$ to the recurrent connection matrices and b_h) to the bias vectors.

2) MEMORY UPDATE

Memory units A and C gets updated with the generalized input $\tilde{h}$, passed through corresponding input memory gates.

$$\sigma_{A/C}[t] = \sigma(V_{A/C}\theta[t] + U_{A/C\_}h[t-1] + U_{A/C_+}h[t+1] + b_{A/C}) \quad (8)$$

Similarly, $V_{A/C}$ to the connection matrices applied to the model's input, $U_{A/C}$ to the recurrent connection matrices and $b_{A/C}$) to the bias vectors, and $\sigma_{A/C}[t]$ is the corresponding input gate for each memory unit:

$$m_{A/C}[t] = \sigma_{A/C}[t] \odot \tilde{h}[t] \quad (9)$$

where $m_{A/C}[t]$ is the memory unit value for time step t.

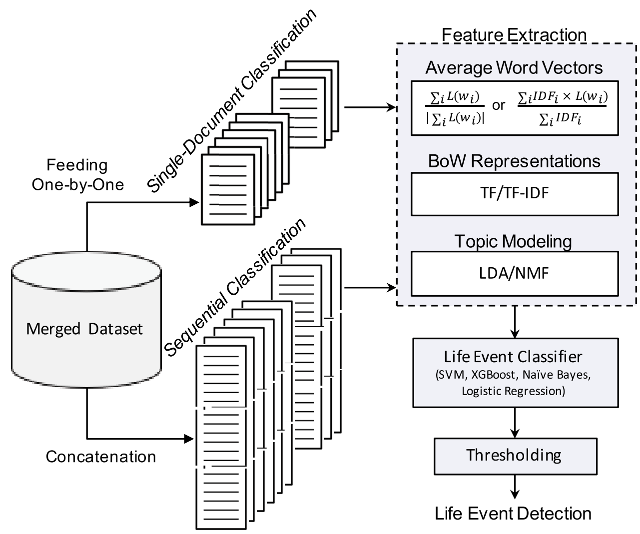

FIGURE 3. Visualization of the text classification benchmarks. Single-document classification benchmarks feed the call transcripts (each has the annotation of its corresponding sequence) one-by-one to the feature extraction, whereas the sequential ones concatenate a sequence of call transcripts into a bid document prior to feature extraction. The concatenated transcript has the same annotation of the sequence.

3) OUTPUT

The output $h[t]$ is computed through a weighted summation of memory C whose weights are calculated from the inner product of the memory unit A and the generalized input $\tilde{h}$ which find the relevant pieces of memory which are required to be passed to the next and previous steps:

For every m_A slot in the memory unit A, we compute the

$$p[t] = \text{Softmax}(m_A[t]^T\tilde{h}[t]) \quad (10)$$

$$h[t] = \sum_t m_C[t]p[t] \quad (11)$$

The final predicted response will be computed from the last memory cell's output passed through several fully connected layers and a softmax operation on the final estimated life event likelihoods, illustrated in Figure 3. The mathematical formulation for three fully connected layers is as below:

$$e = \text{Softmax}(U_{fc_3}(U_{fc_2}(U_{fc_1}h[t_l]+b_{fc_1})+b_{fc_2})+b_{fc_3}) \quad (12)$$

where $h[t_l]$ is the last memory cell's output, and U_{fc_i}s and b_{fc_i} are the connection matrices and biases for the ith fully connected layer. e is the final estimated life events' likelihoods, which is a vector with the size of the number of considered life event types.

IV. EXPERIMENTS AND EVALUATION

In this section, we elaborate upon empirical experiments and evaluation results on a real-life, large-scale dataset. We use more than 800K call transcripts from the call center of a large financial services company in the US and provide a case study which reveals the practical features of our model. First, we discuss the descriptive of the dataset and the details of our experimental settings. Then, we investigate through the ablation analysis of our model, and finally, through comparison with text classification benchmarks, we demonstrate the state-of-the-art performance of the model as well as its capability in managing large-scale, imbalanced data.

A. DATA

The dataset provided to this research consists of 2 sets i) life event set and ii) call transcript set. One is a list of life events reported about every member along with the report dates of these events. These life events are captured either by the call center agent from the conversation they have with the customers[2] or reported to the company through other communication means. This set contained more than 5 million observations extracted from 2.86 million customers. The variables of the dataset include unique customer_id which is a big integer key assigned to every member. Another variable is type which is a string value that identifies the life event name. *Move*, *divorce*, *death*, *baby*, and *sell_car* are few examples of life event types in the data present. Other variable is date_time which is a date and time object, from January 2018 to December 2018, that denotes when an event has been reported to the company. date_time values are in microsecond precision with specified time zones.

The second dataset is a collection of more than 800K call transcripts between customers and agents occurred during August 2018. The dataset variables include customer_id which is exploited in merging the call transcripts set with the life events one, worker_id that is the key identifying the agent worker, start/end_time a date and time object, in the same format of that of the life event dataset, denoting the time the conversation is start/end, and conversation which is made up of indexed list of utterances in the conversation. On average, each conversation has 132.6 number of utterances, ranging from 4 to 3,819. For this project, we utilize the conversations with less than or equal to 470 utterances (which compose more than 99% of the dataset).

Every conversation consists of the following values: channel_index an index defines who is speaking, the member or the agent, duration utterance duration in milliseconds, index index the order in which each utterance occur in the conversation, and transcript which is the utterance in string format. The sensitive information in the transcripts have been censored and approved by the companies lawyers for use. For example, in order to keep customer information secret, customer information is transformed and hidden using string tokens like [redacted phone number] and [redacted address].

The five catagories of *baby*, *retirement*, *deployment*, *marriage*, and *other*[3] are chosen as for the evaluation case study of this research. If the model is able to predict that a member is about to be deployed or get married the company sees it as a win for them.

TABLE 1. Life events and their call transcript frequencies and number of customers flagged with these events. This set contained 865,869 calls from 128,577 customers. It is important to note that in this dataset, observations are not unique customers; that is, there can be one customer with more than one life event reported.

	No. of Calls (N = 865,869)	No. of Customers (N = 128,577)
Marriage	22,841 (2.64%)	2,979 (2.32%)
Deployment	6,303 (0.73%)	1,288 (1.01%)
Baby	2,713 (0.31%)	787 (0.6%)
Retirement	1,451 (0.17%)	530 (0.4%)
Other	832,561 (96.15%)	122,993 (95.66%)

In order to transform the dataset into sequences of transcripts, all transcripts are grouped by the unique customer_id and sorted by date and time of phone calls. What is created from this grouping are sequences of chronological transcripts from a customer to the call center. Once the grouping is completed, we merge the transcripts with the life event set using the tags *baby*, *retirement*, *deployment*, *marriage* and *other*. The tags are used to collect all transcripts from every individual customer prior to the agent's assigning the label. Doing so creates sequences of conversations that lead up to the life events. The descriptive of the merged dataset is shown in Table 1.

In this research, we use the sequences with the lengths of more than or equal to five call transcripts. Once these sequences of transcripts are extracted, there are 89,322 sequences of customer transcripts, out of which, 1,098 of the transcripts have the label *deployment*, 2,156 are labeled *marriage*, 613 are labeled *baby*, 479 are labeled *retirement* and the remaining 84,976 have the label of *other*.

B. SETTINGS

In regards with the hyperparameters of our model, i.e. vocabulary size, learning rate, batch size etc, we use a grid search to find the optimal values that lead to the highest accuracy on the training set. The hyperparameter values are shown in Table 2.

For the topic modeling part of our methodology, we use TF/TF-IDF bag-of-words representation methods with vocabulary size of 5K.[4] The topic modeling components, i.e. vectorization global features as well as the topic modeling features itself, are trained using the train set only to avoid data bleeding [51]. Out of the 71,457 sequences in the training set consisting of anywhere from 5-20 transcripts long, 40 topics

[2] After every conversation between a customer and an agent ends, the agent must update the customer profile with the reason why they called

[3] Call transcripts which are either not labeled or labeled with another life event other than the aforementioned ones

[4] Vocabulary sizes of 1K to 5K are tested and the best performing size is chosen.

TABLE 2. Hyperparameters values. For hyperparameters we highlight the optimal ones among all tried values.

Hyperparameter	Value(s)
$\lvert V \rvert$	1K, 2K 3K, 4K, **5K**
min_df	3, 4, **5**
max_df	0.9, **0.95**, 1
d_{memory}	32, **48**, 64, 128
$alpha$ (*Dirichlet prior*)	**0.1**, 0.2, 0.3
l_1	0.3, 0.5, 0.7
Number of Fully Connected Layers	16, **32**, 48, 64
Size of Fully Connected Layers	1,2,**3**,4
Number of Topics	20, 30, **40**
d_{GloVe}	50, 100
Batch Size	**16**, 32
Learning Rate	0.002, 0.001, **0.0005**
Number of Recurrent Layers	1,**2**,3

are chosen after using a coherence measure as well as the grid search. An example of the top five words present in one of the clusters is "checking", "balance", "account", "transaction" and "charge". This clusters topic is defined when a customer called to check their balance and/or look up a transaction/charge to their account. The topic modeling is trained only on the training set, but used to transform the transcripts in the training and testing set. The testing set topics are extracted using the global features tuned from the training set. The max sequence length chosen is 20. Prepadding is applied to all sequences less than 20 in order to make all sequences equal in length.

The deep recurrent neural network part of our model (i.e., recurrent memory cells) is implemented using TensorFlow (2.0) Eager [52], [53]. We use Adam optimizer and batch sizes of 16 and 32. We train our model for 50 epochs, and apply hyperbolic learning rate decay and early stopping strategies [54], [55].

One of the roadblocks of this research is that the dataset is highly imbalanced towards the *other* class. In order to assist the model in preventing from merely guessing the majority class every time, we attempt to balance our dataset by under-sampling frequent classes during the training session.

These sequences of call transcript data is split using an 80/20 scheme for training and testing purposes. Similarly, the training set is split to train (80%) and evaluation (20%) sets which is used to set the proper hyper-parameters and number of epochs. In the following subsection you will find the results from training our model and the evaluation results for the problem of implicit life event discovery.

C. EVALUATION METRICS

The variety of evaluation measures raises the question of how to understand and compare our model performance with other models. Evaluation metrics addresses this question by evaluating the quality of our model via comparing it against the other state-of-the-art text classification benchmark. However, choosing a proper criteria according to which our evaluation are going to be performed, is the most significant issue. As mentioned in section 3, here in this study we are dealing with a multi-label classification problem meaning that we can not exploit common criteria that is used in classical classification methods such as single-label multi-class problems, such as accuracy, precision, F-measure.

In evaluating a multi-label classification model, since the prediction could be a set of labels, therefore in some cases we may have an additional notion of being partially correct. For capturing this we have two different approaches. First one is called micro-averaging which is evaluating the average difference between the predicted labels and the actual labels globally for each test example, and then averaging over all examples in the test set. The second strategy is macro-averaging evaluation in which each label is evaluated separately then averaged over all the labels.

Evaluation measure based on micro-averaging are more influence by the performance of frequent classes, while conversely, those which are based on macro-averaging are affected by less frequent classes. Therefore, macro-averaging based measures are the key metrics in our presented case study. The followings are the equations for these evaluation metrics:

$$\text{Macro-P} = \frac{1}{N_c}\sum_{j=1}^{N_c}(j - \text{Macro-P} = \frac{\sum_{i=1}^{N} y_i^j \cdot \hat{y}_i^j}{\sum_{i=1}^{N} \hat{y}_i^j}) \tag{13}$$

$$\text{Macro-R} = \frac{1}{N_c}\sum_{j=1}^{N_c}(j - \text{Macro-R} = \frac{\sum_{i=1}^{N} y_i^j \cdot \hat{y}_i^j}{\sum_{i=1}^{N} y_i^j}) \tag{14}$$

$$\text{Macro-F1} = \frac{1}{N_c}\sum_{j=1}^{N_c}(j - \text{Macro-F1} = \frac{2\sum_{i=1}^{N} y_i^j \cdot \hat{y}_i^j}{\sum_{i=1}^{N} y_i^j + \sum_{i=1}^{N} \hat{y}_i^j}) \tag{15}$$

In these equations, i is indexed for call transcripts and j is the index for life event type, so N is the number of transcripts in the test set, and N_c is the number of classes of life events. y and $\hat{y}$ are both binary encoded variables to denote whether life event type j is in transcript i in ground-truth and prediction, respectively.

D. BENCHMARKS

As mentioned in section III, we formulate the problem of implicit life event detection as supervised text classification. Therefore, in order to evaluate the performance of our designed model, we leverage text classification benchmarks customized and fine-tuned on the provided large-scale dataset. We use the two following sets of text classification benchmarks:

1) SINGLE-DOCUMENT TEXT CLASSIFICATION

Text classification benchmarks are designed to classify single-document inputs into pre-defined classes [56], [57]. Initially, we apply text classification benchmarks on every single call transcript of a user prior to the report date in which

TABLE 3. Event detection results of our model on the large-scale dataset. As presented in the table, the model outperformed the benchmarks and achieves comparable performance to the state-of-the-art models in terms of precision and macro f1 score. The table is split into three parts, first part are the single document classification benchmarks while the second part includes the sequential ones. * The third section is dedicated to our model, Temporal Input Transformation Network (TITN).

Model	Macro-R	Macro-P	Macro-F1	Accuracy
All others	33.22	30.6	30.94	**97.66**
XGBoost + **TF-IDF** ($\|V\|$=3k) and **NMF** (30 topics)	**47.07**	35.45	41.11	93.49
Naive Bayes + **TF** ($\|V\|$=5k) and **LDA** (40 topics)	39.33	34.18	37.78	92.24
SVM + **TF-IDF** ($\|V\|$=1k) and **GloVe** (50 dimensional)	37.52	41.34	39.57	93.12
Ensemble of all above (majority vote)	**45.13**	42.91	43.94	94.07
SVM + **TF-IDF** ($\|V\|$=5k) and **GloVe** (200 dimensional)	**46.89**	39.60	43.55	92.79
Ensemble of all above (majority vote)	**48.22**	42.03	**45.67**	93.77
TITN* (**RNN** + **TF** ($\|V\|$=5k) and **LDA (40 topics)**)	**52.82**	**45.53**	**47.48**	**95.85**

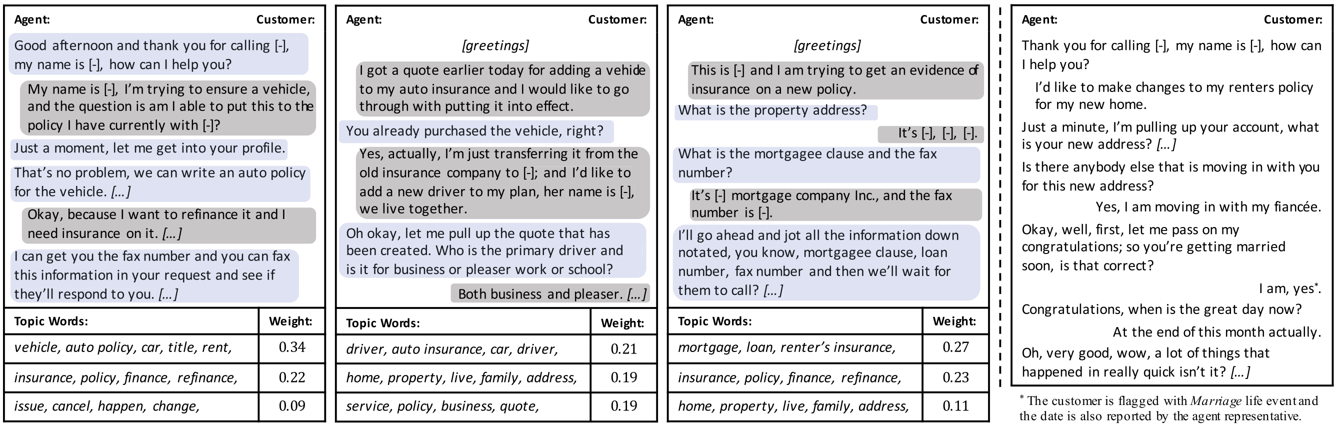

FIGURE 4. Sample call transcript sequence along with their topic distributions–stating the three most significant topics (those with highest weights in the utilized topic modeling scheme–and the last conversation that explicitly mentions the life event with which the customer is flagged. The transcripts yield to marriage life event which is flagged from the last conversation where it is explicitly mentioned by the customer. Our model correctly detect the life event of this sequence prior to the last conversation (explicit transcript), which denotes its ability of learning the implicit cues from the previous conversations.

the label of the transcript is the same as the implicit event existing in their life.

In this regard, bag-of-words (BoW) representation and topic modeling-based approaches are exploited for document representation and feature extraction similar to those of our methodology which yield state-of-the-art results for text classification. We remove the stop words and use TF/TF-IDF bag-of-word representation as well as the averaged word vector representation of transcripts using GLoVe [46], [58] trained on twitter for the document representation of transcripts, and LDA/NMF for topic modeling which are trained on the representations of the training set only to avoid data bleeding. As for the classification, we use Naive Bayes [57], SVM, XGBoost [59], and Logistic Regression which have shown promising results in the literature [60]–[62].

2) SEQUENTIAL DOCUMENT CLASSIFICATION

Regarding the second set of our benchmarks, we treat one's history of call transcripts sequentially. As mentioned in section III, a sequence of transcripts from a customer's previous calls can have jointly invaluable information regarding the implicit life events. As such, instead of feeding single call transcripts to the aforementioned text classification benchmarks one-by-one, we concatenate the call transcripts of a customer's history prior to feeding them to the document representation and topic modeling parts.

In addition, due to the imbalanced nature of the provided dataset, we use an additional baseline in which we label all testing examples with the same label of *other*. This is the majority class baseline which is a reasonable baseline for *Accuracy* and *macro-F1*.

It is important to note that we compare the best performing combinations of these benchmarks to our model. Likewise to our own methodology, sub-module selection and hyperparameter tuning is performed through a grid search algorithm on 20% of the training set.

E. RESULTS AND DISCUSSION

Once the model is designed experiments and the hyper-parameters are tuned, evaluation must be completed in order to test our model's performance. We present a case study on dataset provided to us by a large financial services company and the experiment and evaluation results can be found on Table 3.

Looking at the results, albeit the highly imbalanced distribution of call transcript sequences across different life events, especially class *other*, and the natural difficulty of the problem of implicit detection of life events, our model performance is significantly better than benchmarks and confirm our hypothesis which is the possibility of the detection of implicit life events based on a sequence of call transcripts from an individual customer's call history. Figure 4 shows sample call transcript sequences whose corresponding life event has been detected correctly by our model. Latent topic distribution for the three most important topics of every transcript is provided.

There are some setbacks and roadblocks in the experiment that prevent us from reaching higher results and an overall more accurate model. Data imbalance is a major problem in this situation. As mentioned before, only 5% of all data are tagged with the label *marriage*, *deployment*, *baby* or *retirement*. During training the model will be bias towards guessing the majority class because it can obtain a 95% accuracy score by doing so. By down-sampling the majority class, we are able to get the testing set to an accuracy of 74%. With more examples of the minority class, this model would be able to achieve higher accuracy.

Another roadblock is lack of data. The dataset for the experiment consists of a restricted time frame of transcript data points. This is the dataset approved by the companies lawyers and security teams. With more data and examples of more events, however, the models accuracy, precision, recall, and f score would also improve.

V. CONCLUDING REMARKS

In this project, we investigate an under-examined problem of life event detection from unstructured data of client-agent conversation transcripts. In particular, we bring attention to implicit life event detection from a customer's history of conversations, as a sequence of call transcripts, before the event has occurred or explicitly mentioned by the customer which enables prescriptive analytics for target marketing and call center analysts, in addition to lightening their load in manual analysis of such data. The overall problem is formulated as a supervised, multi-label text classification for which we present a novel methodology based on latent variable modeling (where the latent variables are main topics via which the transcripts can be represented) and deep recurrent neural networks to capture the temporal cues exist in a sequence of call transcripts transformed to their latent variables. As such, we refer to our methodology as temporal input transformation network.

In this regard, we exploit different latent topic modeling approaches in the recent literature to transform unstructured call transcript data into a more structured latent variables. Moreover, we customize an approach based on deep recurrent neural networks and end-to-end memory networks (utilizing recurrent memory cells) to capture the sequential clues from a sequence of call transcripts more effectively, and consequently, detect implicit life events more accurately. Using a real-life, large-scale dataset, collected at the call center of an American financial company, a case study is reported in which the performance of our model is compared to state-of-the-art text classification benchmarks, revealing the practical features and challenges. The experimental results demonstrate that our deep recurrent approach performs significantly better the benchmarks.

In our **future research**, we plan to apply more sophisticated variants of our novel deep learning model to this large-scale problem focusing on improving its ability in managing scalability as well as performance and interpretability which are important in giving more insight regarding the customers life events. Moreover, we would like to add contextual information of the customers, such as age, sex, location, time and date to name but a few, into our life event detection models.

REFERENCES

[1] L. R. Rabiner, "Applications of speech recognition in the area of telecommunications," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 1997, pp. 501–510.

[2] B. C. Herrin, M. S. Johnson, Jr., J. Stein, and X. Zhu, "Managing, monitoring and transcribing concurrent meetings and/or conference calls," U.S. Patent Appl. 10 334 110 B2, Jun. 25, 2019.

[3] B. Gawronski and B. K. Payne, *Handbook of Implicit Social Cognition: Measurement, Theory, and Applications*. New York, NY, USA: Guilford Press, 2011.

[4] C. Clavel, G. Adda, F. Cailliau, M. Garnier-Rizet, A. Cavet, G. Chapuis, S. Courcinous, C. Danesi, A.-L. Daquo, M. Deldossi, S. Guillemin-Lanne, M. Seizou, and P. Suignard, "Spontaneous speech and opinion detection: Mining call-centre transcripts," *Lang. Resour. Eval.*, vol. 47, no. 4, pp. 1089–1125, 2013.

[5] Y. Park and S. C. Gates, "Towards real-time measurement of customer satisfaction using automatically generated call transcripts," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1387–1396.

[6] O. D. Deshmukh, S. Joshi, S. Saurabh, and A. Verma, "Intent discovery in audio or text-based conversation," U.S. Patent 8 983 840 B2, Mar. 17, 2015.

[7] M. B. Burns and K. C. Moffitt, "Automated deception detection of 911 call transcripts," *Secur. Inform.*, vol. 3, no. 1, 2014, Art. no. 8.

[8] S. Ikbal and T. Faruquie, "HMM based event detection in audio conversation," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun./Apr. 2008, pp. 1497–1500.

[9] Y.-C. Chen, G. M. Lee, N. Duffield, L. Qiu, and J. Wang, "Event detection using customer care calls," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1690–1698.

[10] G. Sárosi, B. Tarján, T. Fegyó, and P. Mihajlik, "Automated transcription of conversational Call Center speech—With respect to non-verbal acoustic events," *Intell. Decis. Technol.*, vol. 8, no. 4, pp. 265–275, 2014.

[11] V. Shankar and E. C. Malthouse, "Moving interactive marketing forward," *J. Interact. Marketing*, vol. 20, no. 1, pp. 2–4, 2006.

[12] J. Ripa and R. Cheaib, "Real-time call center call monitoring and analysis," U.S. Patent 9 160 852 B2, Oct. 13, 2015.

[13] L. H. Hoyle, *Event Marketing*. Hoboken, NJ, USA: Wiley, 2002.

[14] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo, "Analyzing online discussion for marketing intelligence," in *Proc. Special Interest Tracks Posters 14th Int. Conf. World Wide Web*, 2005, pp. 1172–1173.

[15] E. Stepanov, B. Favre, F. Alam, S. Chowdhury, K. Singla, J. Trione, F. Béchet, and G. Riccardi, "Automatic summarization of call-center conversations," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2015, pp. 1–2.

[16] B. Lwowski, P. Rad, and K.-K. R. Choo, "Geospatial event detection by grouping emotion contagion in social media," *IEEE Trans. Big Data*, to be published.

[17] M. Osborne and M. Dredze, "Facebook, Twitter and Google plus for breaking news: Is there a winner?" in *Proc. 8th Int. AAAI Conf. Weblogs Social Media*, 2014, pp. 611–614.

[18] A. Ritter, Mausam, O. Etzioni, and S. Clark, "Open domain event extraction from Twitter," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1104–1112.
[19] H. Tanev, J. Piskorski, and M. Atkinson, "Real-time news event extraction for global crisis monitoring," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.* Springer, 2008, pp. 207–218.
[20] J. Li, A. Ritter, C. Cardie, and E. Hovy, "Major life event extraction from twitter based on congratulations/condolences speech acts," in *Proc. 2014 Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1997–2007.
[21] S. Liu, R. Cheng, X. Yu, and X. Cheng, "Exploiting contextual information via dynamic memory network for event detection," 2018, *arXiv:1810.03449*. [Online]. Available: https://arxiv.org/abs/1810.03449
[22] J. W. Orr, P. Tadepalli, and X. Fern, "Event detection with neural networks: A rigorous empirical evaluation," 2018, *arXiv:1808.08504*. [Online]. Available: https://arxiv.org/abs/1808.08504
[23] J. Zhong and W. Li, "Predicting customer churn in the telecommunication industry by analyzing phone call transcripts with convolutional neural networks," in *Proc. 3rd Int. Conf. Innov. Artif. Intell.*, 2019, pp. 55–59.
[24] A. Balahur and A. Montoyo, "Applying a culture dependent emotion triggers database for text valence and emotion classification," *Procesamiento Del Lenguaje Natural*, no. 40, pp. 107–114, 2008.
[25] A. Balahur and R. Steinberger, "Rethinking sentiment analysis in the news: From theory to practice and back," in *Proc. WOMSA*, vol. 9, 2009, pp. 1–12.
[26] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale multi-label text classification—Revisiting neural networks," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Springer, 2014, pp. 437–452.
[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
[28] R. Sahba, N. Ebadi, M. Jamshidi, and P. Rad, "Automatic text summarization using customizable fuzzy features and attention on the context and vocabulary," in *Proc. World Automat. Congr. (WAC)*, Jun. 2018, pp. 1–5.
[29] K. Kummamuru and D. S. Padmanabhan, "Method of analyzing conversational transcripts," U.S. Patent Appl. 11/614 200, Jun. 26, 2008.
[30] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato, "Phrase-based & neural unsupervised machine translation," 2018, *arXiv:1804.07755*. [Online]. Available: https://arxiv.org/abs/1804.07755
[31] J. Perkins, *Python Text Processing With NLTK 2.0 Cookbook: Over 80 Practical Recipes for Using Python's NLTK Suite of Libraries to Maximize Your Natural Language Processing Capabilities*. Birmingham, U.K.: Packt, 2010.
[32] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Interpreting TF-IDF term weights as making relevance decisions," *ACM Trans. Inf. Syst.*, vol. 26, no. 3, 2008, Art. no. 13.
[33] A. Aizawa, "An information-theoretic perspective of TF–IDF measures," *Inf. Process. Manage.*, vol. 39, no. 1, pp. 45–65, 2003.
[34] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," 2017, *arXiv:1711.00117*. [Online]. Available: https://arxiv.org/abs/1711.00117
[35] R. Vedantam, K. Desai, S. Lee, M. Rohrbach, D. Batra, and D. Parikh, "Probabilistic neural-symbolic models for interpretable visual question answering," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6428–6437.
[36] T. Shen, M. Ott, M. Auli, and M. Ranzato, "Mixture models for diverse machine translation: Tricks of the trade," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5719–5728.
[37] C.-J. Simon-Gabriel, Y. Ollivier, L. Bottou, B. Schölkopf, and D. Lopez-Paz, "First-order adversarial vulnerability of neural networks and input dimension," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5809–5817.
[38] D. M. Blei, "Build, compute, critique, repeat: Data analysis with latent variable models," *Annu. Rev. Statist. Appl.*, vol. 1, pp. 203–232, Jan. 2014.
[39] G. E. P. Box, "Sampling and Bayes' inference in scientific modelling and robustness," *J. Roy. Stat. Soc., A (Gen.)*, vol. 143, no. 4, pp. 383–404, 1980.
[40] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for latent Dirichlet allocation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 856–864.
[41] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[42] S. Sra and I. S. Dhillon, "Generalized nonnegative matrix approximations with Bregman divergences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 283–290.
[43] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.
[44] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, 2015, pp. 399–408.
[45] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 100–108.
[46] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[47] A. Karpathy, "The unreasonable effectiveness of recurrent neural networks," *Andrej Karpathy Blog*, vol. 21, May 2015.
[48] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen, *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis*. Springer, 2017.
[49] J. Weston, S. Chopra, and A. Bordes, "Memory networks," 2014, *arXiv:1410.3916*. [Online]. Available: https://arxiv.org/abs/1410.3916
[50] S. Sukhbaatar, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2440–2448.
[51] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, "A simple but tough-to-beat baseline for the fake news challenge stance detection task," 2017, *arXiv:1707.03264*. [Online]. Available: https://arxiv.org/abs/1707.03264
[52] A. Agrawal, A. N. Modi, A. Passos, A. Lavoie, A. Agarwal, A. Shankar, I. Ganichev, J. Levenberg, M. Hong, R. Monga, and S. Cai, "TensorFlow eager: A multi-stage, python-embedded DSL for machine learning," 2019, *arXiv:1903.01855*. [Online]. Available: https://arxiv.org/abs/1903.01855
[53] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and S. Ghemawat, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: https://arxiv.org/abs/1603.04467
[54] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," *Construct. Approx.*, vol. 26, no. 2, pp. 289–315, 2007.
[55] Q. Jin, B. Dhingra, W. Cohen, and X. Lu, "Attentionmesh: Simple, effective and interpretable automatic mesh indexer," in *Proc. 6th BioASQ Workshop Challenge Large-Scale Biomed. Semantic Indexing Question Answering*, 2018, pp. 47–56.
[56] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.zip: Compressing text classification models," 2016, *arXiv:1612.03651*. [Online]. Available: https://arxiv.org/abs/1612.03651
[57] B. Tang, H. He, P. M. Baggenstoss, and S. Kay, "A Bayesian classification approach using class-specific features for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1602–1606, Jun. 2016.
[58] D. Mrowca, E. Wang, and A. Kosson, "Stance detection for fake news identification," Tech. Rep., 2017.
[59] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.
[60] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explor. Newslett.*, vol. 19, no. 1, pp. 22–36, 2017.
[61] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, "Text mining using non-negative matrix factorizations," in *Proc. 2004 SIAM Int. Conf. Data Mining*, 2004, pp. 452–456.
[62] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Inf. Process. Manage.*, vol. 42, no. 2, pp. 373–386, 2006.

NIMA EBADI received the B.Sc. degree from the Electrical Engineering Department, Sharif University of Technology, in 2014. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, The University of Texas at San Antonio (UTSA). He is also a Research Fellow with the Secure AI and Autonomy Lab, UTSA. His research interests include machine learning, deep learning, probabilistic graph modeling, natural language processing and understanding, information retrieval systems, and conversation as a platform.

BRANDON LWOWSKI received the B.Sc. degree in mathematics and the B.Sc. degree in computer science from The University of Texas at San Antonio (UTSA), USA. He is currently with the UTSA AI and Autonomy Lab, UTSA. His researches are focused on social media data mining, data analysis, and topic modeling.

MEHRDAD JALOLI received the B.Sc. degree from the Electrical Engineering Department, IKIU, in 2012, and the M.Sc. degree from the Department of Biomedical Engineering, SRBIAU, in 2018. He is currently pursuing the Ph.D. degree with the Electrical Engineering Department, The University of Texas at San Antonio (UTSA). He is also a Graduate Research Assistant with the Secure AI and Autonomy Lab, UTSA. His research interests include machine learning, deep learning, and the application of AI in healthcare and medical dataset.

PAUL RAD received the B.S. degree in computer engineering from the Sharif University of Technology, in 1994, the first master's degree in artificial intelligence from Tehran Polytechnic, and the second master's degree in computer science from The University of Texas at San Antonio *(Magna Cum Laude)*, in December 1999, and the Ph.D. degree in electrical and computer engineering on cyber analytics from The University of Texas at San Antonio. He is a Co-founder and the Associate Director of the Open Cloud Institute (OCI), and an Associate Professor with the Department of Information Systems and Cyber Security. He is a member of OpenStack Foundation, the open source cloud software, San Antonio Tech Bloc Founding, and a Board Member of Childrens Hospital of San Antonio Foundation. He was a recipient of the Most Outstanding Graduate Student in the College of Engineering, Achieving Rackspace Innovation Mentor Program Award for establishing Rackspace patent community board structure and mentoring employees, in 2012, the Achieving Dell Corporation Company Excellence (ACE) Award in Austin for exceptional performance and innovative product research and development contributions, in 2007, and the Dell Inventor Milestone Award, Top 3 Dell Inventor of the year, in 2005. He holds 15 U.S. patents on cyber infrastructure, cloud computing, and big data analytics with over 300 product citations by top fortune 500 leading technology companies, such as Amazon, Microsoft, IBM, Cisco, Amazon Technologies, HP, and VMware. He has advised over 200 companies on cloud computing and data analytics with over 50 keynote presentations. He serves on the advisory board for several startups, high performance Cloud Group Chair at the Cloud Advisory Council (CAC).