Spring 5-28-2019

A Machine Learning Approach to Predicting Alcohol Consumption in Adolescents From Historical Text Messaging Data

Adrienne Bergh
Chapman University, bergh105@mail.chapman.edu

A Machine Learning Approach to Predicting Alcohol Consumption in Adolescents From Historical Text Messaging Data

A Thesis by

Adrienne Melissa Martin Bergh

Chapman University

Orange, CA

Schmid College of Science and Technology Submitted

in partial satisfaction of the requirements for the

degree of

Master of Science in Computational and Data Sciences

May 2019

Committee in charge:

Erik J. Linstead, Ph.D., Chair

Elizabeth Stevens, Ph.D.

Samuel E. Ehrenreich, Ph.D.

The thesis of Adrienne Melissa Martin Bergh is approved.

Erik Linstead. Ph.D.. Chair

Elizabeth Stevens. Ph.D.

Samuel E. Ehrenreich. Ph.D.

May 2019

A Machine Learning Approach to Predicting Alcohol Consumption in Adolescents From Historical Text Messaging Data

by Adrienne Melissa Martin Bergh

ACKNOWLEDGMENTS

Many many thanks to my family and friends for their constant support and encouragement throughout this process, and in everything I do. I'd like to thank Dr. Erik Linstead especially for believing in my potential and always giving me the tools to recognize it, Dr. Elizabeth Stevens for embodying everything it means to be the best role model for me, and Abby Atchison, Julie Hoag, and Chelsea Parlett-Pelleriti for being excellent and inspiring colleagues and friends.

I would also like to send my heartfelt thanks to Dr. Marion K. Underwood and the BlackBerry Project, specifically Dr. Samuel E. Ehrenreich for his continued support in interpreting this data, the collection of which would not have been possible without grants from the National Institutes of Health (#R01 HD060995).

VITA

Adrienne Melissa Martin Bergh

EDUCATION

Master of Science in Computational and Data Sciences **2019**

Chapman University *Orange, CA*

Bachelor of Science in Computer Science **2017**

Chapman University *Orange, CA*

ACADEMIC AWARDS

Outstanding Senior Award, B.S. in Computer Science, May 2017

RESEARCH EXPERIENCE

Undergraduate Research Assistant **2014–2017**

Chapman University *Orange, California*

Graduate Research Assistant **2017–2019**

Chapman University *Orange, California*

TEACHING EXPERIENCE

Graduate Teaching Assistant **2017–2019**

Chapman University *Orange, CA*

WORK EXPERIENCE

Machine Learning Intern **June 2018 – January 2019**

The Aerospace Corporation *El Segundo, CA*

Software Engineering Intern **June – July 2017**

Thuuz, Inc. *Palo Alto, CA*

Software Engineering Intern **May – August 2016**

Badger Meter Silicon Valley Innovation Center *Los Gatos, CA*

Software Engineering Intern **June – August 2015**

SunSpec Alliance *San Jose, CA*

SOFTWARE

Python Java C++ R MySQL

LIST OF PUBLICATIONS

Adrienne Melissa Martin Bergh

Arbuckle, C., Greenberg, M., Bergh, A., German, R., Sirago, N., Linstead, E. **T-Time: A data repository of T cell and calcium release-activated calcium channel activation imagery.** **August 2017**

BMC Research Notes, 10, 408

Ott, J., Atchison, A., Harnack, P., Bergh, A., Linstead, E. **A Deep Learning Approach to Identifying Source Code in Images and Video** **May 2018**

IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)

ABSTRACT

A Machine Learning Approach to Predicting Alcohol Consumption in Adolescents From Historical Text Messaging Data

by Adrienne Melissa Martin Bergh

Techniques based on artificial neural networks represent the current state-of-the-art in machine learning due to the availability of improved hardware and large data sets. Here we employ doc2vec, an unsupervised neural network, to capture the semantic content of text messages sent by adolescents during high school, and encode this semantic content as numeric vectors. These vectors effectively condense the text message data into highly leverageable inputs to a logistic regression classifier in a matter of hours, as compared to the tedious and often quite lengthy task of manually coding data. Using our machine learning approach, we are able to train a logistic regression model to predict adolescents' engagement in substance abuse during distinct life phases with accuracy ranging from 76.5% to 88.1%. We show the effects of grade level and text message aggregation strategy on the efficacy of document embedding generation with doc2vec. Additional examination of the vectorizations for specific terms extracted from the text message data adds quantitative depth to this analysis. We demonstrate the ability of the method used herein to overcome traditional natural language processing concerns related to unconventional orthography. These results suggest that the approach described in this thesis is a competitive and efficient alternative to existing methodologies for predicting substance abuse behaviors. This work reveals the potential for the application of machine learning-based manipulation of text messaging data to development of automatic intervention strategies against substance abuse and other adolescent challenges.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Chapter 1

Introduction

Text messaging is the modus operandi for communication amongst adolescents. As teenagers' social lives become increasingly digitized, it follows that the content of their text message conversations would reveal their habits and behaviors. According to a large-scale survey on teens and texting conducted by the Pew Internet and American Life Project, 88% of all adolescents who use cell phones engage in text messaging regularly [1]. Furthermore, this texting communication occurs at a staggering rate and volume. In a longitudinal study conducted between 2009 and 2013, a team of psychology researchers led by Dr. Marion Underwood provided a sample of 175 high school students with BlackBerry devices that were configured to capture all text messaging communication. Their results show that teens send an average of 55 text messages per day, and receive roughly the same amount [2]. The magnitude of their communication can, and frequently does, exceed these numbers during more typically "social" times, such as weekends and school-related events. A two-day sample of the text messaging gathered in the fall of 2009 by the BlackBerry Project during the students' Homecoming weekend included 43,305 text messages, for an average of approximately 127 messages sent and received per participant per day [2].

Due to the near ubiquitous engagement in text messaging by adolescents, as well as the uptick in communication patterns using this medium during typically social periods, it can be inferred that text messaging forms an integral part of teenagers' social lives. Particularly given the discreet nature of text messaging, and the privacy that possessing a mobile device lends an individual, an adolescent's texting correspondence provides a unique window into his or her habits concerning school, family, friends, and risky behaviors. Therefore, studying this digital communication provides researchers "a window into the secret world of adolescent peer culture" [3]. Indeed, the BlackBerry Project showed through their two-day sample that teenagers communicate openly and frankly over text message even when aware that the messages are being observed. Participants often used profanity and sexually explicit language and discussed substance abuse behaviors.

Previous studies on both text messaging and adolescent alcohol use have primarily been based on self-reported behavior, collected in the format of surveys and questionnaires, often administered in a school setting [1, 4, 5, 6, 7, 8]. Teenagers, however, are prone to misrepresenting themselves in such studies, whether due to erroneous recollection or self-preservation in the cases where they fear retribution [2]. Access to the content of adolescents' cellular correspondence allows for avoidance of this bias and provides a comprehensive look at their tendencies to engage in risky behaviors, which has been hitherto largely unexplored.

The raw text data captured by the BlackBerry Project represents a wealth of information prime for large-scale mining and machine learning efforts. In fact, the BlackBerry Project itself highlights the demand for automatic analysis of data of this nature, stating "The amount of data is so large as to be overwhelming; microcoding is under way but is highly labor-intensive" [2]. Development of a machine learning approach to processing this data will provide a means for efficient analysis and be instrumental in facilitating the examination of theoretically motivated research questions.

In recent years, machine learning approaches, driven by improved algorithms and cheap hardware, have become an increasingly viable option for analyzing large corpora of unstructured text across domains. Most recently, word embedding models fueled by advances in deep learning have yielded state-of-the-art results compared to algorithms based on statistical topic modeling. These neural-network-inspired word embedding models, particularly the skip-gram architecture with negative sampling (SGNS), outperform count-based distributional models on word similarity and analogy detection tasks [9]. word2vec, a popular implementation of the skip-gram method, has been shown to outperform count-vector-based distributional semantic approaches in synonym detection, semantic relatedness, concept categorization, and analogy recognition [10]. An extension of word2vec which allows embeddings to be learned on text excerpts (documents), doc2vec, also achieves state-of-the-art results on duplicate recognition and sentence similarity prediction [11]. Encouragingly, an analysis of SGNS indicates that the objective function and information available to the model is quite similar to those of more traditional methods [12], lending it mathematical credibility.

In this thesis, we present a deep learning approach to processing texting correspondence by adolescents for prediction of alcohol use. By employing the neural network based doc2vec algorithm, an implementation of the Paragraph Vector architecture introduced by Mikolov et al. in 2014, to develop vector embeddings for text communication, we architect a system in which microcoding is unnecessary to pursue interesting research questions [13]. The unsupervised nature of this algorithm and its ability to pick up on semantic similarity between words and variable length sections of text make it a good fit for our corpus. As such, we apply doc2vec to samples of text messaging from individual participants in the BlackBerry Project study, then leverage the resultant vectors as input to a standard logistic regression algorithm to predict whether the individual self-reported alcohol use during the time period over which the texts were collected. We achieve above 76.5% accuracy on this binary classification task without the need for curating domain-specific keyword lists or manually filtering individual samples, as are typical in studies of this nature [14, 15, 16].

The remainder of this thesis is organized as follows. Chapter 2 details the dataset used in this paper, including the method of collection and parsing the raw data into a database for machine learning tasks. Chapter 3 explains the methods used to process the data for classification, as well as our subsequent prediction tasks. It also includes a discussion of our efforts towards topic modeling on this dataset, and the shortcomings of the data that made this task impossible. Chapter 4 presents the results of our efforts, both in terms of the ability of the text-embedding model to pick up on semantic similarities within the text and the classification accuracy of our prediction models. In Chapter 5, we discuss the results of this paper in the general context of text analytics as well as the potential it reveals for influencing the way similar datasets are studied within the field of psychology. Chapter 6 describes previous work in the fields of alcohol risk analysis amongst adolescents and text analytics, specifically on short texts. Final conclusions are drawn in Chapter 7.

This thesis makes concrete contributions to the application of machine learning in the social sciences by providing an efficient and unsupervised method for transforming text to be used as input to other machine learning architectures. The deep-learning-based approach described in this paper is particularly useful when applied to text message data due to its robustness in deriving semantic information when faced with syntactical disorganization. Our work demonstrates doc2vec to be a good choice for natural language processing tasks in which the input data features a similarly fragmented and informal vocabulary.

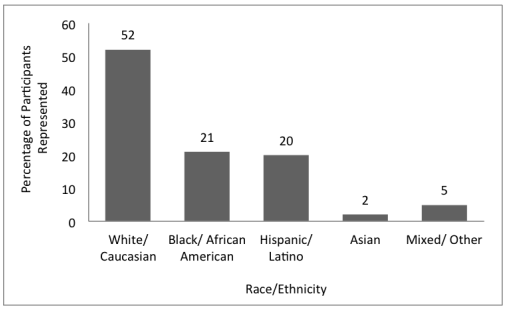
Chapter 2

Data

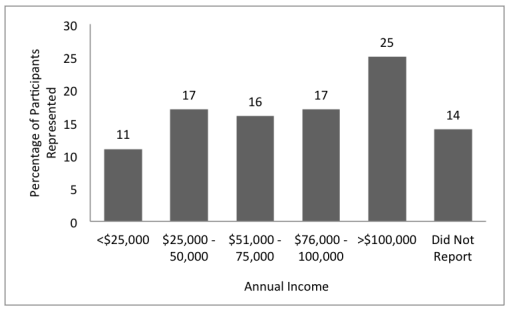
In this chapter, we describe the original study that facilitated the collection of the text messaging data used in this thesis, and our process to parse the data into a final dataset for machine learning. We then present the database structure which stores the raw texts along with pertinent information related to our research question, mainly participants' alcohol consumption patterns. We include summary statistics on the final dataset.

2.1 The BlackBerry Project

The text messaging data used in this thesis represents the efforts of a team of researchers led by Dr. Marion K. Underwood between the years of 2009 and 2013 [2]. The BlackBerry Project was a longitudinal study in which adolescents' digital interactions were directly observed via procurement of their text messaging communication. Originally recruited to a related study into the origins and outcomes of social aggression during third grade, the subjects were invited to participate in The BlackBerry Project prior to entering high school. A total sample of 214 students from suburban schools in the Southwestern United States par-



(a) Race/Ethnicity.



(b) Annual Household Income.

Figure 2.1: Participant Demographic Information by Percentage of Total Sample.

ticipated in the study, henceforth to be referred to as "participants". 50% of the participants were female. Additional demographic information, reported by parents upon enrollment in the study, is displayed in Figure 2.1.

As part of the study, researchers provided paid service plans for one BlackBerry device per participant, with unlimited texting and data. Each year, participants were provided with new models of their cell phones, with the intention that the devices remained compelling and easy to use. Participants were encouraged to use the BlackBerries as their primary communication devices, but were not restricted from other means of digital communication. However, questionnaires administered to participants throughout the process indicate strongly that students did conduct the majority of their communication on the devices. Moreover, analysis of the messaging content proves that the participants did not filter their conversations despite the knowledge that they were being observed [17]. Participants and their parents provided annual informed consent, and all data collection was approved by the Institutional Review Board.

The BlackBerry devices administered to the students were configured to capture all incoming and outgoing text message communication from the device, which was stored securely on BlackBerry Enterprise Servers (BESs), maintained by Ceryx and archived by Global Relay. These two companies together provided a daily digest of cell phone communication for each

participant, consisting of a detailed record of all text messages sent and received, labeled with date, time, and phone numbers of both the sender and the receiver. Messages containing photographs were not archived. Of some note, many of the messages appeared as the character "@" repeated many times. These messages are hypothesized by The BlackBerry Project team to represent messages deleted by the participant, and were thus removed from our final dataset due to their lack of meaningful content.

The data collection process also involved a series of annual visits to the participants' homes during the summer months, at which the participants completed questionnaires indicating their substance use habits during the previous school year. Specifically, students were asked to report whether they had consumed alcohol, tobacco, or marijuana. The final survey was administered online during the summer after the participants' first year out of high school. This thesis will focus specifically on participants' alcohol use during the later years of high school, self-reported during the summers after the 10th, 11th, and 12th grades as well as the summer following participants' first year out of high school. The specific time periods (TCs) represented are summarized in Table 2.1. Table 2.2 describes the number of participants for each TC who reported consuming alcohol, not consuming alcohol, or who failed to report on their alcohol usage. Of particular note is the 'No Answer Provided' column for TC13, which lists an inordinate amount of missing answers. As this TC represents the participants' first year out of high school, and the questionnaires were administered online instead of in the participants' homes, it is unsurprising that the response rate is lower. Despite this, our classification methodology, to be described in the following chapter, is not adversely impacted as we model each TC separately.

Table 2.1: Descriptions of TC time periods and associated academic years.

TC	Duration	School Year Represented
TC10	9/1/2009 – 8/31/2010	Sophomore
TC11	9/1/2010 – 8/31/2011	Junior
TC12	9/1/2011 – 8/31/2012	Senior
TC13	9/1/2012 – 8/31/2013	Year After Graduation

Table 2.2: Counts of self-reports of alcohol use during each TC.

	Alc-User (1)	Non-Alc-User (0)	No Answer Provided (N/A)
TC10	87	90	42
TC11	88	93	38
TC12	110	64	45
TC13	81	21	117

2.2 Database

In order to process the text messages efficiently at scale, we created a relational MySQL database to hold the raw texts, along with metadata. This metadata includes the sender and recipient of each text message and the time at which each message was recorded by the participant's device, the contact information for all unique users discovered within the data, and the substance use information for each participant during each time period. The schema of the resulting database is depicted in Figure 2.2.

The daily correspondence digests from Ceryx and Global Relay were originally provided in the format of eml files, a special BlackBerry messaging format. To extract the data from

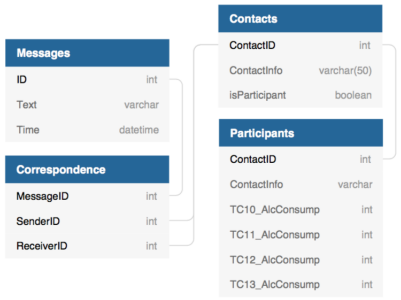


Figure 2.2: Schema for MySQL database storing text messaging and participant metadata.

these files, we first parsed the time, text, sender and recipient from each message into text log files using JavaMail[1], and then migrated the information from these files into the proper database tables. As specific messages were migrated, we created records for each new phone number encountered within the Contacts table, along with an indication of whether that user represented a participant in the study. The time and text of the message were entered into the Messages table, whereas the message ID and the IDs of the sender and recipient of the message were maintained in the Correspondence table. The reason for this separation is that occasionally, a message with multiple recipients was encountered. For group messages of this nature, multiple entries were made for that message in the Correspondence table, such that every record in that table represented a unique sender and recipient combination, but multiple records could exist for a single text message.

Once all raw text data had been parsed, substance use information was added to the participants' user data within the Participants table in the database by way of an excel file provided by The BlackBerry Project research team, which included categorical usage each year for each member of the study. Within the excel file, NULL values were sometimes recorded, indicating that the user had not provided an answer for that data point. Typically, if a null value was encountered, the responses from the user for all categories at that time point were null as well, corroborating the evidence that the null value indicated a skipped survey for that year by the user. To avoid clouding the results of our prediction task with inaccurate negatives, we left these fields within the database null as well. Thus, the 'AlcConsump' fields in the Participants table contain a 1 if the user reported alcohol use during that time period, a 0 if they did not report drinking, and a null value if the participant declined to provide an answer on the original survey. The source code for the data extraction and migration process can be found at https://github.com/adriennebergh/TMM.

[1]https://javaee.github.io/javamail/#API_Documentation

Our database schema allows for easy identification of not only the number of participants, but the total number of individuals involved in all communication throughout the duration of the study. Upon completion of the raw data migration, we estimate the following statistics: 27,746,591 total messages, 94,792 unique individuals, and 214 study participants.

Chapter 3

Methods

3.1 Topic Modeling

3.1.1 Latent Dirichlet Allocation

In the last ten years, with the introduction of Latent Dirichlet Allocation (LDA) [18], topic modeling has become the de-facto machine learning method for analyzing text. Applications range from bug detection and localization in software to semantic annotation of satellite images; strategic brand analysis via online commentary inspection to discovering functional miRNA regulatory modules with bioinformatics data [19, 20, 21, 22]. As such, our initial approach was to fit an LDA model to the corpus of text messages in the database to ascertain whether the algorithm could detect specific conversational topics among the participants. Our hope was that the model would highlight topics pertaining to drinking activities or other social clues that could aid in predicting alcohol use.

When applied to text, topic modeling with LDA treats each word in a corpus as a finite mixture over a set of underlying topics. Words, or tokens, in this sense are defined to be

members of the vocabulary of a corpus indexed by $\{1, ..., V\}$, and can refer to recognizable language or other discrete units of data. For example, for this dataset, we used the TweetTokenizer from the Natural Language Toolkit[1] in Python, which parses "emoticons" such as ":)" as words in addition to separating all punctuation into individual tokens. The colloquial nature of text messaging is often mimicked in tweeting, so the TweetTokenizer provided a good baseline for the discrete units of data we hoped to extract from our text.

LDA treats words as one-hot encoded vectors of length V. One-hot encoding represents a vocabulary as a binary vector, such that a word can be represented by a vector that is all zero values except its index within the vocabulary, which is marked with a 1. A document, W, is thus made up of a collection of these word vectors, constructed in bag-of-words fashion such that order does not play a role, and a corpus, D, is a collection of M documents. The number of topics represented in a corpus is taken to be a fixed number T, typically arrived at empirically using domain knowledge for a particular corpus.

LDA aims to generate a probabilistic model of a corpus, where latent "topics" are represented by distributions over the words found in the corpus and documents are further represented as random mixtures over these topics. It assumes that all documents are generated by randomly sampling from a Poisson distribution a number of words, n, to include in the document, then selecting the specific words from a multinomial probability, θ_d, conditioned on a randomly selected topic T. Topics within the model are selected based on a multinomial distribution with parameter Θ which is sampled from a Dirichlet distribution of k dimensionality, where k is known and fixed. The parameters for LDA are therefore given by two matrices: a $T \times D$ matrix $\Theta = (\theta_{td})$ of document-topic distributions, and a $W \times T$ matrix $\Phi = (\phi_{wt})$ of topic-word distributions. A fully Bayesian model is derived by taking symmetric Dirichlet priors with hyper-parameters α and β over the distributions θ_d and ϕ_t.

[1] https://www.nltk.org/api/nltk.tokenize.html

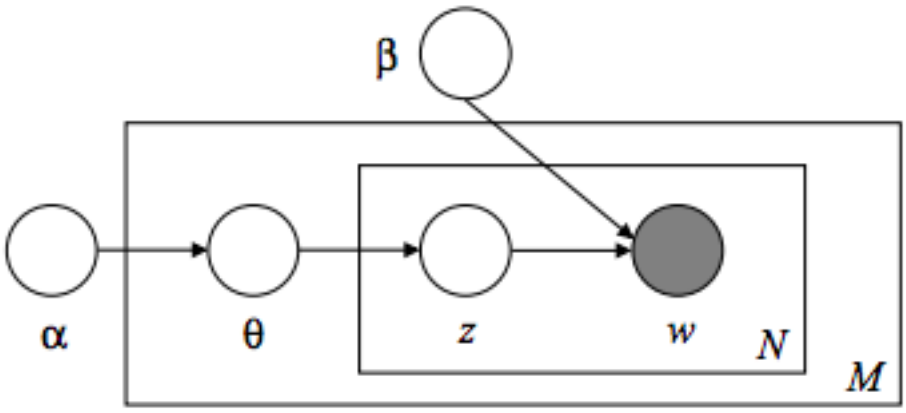


Figure 3.1: A graphical representation of the LDA model.

The prior probability for θ_d is given as

$$D_\alpha(\theta_d) = \frac{\Gamma(T\alpha)}{(\Gamma(\alpha))^T} \prod_{t=1}^{T} \theta_{td}^{\alpha-1}$$

with ϕ_t following the same distribution.

The probability of each document is found by integrating the likelihood over parameters ϕ and θ and their respective Dirichlet distributions. Θ and Φ are estimated by Maximum A Posteriori (MAP) or Mean Posterior Estimate (MPE) methods, and the posterior can be sampled using Markov Chain Monte Carlo Methods. A graphical representation of the LDA model is given in Figure 3.1, where the inner box represents the repeated procedure of sampling topics and then words for that topic for a document, and the outer box represents the selection of a Dirichlet variable for each document in the corpus.

Preliminary topic modeling efforts for this thesis were conducted on a subset of the data from TC10, using the popular topic modeling toolkit MALLET, which was written by Andrew McCallum [23]. For this effort, all texts sent by a given participant were aggregated into

```
0       0.0264  it's hahaha yeah hahahaha i'll you're love therealjordin yep don't work stephaniepratt people cuss gabe wut awh dunno party y'all
1       0.75346 love feel bad make talk time good told dont back things yeah stop guess thought care mad happy thing wanted
2       0.01429 haha reply i'm yeah it's don't that's hahaha trevor katie you're back status stroud_trevor facebook lol can't didn't what's erinn
3       0.02238 reply comment status text ignore unsubscribe facebook stop texts turn day guljeen summer kheleel barzani tonight love today hannah work
4       0.02324 haha i'm dnt yeah wat yur lol wit tht bro kno niqqa wnt cux babe ill qot love shit whyy
5       0.02992 lol lmao i'm don't aint babi damn that's babe yea bout shit knw ill tha love ass wanna real gotta
6       0.02239 lol i'm yea yep wat cuz awesome i'll hmmmm gunna that's good bree hey queen idk can't don't pretty kool
7       0.00952 wat kno yea dat aint i'm wit juss love bout dnt yep jus hey cuz true doin nun naw wht
8       0.01958 babi wat yea lyk babyee cuz dat luv nun mz.dimplez gud naw cum miss wats wit uld idk okayy didnt
9       0.01975 dont hey yeah cuz dang dude idk luv person ill cool buddy cake hahaha ruth wonderful loves nathan didnt complete
10      0.01989 mii yea jus dha dnt wah cuz man kno buh iim liik bro dat naw iit shyt wen wiid sum
11      0.03935 haha i'm yea don't tht ooh that's good gonna yeah i'll it's guess hehe idk awesome didn't yay yep omg
12      0.02584 hah yall eve christmas idk ill people it's yea game tonight party dont evan hahah wow sid night connor lauren
13      0.27311 i'm don't that's good didn't you're can't hey gonna i'll what's i've home fun tomorrow doesn't ill won't yay we're
14      0.06112 love baby i'm don't dont healy hun miss hope i'll feel happy didn't world babygirl great work talk call amazing
15      0.28036 haha yeah hahaha good idk nice it's hahah cool ohh fun guess kinda weird funny cute pretty wow hahahaha aww
16      0.03671 haha wat dat den yea gonna rlly hey dats dont man hav ill abt wit idk waz rite wen soo
17      0.03016 wut psroqstar aint king doin jus dont love shit yuh wen kno nun wit justin git dat damn put gon
18      0.35361 school hey good class day today work people year cool tomorrow home test friday lunch give find week time late
19      0.02689 i'm lol don't love gonna yeah ill dee that's idk wanna didn't fun what's can't kam shit haha tomorrow cool
20      0.00669 babe c.c.r&w.d.r i'm yea love chi lyke mama juz knw wat dat wife chi's est wifey diz ran wit dnt
21      0.01636 yeah ill joseph hannah audrey umm vasquez what's michael alexis chase bosch hehehe dylan god sucks abt sandra tomorow carranza
22      0.03155 yea hahaha hey i'll that's party lindsey game justin emily cool kevyn b/c yall people we're hahahaha girl it's he's
23      0.03247 baby bro hahaha don't alright i'm yeah nigga yhu dont ohhh love that's aint ass awww ohh damn christine naw
24      0.02893 yeah dont nigga bitch ass bro youngtune tyler love doin girl hahaha shut mama shit baby nun aint didt stupid
25      0.29439 i'm yeah don't gonna shit that's wanna fuck home cool didn't can't damn what's back hey text mom tomorrow good
26      0.03187 gonna dnt cuz ill bro baby bye yeah txt hey sofia.rangel alright didnt yall aww idk jst boo jamie bakk
27      0.49922 good time sleep fun movie pretty man wait dad day long watch night food bed eat nice car girl hair
28      0.00198 spam hey heyy spamspam what's yeah vote msg myles goin nathan btw gonna answer spamming facebook subj names woman wrote
29      0.01553 o'neal eva wat boo baby i'm note yeaaa yea doin nun bro jus rite real don't bout wit lol wen
30      0.02951 dnt dat wut wit yeah dha bitch qet hoe dhat knw wat qot isz ohh badest yew yeahh jux sum
31      0.02556 yeah hehe i'm sigh idk okayy ohhh gonna mmkay mhmm idkk what's good yeaa yupp that's yep talk ohh heyy
32      0.03108 haha i'm hav that's kno rly okayy dnt good srry wit hahah tho ill tis gonna hey bout nice idk
33      0.02737 i'm bout shit wit hahaha ill hve wat home cuz idk hit doin naw angela gona smoke lil ima jose
34      0.04303 babe haha i'm man alex ill gonna miss yeah idk good wanna awh promise awhh stuff work call hang today
35      0.02381 haha i'm lol lmao yea hey gunna soo babe wow hehe dont kus man good we're uhm don't ill didn't
36      0.01685 haha dat i'm juz wut lyk cuz ill wuz dats dude shit gonna bout den doin don don't hehe wucha
37      0.11239 baby dont sexy day ass dick fuck babe hell drive faster happy pussy back sex didnt wait hurt daddy shit
38      0.07507 hahah hahaha yeah hahahah cuz idk hahahaha hah that's you're gonna don't yeahhh love yeahh hahahahah ohhh wow hahahahaha funny
39      0.0814  youu ohh yeahh okayy whatt umm heyy yeahhh ohhh hahaha myy mee hahahah whyy hahah thatt cool wat andd yess
40      0.01178 babycakez wat boo i'm dnt jus yeaaa nun bout yea doin status wit wen rite bro love bae ummmmm commented
41      0.06409 status reply facebook commented text comment msg back wall subscribe wrote add poke post poked happy photo ashley birthday info
42      0.00007 ilocblob dsdb iloc bud
43      0.22535 wat dnt hey cuz idk wit wen wats send goin doin txt bout time dat pic fwd ima call talk
44      0.01644 bay dat kno i'm don't aint iight cuz dats iam wat den yuh mii wen bout gotta fina shit dnt
45      0.0307  bay kno lil_drama_moma dat yea wat jus doin ain cuz dnt hey love wit wut gne dats nun bae lik
46      0.46525 lol yea idk wanna hey doin day house good make home guess huh bad aww funny cuz wow yup bored
47      0.03589 pero okey para whit mami con por como tun ooh tho las bien gonna kno una c@ntifl@s co-exist yur nada
48      0.03359 youu babe yeah lol babee i'm lovee dont gonna yeahh love wait hey likee call aree mom that's text it's
49      0.03203 wht yeah i'm doin aint bytch tha lmao nigga chu yea bitch hay tht damn lol ass hell thas bby

[beta: 0.00794]
<1000> LL/token: -8.05637
```

Figure 3.2: Example topics generated by LDA.

a single training instance representing that participant's outgoing communication, and the model was asked to infer 50 topics using the instances from all participants. MALLET's built-in list of English stopwords was used to first filter the texts. Contents of this list are included with the MALLET distribution in the file stoplists/en.txt.

While the template-like quality of certain messages within the corpus (for example automatic notifications, such as Facebook status updates) provide an obvious pattern for topic modeling to extract meaning from, and "sexting" and school-related topics emerge, there was little other information that the model was able to discern. Figure 3.2 shows a sample output from this effort. Topics 2 and 41 appear to pertain to Facebook and other messaging activities, whereas topic 18 contains multiple words related to school, such as "class", "test", and "school", and topic 37 clearly pertains to sexting. The remaining 46 topics show little indication of coherent subject matter.

Based on the MALLET model's ability to extract at least these themes from the small sample of text given, we approached the issue of topic modeling the entire corpus of text messaging data from all four time periods captured, TC10 – TC 13. With these efforts, we used the LDA Multicore model implementation available through a popular machine learning library for Python, gensim [24]. LDAMulticore uses multiprocessing and streamed training for significant wall-clock speed ups over its predecessor, the LdaModel class from gensim[2]. For our model, we trained 40 topics using 40 workers for parallelization. As in our previous MALLET experiment, we filtered out English stopwords, this time using the list from NLTK [25], and again prepared input documents in the format of all communication sent by a particular user. However, following this, the model returned noisy results with no properly discernible topics. Working under the assumption that the documents were far too long to represent any latent topics accurately, we shortened input documents to instead include just the messages sent by each participant during a single given day, maintaining the same model hyper-parameters. Again, however, the results of the model were inconclusive. Next, we attempted to extend the list of stopwords to be filtered out of the texts before training. To do this, we calculated the most commonly occurring words in the data for each document using a computed TF-IDF score of that documents' particular vocabulary.

In TF-IDF scoring, term frequency within a document is scaled by the log of the ratio of the total documents to the number of documents containing the given term[3]. Words with very low TF-IDF scores could be assumed to be ubiquitous across documents, and thus good candidates for the stopword list. For example, we observed from our data that teenagers punctuate their messages with some variation of "haha". Laughter serves merely as a filler as opposed to conveying any information on the level of amusement felt by the involved party. "Haha" thus forms an ideal stopword. However, even with the removal of these additional

[2]https://radimrehurek.com/gensim/models/ldamulticore.html
[3]http://www.tfidf.com/

stopwords, we were unable to draw any meaningful topic analysis of the corpus based on LDA modeling.

We hypothesize that the failure of LDA to produce consequential results can be attributed to the fragmented and chaotic nature of the original data. It is very common for text messaging vernacular to incorporate slang and other colloquialisms that would cloud any typical text processing approach. In the absence of a very good spell-checker and slang-translator, which would require domain expertise and most likely would need to be customized for the project, it is an insurmountable challenge to process the text of this corpus for LDA modeling. The concept of laughter in text messages provides an example of the complexities of developing a reliable stopword list for a corpus of this nature. Even beyond the spelling variations found for "haha", i.e. "hahaha", "ahahaaha", etc., the acronyms "lol", "lmao", and others are often used for the same purpose. Adding these words, however, can remove meaning from certain text messages where they were in fact intended to convey mirth. It would almost require viewing texts on a case-by-case basis to determine which words were relevant to the true meaning, which negates the purpose of developing a machine learning pipeline to derive meaning from the data. However, pursuing a computational solution to these issues does pose an interesting avenue for future work.

3.2 Distributed Vector Representations for Input Text

A natural next step in light of the preprocessing issues discussed above is to pursue a solution that does not require such manual tactics to prepare the data. In particular, it is valuable to derive relationships between words based on semantic similarly. These relationships would allow for conclusions drawn for terms that appear frequently to be extended to infrequent terms in the corpus. In recent years, great strides have been made towards modeling just such connections in text by leveraging distributed representations of words in a vector space.

3.2.1 word2vec

The state-of-the-art algorithm for learning such embeddings for words is word2vec, developed by Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean in 2013 [26]. In this model, one-hot encoded vectors representing words sampled from a text excerpt are passed through a shallow neural network to predict neighboring words from the excerpt. As the input weight matrix of this network is updated through back-propagation to maximize the likelihood of predicting correct term co-occurrences, each column in the matrix comes to represent the "meaning" of one of the words in the vocabulary in the context of its surrounding words within the corpus. These weights can thus be used as words' distributed vector representations.

The word2vec algorithm can be configured to use either a Skip-Gram or Continuous-Bag-of-Words method, both of which consist of just an input layer, a projection layer, and an output layer (Figure 3.3). This allows high-dimensional vectors to be trained from very large data sets with great computational efficiency due to the lack of dense matrix multiplications. Indeed, it was the authors' explicit goal to develop techniques to be used for large-scale vocabularies on the order of millions of words.

In the Continuous-Bag-of-Words (C-BOW) model, many context words are taken as inputs to predict a single output word. This method is so named because the order of the words used to predict the output does not matter. In this framework, each word in the vocabulary, V, is represented by a unique column vector in a matrix W. At the input layer, N context words are selected from about a target word, and encoded in one-hot vectors, or vectors of size $V \times 1$, wherein each individual word in the vocabulary is assigned an index, and words are encoded by placing a 1 at their corresponding index and a 0 elsewhere. At the projection layer, the input layer is projected to a $N \times V$ space with a shared weight matrix. Finally, the target word is predicted with a log-linear classifier.

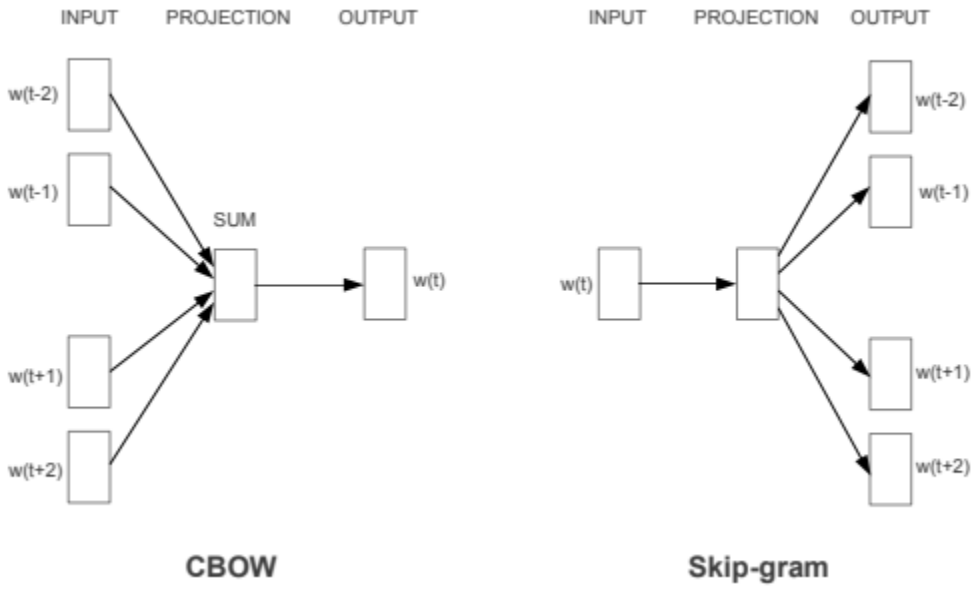


Figure 3.3: Word2Vec model architectures.

The objective for the C-BOW model is to maximize the log probability, given a sequence of training words $w_1, ...w_T$:

$$\frac{1}{T}\sum_{t=N}^{T-N} logp(w_t|w_{t-N}, w_{t-N+1}, ...w_{t-1}, w_{t+1}, w_{t+2}, ...w_{t+N})$$

where

$$p(w_t|w_{t-N}, w_{t-N+1}, ...w_{t-1}, w_{t+1}, w_{t+2}, ...w_{t+N}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

and y_i are the log-probabilities for each output word i:

$$y_i = b + Uh(w_{t-N}, ..., w_{t+N}; W)$$

h defines the method of combining the word vectors from the N context words, extracted from W as inputs to the model. Popular choices are concatenation or averaging. U and b are softmax parameters.

The C-BOW model is thus trained to predict a word w_t given a window of N words occurring before and after it. Upon completion of model training, the weights matrix can be broken into vectors representing each word. Since these weights have been trained to transform the one-hot encoded vectors for each word in the vocabulary such that they maximize the probability of their correct neighboring words, the weight vectors accurately represent the "context" of each word. The weights matrix encodes how each one hot vector needs to be altered to represent most completely its position in the original text, and therefore its distributed embedding amongst the other words. These vectors form the representations of the words in a corpus which are highly useful as inputs to other machine learning algorithms.

In the Continuous-Skip-Gram Model, conversely, a single input word is used to predict other words that have high probabilities of occurring in the context of the input (within N words before or after the input). Again, a continuous projection layer followed by a log-linear classifier is employed. To train the Skip-Gram model, given a sequence of training words $w_1, w_2, ..., w_T$ from a vocabulary of size V, and a training context N, the average log probability of each word in the context given an input word is maximized with the objective

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-N\leq j\leq N, j\neq 0} logp(w_{t+j}|w_t)$$

where $p(w_{t+j}|w_t)$ is given by the softmax function:

$$p(w_o|w_I) = \frac{exp({v'_{w_o}}^T v_{w_I})}{\sum_{w=1}^{V} exp({v'_w}^T v_{w_I}}$$

and v_w and v'_w are the input and output vector representations of a word w, respectively.

Several extensions to the original model were introduced in [9], which found that replacing the softmax function at the output layer with Negative Sampling and adding subsampling of frequent words vastly improved both training speed and accuracy of word representation.

Negative sampling is defined by the objective

$$log\sigma({v'_{w_o}}^T v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i} \sim P_n(w)[log\sigma(-{v'_{w_i}}^T v_{w_I}]$$

which replaces the softmax formula $P(w_o|w_t)$ in the original objective formula. Negative sampling represents a simplification of Noise Contrastive Estimation, by comparing a target outcome against k negative samples for each input data sample. A default value of 5 is used for k by the gensim implementation[4] we employ on our data, as well as by the original code distributed by Google. Subsampling of frequent words mimics the removal of stopwords during preprocessing; as the authors note, the vector representations of these very frequent words do not change significantly from epoch to epoch once presented with enough training examples (on the order of several million). To maintain the ranking of frequencies while aggressively subsampling words above a frequency threshold, Mikolov et al. discard words from the training set with probability

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

During training, words further from the input term are also sampled less due to their supposed decreased relevancy. The authors show that using the models outlined previously, analogous relationships can be drawn between vocabulary words using simple vector algebra. In their example, X = vector("biggest")vector("big") + vector("small") yields a vector X very similar to the trained embedding for "smallest", using cosine distance as a metric. Other relationships the algorithm can be shown to detect include Man-Woman, City-in-State, Singular-Plural, and Opposite.

With over 1.2 million distinct words in the vocabulary of our corpus, word2vec is the ideal solution for developing word representations for this thesis. With data of this scale, more

[4] https://radimrehurek.com/gensim/models/doc2vec.html

traditional approaches such as one-hot encoding are impractical, if not computationally prohibitive. Further, the capability of the model to derive semantic similarity and analogous grouping of vocabulary words offers a solution to the failures confronted when applying LDA to this data. The subsampling of frequent words during training makes it unnecessary to filter stopwords from the data, and the method of adjusting word vector representations based on surrounding context words results in similar embeddings for words used in similar contexts. Thus infrequent misspellings should closely resemble their correct counterparts. Ostensibly "hahahhahhh" will occur under the same circumstances as "hahaha", leading word2vec to assign very similar vectors to each.

3.2.2 doc2vec

Paragraph Vector, or doc2vec, as a popular implementation of it is known, is an extension of word2vec that trains a vector representing the full training instance, or document, along with the representations of the embedded words. This algorithm is especially powerful because it allows variable-length pieces of text to be represented as fixed-length vectors, a necessary condition for using these texts as input to most machine learning algorithms. The architecture of the doc2vec algorithm can again take on two forms, the Distributed Memory Model of Paragraph Vectors (PV-DM) or the Distributed Bag of Words Model of Paragraph Vectors (PV-DBOW). These architectures are represented in Figure 3.4. In the Paragraph Vector paradigm, every piece of text in the corpus is mapped to a unique vector in a matrix D. Thus for a corpus of p documents, $D = \{d_1, d_2, ..., d_p\}$.

For PV-DM, training occurs in much the same way as for the C-BOW Model for word2vec; multiple context words are passed as input to the model with the objective of predicting the next word given the probability formulas found above. In this case, however, a paragraph vector, d_a, is taken from D to represent the text that the words in question were drawn from.

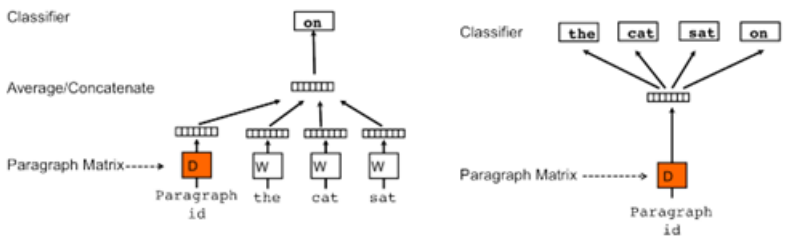


Figure 3.4: Doc2vec model architectures.

d_a is averaged or concatenated with the word vectors as input to the model. Thus, as the vectors are updated during training, the paragraph vector comes to represent a collective "memory" of the states of all of it's words' vectors. Importantly, the word vector matrix is shared across all paragraphs, but the paragraph matrix is shared only across contexts generated from the same paragraph. Hence, trained word vectors represent the context in which words are seen across the entire corpus, but paragraph vectors represent specific instances of the words and the co-occurrences of words within each paragraph.

During training, the only change that need be made to the C-BOW framework is to extend the input combination method, h, to concatenate or average the document vector with the context word vectors as input. The model is thus trained by maximizing the average log probability

$$\frac{1}{T}\sum_{t=N}^{T-N}\log p(w_t|w_{t-N},...,w_{t+N},d_a)$$

to predict word w_t given context $w_{t-N},...,w_{t+N}$, where all words have been drawn from document d_a. The word vectors and paragraph vectors are trained using stochastic gradient descent and back-propagation. To compute a vector for a new paragraph, a new vector d_a is added to the paragraph matrix D, and the word vectors and model weights are fixed while gradient descent is performed to adjust only the paragraph matrix.

The second format for the Paragraph Vector model, PV-DBOW, mimics the Skip-Gram model for word2vec. A single paragraph vector is taken as input, and the model is asked to predict several words as being randomly sampled from that paragraph. The model is thus given a single paragraph vector d_a from D as input and is trained to predict several vectors w from W. During gradient descent, a word is sampled from an excerpt of text from one of the paragraphs. Then, given the paragraph it was taken from, the word is classified as one of the vectors in W. Since this format focuses on classifying words, the vectors for the individual words need not be stored during training, making this method less computationally intensive than the last. The authors note that PV-DM used alone tends to perform well, but a concatenation of the vectors learned by each method leads to the best results, with an error rate of 7.42% on a benchmark data set for sentiment analysis, and just 3% on a custom information-retrieval task [13].

An important aspect to the Paragraph Vector model is that it is an unsupervised learning algorithm. In unsupervised learning, algorithms use co-occurrences of input data, rather than labels, to draw conclusions about the statistical structure implicit in the input space [27]. doc2vec is unsupervised in that it does not require any explicit target outputs or labels to be associated with input documents. Instead, it draws from sliding windows of word contexts occurring within the input texts to make its predictions.

As mentioned previously, word2vec and doc2vec were developed with the intention of training on very large datasets. Removing the restriction of needing labeled data makes the use of these algorithms practical for cases where labeled data is scarce but unlabeled data is readily available. The paragraph vector algorithm can be trained on a total dataset of which a subset is labeled, then just the vectors for which labels are available can be passed to a classifier for a prediction task [13]. The unsupervised nature of the training process allows researchers to capitalize on as much related data as they have available to them, while maintaining only labeled data for classification models, which may require fewer training samples for good

performance. The applicability of this facet of the algorithm is discussed in the following section.

3.2.3 Compiling Training Documents for doc2vec

The mechanism of learning distributed vector representations for variable-length texts provides our project with a unique opportunity to generate representations for communication based on the similarities and nonlinearities between the messages themselves. Rather than needing to query for only communication explicitly related to drinking activities, we can include all communication as input to the doc2vec algorithm. As word vector representations are trained along with the doc2vec model, we capitalize on an efficient training mechanism that yields multi-faceted data that can be used in several tasks.

In total, we experimented with four strategies for compiling documents to transform with doc2vec. First, we concatenated all sent messages by a given user during a specific TC into a single document, such that each document in the training set represented the outgoing communication from one participant in the study during one school year. Next, we tried a more granular approach by sampling a participant's correspondence on a weekly scale, in this case selecting the messages both sent and received by the given user during the given time period. This way, many training instances were created for each user during each school year. A week-long time period was selected so that both weeknights, where students can be assumed to take part in more studious or extra-curricular endeavors, and weekends, which tend to be more social, were captured in the communication data. In an effort to extend the applicability of this work to analyzing any text messaging data regardless of chronology, we created input documents of 50 randomly selected messages from participants' correspondence. All messages sent by a given user during a given TC were queried, and these messages were shuffled and partitioned into documents of 50 messages

each. A document was created for the remainder of the texts after partitioning evenly, the total process resulting again in many documents per participant per TC. For example, if a participant sent a total of 20,075 text messages during TC10, there would be 402 documents generated for the user, 401 documents of 50 messages, and 1 document with the remaining 25 messages. By restricting documents to only outgoing messages from a participant, we constructed inputs to our classifier where we could be certain that messages from a teenager's peers, which may not have reflected the activities or attitudes of the participant himself or herself, would not infiltrate the data. Finally, our last method for generating input documents resembled the previous one greatly, but did not carry the restriction of including only sent messages. Rather, we captured all outgoing and incoming communication from the participant's BlackBerry device. This allowed for a more comprehensive picture of the user's correspondence in each document.

For each of the document compilation methods described above, the next step was to "tokenize" each document, or split it into a list of its component parts, such as words and emoticons, again using the TweetTokenizer. Crucially, we did not perform any further preprocessing on the data before passing it as input to the doc2vec model. One of the main advantages to doc2vec for our corpus is its ability to distinguish meaning for all tokens based on context. It was therefore unnecessary to filter out misspellings, stopwords, or slang terms, as the model would learn to associate these terms with their proper English counterparts.

As documents were parsed, we assigned each a label based on the stored answers to the substance abuse survey for each participant. If the participant had reported consuming alcohol during the TC during which the specific communication was recorded on their device, the document was labeled with a 1. If they reported not drinking alcohol, the document was labeled as a negative sample, with a 0. It is important to note that not all participants gave answers to the health survey at every TC. For each TC, there are a handful of students who did not respond. For the doc2vec training stage, this could be ignored, as the

Table 3.1: Total dataset sizes per TC per document generation mechanism.

	TC10	TC11	TC12	TC13
All Messages	59,270	147,405	169,824	37,269
Sent Only	33,113	83,198	104,462	23,120
Weekly	2,805	6,903	7,267	1,184

Table 3.2: Labeled dataset sizes per TC per document generation mechanism.

ALL MESSAGES		
	Alc-Users	Non-Alc-Users
TC10	34,813	24,457
TC11	81,099	66,306
TC12	111,377	58,447
TC13	23,717	13,552
SENT ONLY		
TC10	19,532	13,581
TC11	46,418	36,780
TC12	68,862	35,600
TC13	14,842	8,278
WEEKLY MESSAGES		
TC10	1,430	1,375
TC11	3,341	3,562
TC12	4,401	2,866
TC13	918	266

algorithm is unsupervised. However, for the classification stage described in the following section, these participants' documents are omitted from the testing dataset for those TCs during which they did not report on alcohol consumption. The data preparation efforts detailed above resulted in four distinct training sets for each of the four document compilation methodologies. Preliminary testing revealed that the documents containing all sent messages during a TC resulted in inferior classification performance, and so we leave these experiments out of this thesis. The remaining three methodologies are retained, resulting in 12 sub-datasets, described in Table 3.1. After labeling the data, the total dataset was reduced to the distribution described in Table 3.2.

3.2.4 Training doc2vec Model

To implement the Paragraph Vector model for our data, we used the Doc2Vec implementation from the popular gensim library [24]. We selected the PV-DM implementation of the model based on remarks by the original authors that it tends to work well alone for most tasks. We also induced the model to train word-vectors in Skip-Gram fashion simultaneously to training paragraph vectors, in order to leverage the distributed representations of the words within the texting documents for future experimentation. We initialized the model to learn vectors of size 200, using a window size of 8. The models were trained for 40 epochs, based on the results reported by [11]. We maintained this architecture across all training datasets for comparability between methods of document compilation. For each training set, a new model was initialized and trained, then used to fit vector representations for each of the labeled documents for classification. Each of our doc2vec models were trained in under 3 hours, representing a marked decrease in the time required to process this data in the original study [2].

Each doc2vec model was trained on the full text message corpus to allow the algorithm to capture the texting vernacular on a greater scale and the word representations that inform the paragraph embeddings to be universal inter-TC. The actual messaging content may vary year to year as colloquialisms evolve, but the contextual basis for the doc2vec training algorithm makes our methodology robust to such changes. If a new slang term replaces an old favorite utterance, the word2vec process underlying doc2vec will learn to associate the same context, and thus similar vector representations, with both words.

Results for the weekly method of developing communication embeddings, as well as the methods of sampling 50 random sent messages and 50 random messages from the total communication of the participant, are reported in Chapter 4.

3.3 Classification

The focus of this thesis is building "communication profiles" for participants by learning vector representations of their text messaging communication during a given school year using doc2vec. We apply logistic regression with these communication profiles as input to predict adolescent's self-reported alcohol use. This breaks down into four specific sub-tasks: predicting self-reported alcohol use during TC10, TC11, TC12, and TC13. One particular participant often gave different answers about drinking habits year to year, or even failed to provide a response during one or more years, so the classification cannot be performed in aggregate. Rather, the logistic regression algorithm is provided with only the messaging content from a specific time period for which relevant alcohol use data is available. As mentioned previously, the data is incomplete for certain participants during each TC. The doc2vec instances generated from communication by a given participant are omitted from a TC's dataset if that participant failed to report on their alcohol consumption activities during that TC.

3.3.1 Logistic Regression

Logistic regression is an optimal method for regression analysis of dichotomous variables, as it transforms the output to a scale of (0,1), where outcomes above 0.5 are assigned to one classification and less than 0.5 to the other. Logistic regression fits a sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ to the input data to determine a binary classification, essentially formulating the problem as the posterior probability of a positive outcome. If $\sigma(\text{f(x)}) > 0.5$, the "positive" class is selected; otherwise, the "negative" class is assigned. Given f(x) as a simple weighted regression $f(x) = \sum_i w_i x_i + b$, the posterior probability of the outcome y is defined as

$$p(y|x; w) = (\sigma(W^T x))^y (1 - \sigma(W^T x))^{(1-y)}$$

with likelihood

$$p(y|x;w) \sim \prod_{i}^{N} (\sigma(W^T x_i))^{y_i} (1 - \sigma(W^T x_i))^{(y_i - 1)}$$

Substituting $f(x)$ for $W^T x_i$ and applying the logistic function σ, the negative log likelihood becomes

$$L(W) = \sum_{i}^{N} log(1 + e^{-y_i f(x_i)})$$

Learning is performed to update the weights w_i by minimizing the negative log likelihood. Thus, the optimization problem is formatted as

$$\min_{w \in \mathbb{R}^d} \sum_{i}^{N} log(1 + e^{-y_i f(x_i)}) + \lambda ||w||^2$$

were $\lambda ||w||^2$ is the L2 regularization term.

For this thesis, we employed the scikit-learn implementation of Logistic Regression[5], using the stochastic average gradient (SAG) algorithm for optimization and the L2 norm as the penalty, as is necessitated by SAG [28]. SAG is a method of optimizing a finite sum of smooth convex functions whose iteration cost is independent of the number of terms in the sum. Thus, this method is an efficient optimization algorithm for logistic regression when applied to very large datasets. SAG incorporates a memory of the previous gradient values, and thus achieves a faster convergence rate than other stochastic gradient methods, by applying the following iteration function

$$x^{k+1} = x^k - \frac{\alpha_k}{n} \sum_{i=1}^{n} y_i^k$$

where at each iteration a random index i_k is selected and y_i^k is set as

[5]https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

$$y_i^k = \begin{cases} f_i'(x^l) & i = i_k \\ y_i^{k-1} & otherwise \end{cases}$$

Here we train multiple instances of a logistic regression classifier, initialized to use a SAG optimization algorithm and L2-norm penalty, to complete each of the prediction tasks detailed above. The output in each case is a binary classification, with 1 indicating that the student whose communication is portrayed by the input reported drinking during the TC in which the communication occurred, and 0 meaning that the participant indicated they did not drink during the TC that the messages were captured.

3.3.2 Training Considerations

As is evident from Table 2.2, there is a significant class imbalance represented in the total dataset. Particularly in TC13, 79.4% of the dataset represents the positive class, or the Alcohol Consumers. This poses a risk to our classifier in that high accuracy could be reached just by consistently predicting the majority class. To mitigate the negative effects of an unbalanced dataset, we downsample the majority class to match the size of the minority group. In experiments with cross-validation, downsampling led to more consistent results across all folds than training and testing on the entire dataset.

To verify the validity of our models, we train using ten-fold cross validation. The process of K-Fold cross validation consists of partitioning the entire dataset into K folds, and training K models, each time holding out one of the K partitions as a testing set and training on the remaining $K-1$ groups of data. Thus for 10-Fold cross validation, the entire dataset for each TC for each specific input document type is partitioned into ten parts, and ten separate

classifiers are trained, each time leaving one partition out as testing data. At the start of each fold, a new instance of the classifier is initialized before fitting with the training data for that fold. Results of these classification methods are described in the following chapter.

Chapter 4

Results

Here we detail the results of applying logistic regression to the vector representations learned by the doc2vec algorithm for the previously described correspondence documents. We additionally highlight the ability of the doc2vec algorithm to concurrently train a word2vec model capable of embedding the vocabulary of the communication corpus such that known topics of conversation exist in relative proximity to one another within the vector space. We demonstrate the robustness of the word2vec algorithm against spelling mistakes and slang terms by showing that words and their common misspellings group together in this vector space.

4.1 Word2Vec Experiments

Visualization of the embeddings of the texting vernacular reveals the ability of the word2vec algorithm to learn similar vector representations for similar words in this corpus. Figure 4.1 plots selected words from the vocabulary along with the 30 most similar terms for each word

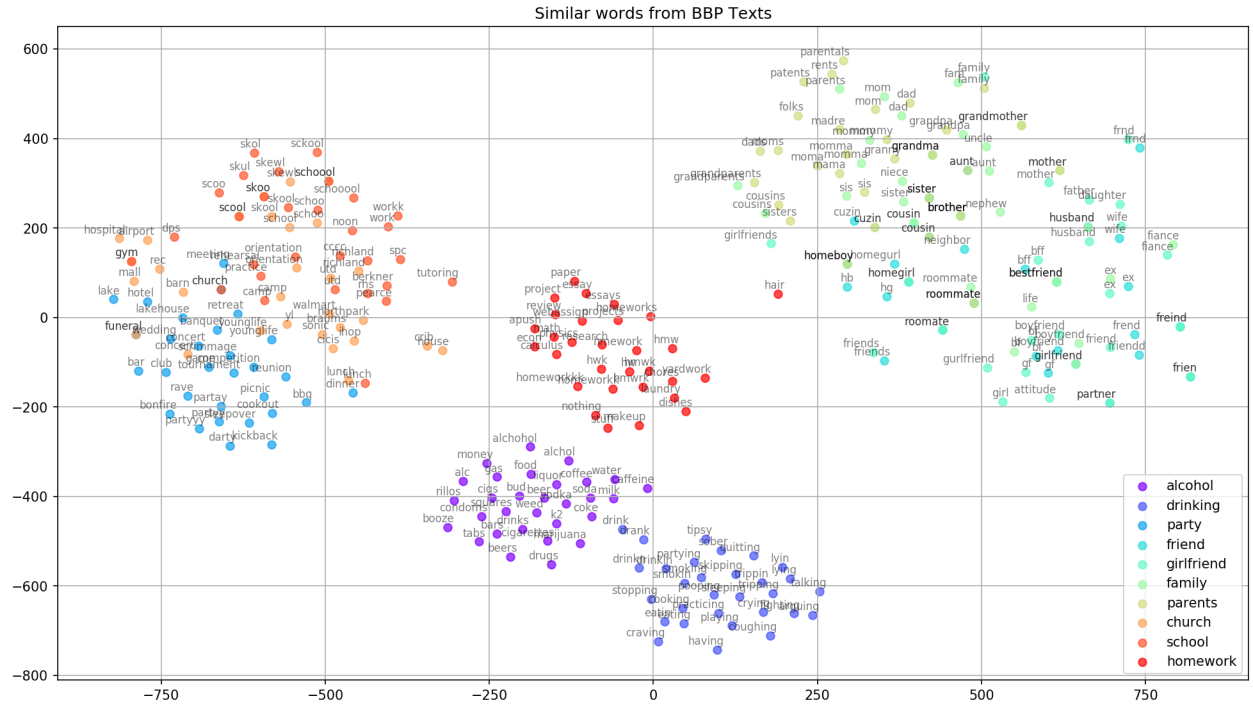


Figure 4.1: Embeddings for a selection of words representing common conversation topics. Word vectorizations were learned by a word2vec model trained on documents of 50 randomly selected sent and received messages.

based on cosine similarity. Given two word vectors A and B, cosine similarity is defined as

$$similarity = cos(\theta) = \frac{A \cdot B}{\| A \| \| B \|}$$

Output values range from -1, meaning exactly opposite, to 1, meaning exactly the same, with 0 indicating orthogonality. The 30 terms that appear in the same color as each of the target words listed in the legend are the words within the vocabulary which yield the highest cosine similarities to the target words.

The plot was created using the t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm, which was introduced by Laurens van der Maaten and Geoffrey Hinton in 2008 as a non-linear dimensionality reduction technique [29]. The algorithm embeds high-dimensional data into two dimensions for visualization such that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability. It should

be noted that this algorithm has been shown to be highly susceptible to its hyper parameters, especially the "perplexity" value, a smooth measure of the effective number of neighbors anticipated within the data. As such, it can reveal clusters which do not exist [30]. However, we conducted multiple experiments with a range of perplexity values for our data and did not observe a notable change in the clusters formed. Figure 4.1 was created using a perplexity of 30, given that we plotted 30 similar words for each input term.

As is evidenced by Figure 4.1, the word clouds generated by "family", "parents", "friend", and "girlfriend" all cluster closely together, with little distinguishability between the distinct clouds for each term. This suggests a strong association between words relating to interpersonal relationships. The word clouds for "drinking" and "alcohol" also appear close to one another in the vector space, although here the two clouds do show linear separability. This can be attributed to the differences in part of speech for the two terms. The intrinsic meaning behind "drinking" is similar to that of "alcohol", but one is a verb while the other is a noun. Indeed, in inspecting the similar terms selected from the vocabulary for each word, it is clear that other actions, such as "craving" and "smoking" are grouped with "drinking" whereas other substances, such as "coffee" and "drugs" are grouped with "alcohol". These clusters reveal not only the word2vec algorithm's tendency to group similar topics together, in this case illicit behaviors and paraphernalia, but to highlight the context in which the terms are used.

The latter is also clearly shown by the intermingling of the vector representations for "church", "school", and "party" and their respective most similar words. This reveals a correlation between those topics in the context of the participants' conversations, and can be taken to indicate that all play roles in the students' social lives. This theory is compelling especially in light of the inclusion of terms such as "younglife", a popular Christian youth organization, within the most similar clouds for both "church" and "party". Given that each of "church", "school" and "party" are places where people gather it is not surprising

that the algorithm would group them. Imagine the training instances "Are you going to church?", "When are you going to the party?", "I have to go to school later." In each of these sentences, "church", "school", and "party" are interchangeable. This is exactly the type of contextual relationship that the word2vec algorithm's learning is based on. It is encouraging to our research question to note that while there is significant intermingling between the most similar words to "church" and "school", the word cloud for "party" remains relatively distinct from the others, though directly neighboring. Examination of the most similar terms within the cloud reveals that "kickback", "darty", "rave", and "sleepover" are all included. These are all venues at which substance abuse is known to occur in high school, showing that the algorithm has learned to associate words linked with risky behaviors.

For more fine-grained analysis, a selection of words from the communication corpus along with five of the most similar words to each selection is displayed in Table 4.1. These lists clearly show the word2vec model's ability to derive similar vector representations for similar words, whether syntactically or semantically. The list of similar words for "mom" versus "dad" even shows that the model has deciphered a difference between how adolescents talk about their female versus male relatives. The most similar words for "test" are the different forms of assessments often used at school, and the most similar words for "alcohol" are other substances typically consumed at parties. Meanwhile, the similar terms to "relationship" reveal the vocabulary typically associated with relationships in high school; for example "situation", a word frequently used before a pair is comfortable with defining their relationship status, appears. Table 4.2, on the other hand, displays the word2vec model's ability to group misspellings or alternative spellings to commonly used words together. This table verifies the legitimacy of word2vec as a means for building keyword lists for filtering texts based on topic, as discussed in Chapter 6.

The actual cosine similarity value returned for each pair of words reveals how closely coupled those terms are within the corpus, or how interchangeable the words are in the context of

Original Word	Most Similar Word	Cosine Similarity Metric
“mom”	“dad”	0.969
	“sister”	0.901
	“momma”	0.901
	“mommy”	0.900
	“grandma”	0.899
“dad”	“mom”	0.969
	“brother”	0.902
	“grandma”	0.888
	“grandpa”	0.882
	“momma”	0.882
“relationship”	“friendship”	0.766
	“realtionship”	0.756
	“situation”	0.752
	“conversation”	0.712
	“group”	0.697
“fight”	“fite”	0.7990
	“argue”	0.603
	“argument”	0.591
	“lie”	0.565
	“arguement”	0.563
“alcohol”	“weed”	0.768
	“beer”	0.690
	“alc”	0.688
	“drinks”	0.638
	“booze”	0.637
“test”	“quiz”	0.897
	“exam”	0.881
	“midterm”	0.776
	“tests”	0.691
	“exams”	0.666

Table 4.1: Word embedding similarities based on word2vec model trained on documents of 50 randomly selected sent and received messages.

Original Word	Most Similar Word	Cosine Similarity Metric
"haha"	"hahaha"	0.875
	"haha"	0.871
	"hahah"	0.804
	"hah"	0.787
	"hahahaha"	0.697
"school"	"skool"	0.937
	"skoo"	0.882
	"schoo"	0.871
	"church"	0.835
	"scool"	0.827
"fuck"	"fuk"	0.734
	"fuckk"	0.720
	"fukk"	0.712
	"eff"	0.702
	"fck"	0.658
"love"	"miss"	0.844
	"wuv"	0.700
	"lovee"	0.688
	"lov"	0.652
	"luvv"	0.649
"hey"	"heyy"	0.832
	"heyyy"	0.757
	"aye"	0.752
	"ay"	0.687
	"ayy"	0.675

Table 4.2: Word embedding similarities highlighting misspelling/ alternative spellings.

the adolescents' text messages. While a term may be one of the most similar words to a given target word, the actual vectors for the two words may not resemble one another substantially. For example, "mom" and "dad" have a cosine similarity of 0.969; their vector representations are nearly identical, indicating that adolescents use those words in the same places within their texts. However, the vectors for "relationship" and "arguement" [*sic*] only have a cosine similarity of 0.563. The vectors for those two terms were pulled in different directions by a more varied range of training examples, but are still the most similar to one another given the full scope of the entire vocabulary.

	Weekly		All		Sent Only	
Original Word	Sim Word	Cos Sim	Sim Word	Cos Sim	Sim Word	Cos Sim
"homework"	"math"	0.72	"hw"	0.89	"hw"	0.92
	"physics"	0.70	"hmwk"	0.84	"hmwk"	0.79
	"finish"	0.69	"hmwrk"	0.80	"chores"	0.76
	"study"	0.62	"project"	0.63	"hmwrk"	0.72
"relationship"	"understand"	0.69	"friendship"	0.77	"situation"	0.75
	"feelings"	0.68	"realtionship"	0.76	"friendship"	0.74
	"honest"	0.66	"conversation"	0.71	"realtionship"	0.74
	"trust"	0.65	"argument"	0.67	"conversation"	0.69
"party"	"invited"	0.75	"partyy"	0.81	"partyy"	0.76
	"house"	0.63	"concert"	0.77	"concert"	0.76
	"fun"	0.63	"cookout"	0.76	"wedding"	0.67
	"tonight"	0.62	"rave"	0.70	"sleepover"	0.65

Table 4.3: Model-specific embedding similarities highlighting the differences between models trained on different document types.

An interesting outcome to explore is the difference between the similarities of the learned vectors based on the type of document each model saw as input to training. While there is a significant amount of overlap, Table 4.3 displays a selection of terms for which the five most similar terms did not agree. From these results, it can be determined that training on randomly selected messages ("All" messages or "Sent Only" messages) leads to very similar vector representations. The models are not presented with cohesive conversations where one message would relate to the next with high probability, but rather are given documents where the only applicable context is the current sentence. These two models learn to associate synonymous words. Examining the term sets for the "Weekly" model, however, reveals that when provided with all messages sent and received during one week, the model is able to develop similarity between more conceptual, descriptive words. For example, "relationship" shows high similarity to "understand", "feelings", "honest", and "trust", all things that are known to be important components of a relationship. It can be inferred that the conversations that adolescents have about relationships involve discussion of these other elements.

Given that the document vectors are trained along with the word vectors in the doc2vec algorithm, they can be expected to carry the same contextual information. As the weights for the word vectors are updated, so too are the weights for the paragraphs from which they originated. Thus, the context that effects the words' representations would influence the document vectors in like manner. As the word vectors have been shown above to indicate conceptual and semantic significance between terms in the vocabulary, the document vector representations can be anticipated to incorporate these things at the document level as well.

4.2 Alcohol Consumption Prediction Experiments

Tables 4.4, 4.5, and 4.6 show mean accuracy results for the classifiers trained on, respectively, 50 randomly selected messages sent and received by a participant, 50 randomly selected messages sent by a participant only, or all weekly messages sent and received by a participant, for each TC. Precision and recall metrics are included along with accuracy information to judge each model's ability to completely discern true predictions from false.

In any binary classification task, one potential outcome can be assigned a value of "True" while the other is represented as "False". Then, a True Positive (TP) outcome occurs if the model predicts True and the truth value is True, and a True Negative (TN) arises if the model predicts False and the truth value is False. A False Positive (FP) indicates the model predicting True but the real outcome being False, and a False Negative (FN) is the inverse, where the model predicts False but the truth value is True. Precision is a metric which indicates the amount of times a model correctly predicts a positive outcome ("True"), and is defined as

$$precision = \frac{TP}{TP + FP}$$

	Mean Accuracy	Precision	Recall
TC10	0.765	0.767	0.761
TC11	0.793	0.797	0.786
TC12	0.791	0.788	0.797
TC13	0.811	0.813	0.811

Table 4.4: Mean accuracies with precision and recall scores for 10-fold cross-validation experiments using logistic regression. Inputs were doc2vec vectors learned from documents generated from 50 randomly selected sent and received messages.

Recall is a measure of how many of the true positive outcomes were indicated by the model, and is defined as

$$recall = \frac{TP}{TP + FN}$$

Together, precision and recall can be used to determine the proportion of correct predictions a model makes given the actual number of true outcomes in the data and given the amount of true outcomes the model predicts. They are particularly useful metrics to leverage on an unbalanced testing set, where simply predicting one class each time could lead to a high standard accuracy.

Among the three types of training documents, the models trained on adolescents' weekly correspondence outperform those for the 50 randomly selected texts, whether sent only or sent and received. The highest reported mean accuracy for the weekly model was 88.1% and the lowest was 80.2%, in contrast to maximum mean accuracies of 81.1% and 84.0% and minimum mean accuracies of 76.5% and 80.1% for the sent and received model and sent only model, respectively. Given that the doc2vec model is trained in sliding window fashion, where context plays a key role in the training input pairs the algorithm is provided, documents composed of chronologically recorded messages should provide more cohesive training samples than random ones. The context of aggregated conversations helps the doc2vec model learn more comprehensive representations for documents, leading to better input vectors for the logistic regression classifier.

	Mean Accuracy	Precision	Recall
TC10	0.801	0.803	0.796
TC11	0.806	0.811	0.802
TC12	0.814	0.813	0.814
TC13	0.840	0.842	0.836

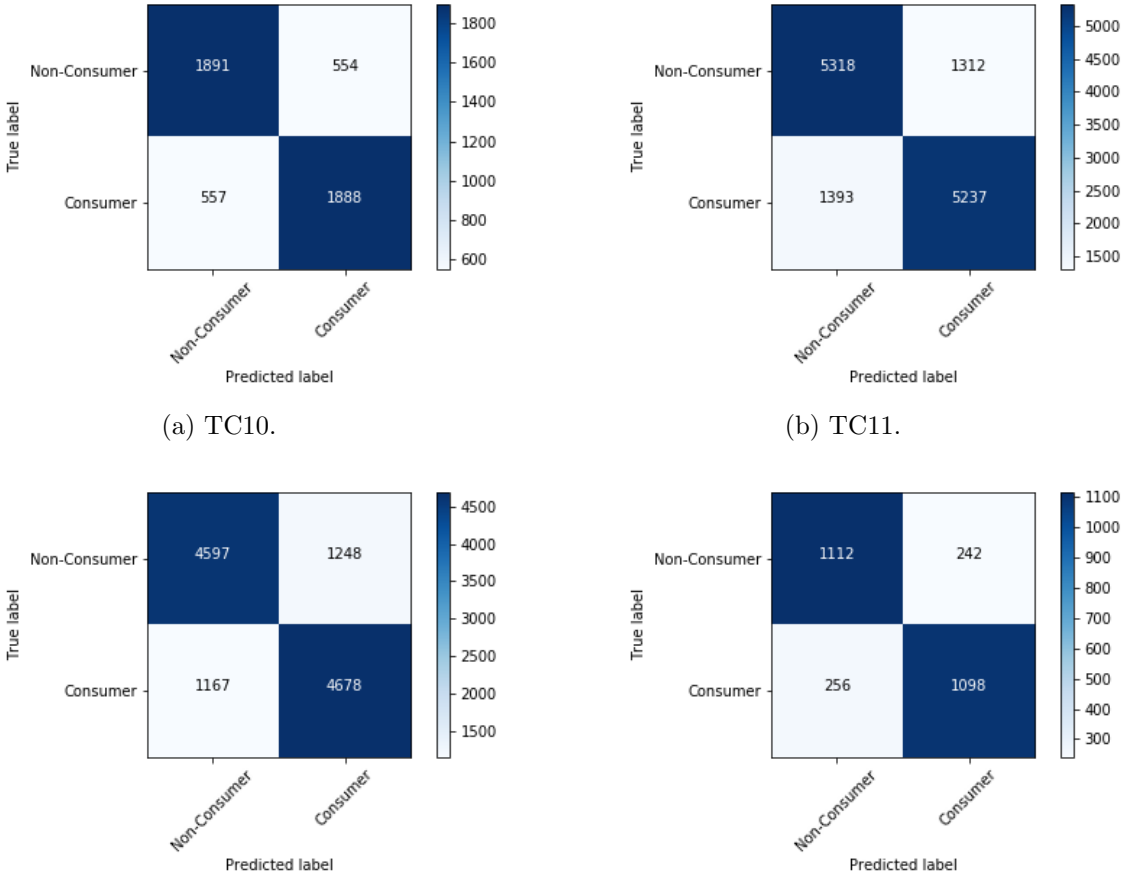
Table 4.5: Mean accuracies with precision and recall scores for 10-fold cross-validation experiments using logistic regression. Inputs were doc2vec vectors learned from documents generated from 50 randomly selected sent messages only.

	Mean Accuracy	Precision	Recall
TC10	0.802	0.809	0.791
TC11	0.826	0.825	0.811
TC12	0.824	0.832	0.813
TC13	0.881	0.901	0.861

Table 4.6: Mean accuracies with precision and recall scores for 10-fold cross-validation experiments using logistic regression. Inputs were doc2vec vectors learned from documents generated from weekly sent and received messages.

When the three types of models are compared based on the TC the data was selected from, another interesting pattern emerges. The highest accuracies for each training document type are reported for the models trained on data from TC13. The weekly model achieved 88.1% accuracy, the sent and received messages model achieved 81.1%, and the sent only model achieved 84.0% during TC13. This TC represents the year after high school for each of the participants. Therefore, many of the consistencies in the lifestyles of all participants can be assumed not to hold for TC13. Whereas during TCs 10, 11, and 12, a fair amount of the adolescents' conversation presumably circled around the families they lived with, the schools they attended every day, and their high school extra curricular activities, in TC13, these commonalities may not exist between the subjects. Thus, conversations carry less information consistent across all samples in TC13 and the classifier can be expected to perform better.

For all training document types, the precision and recall values for each classification task are high and correspond closely with the accuracies reported, indicating high fidelity in predictions by our model. The confusion matrices for each of the TCS are displayed in Figure



(a) TC10.

(b) TC11.

(c) TC12.

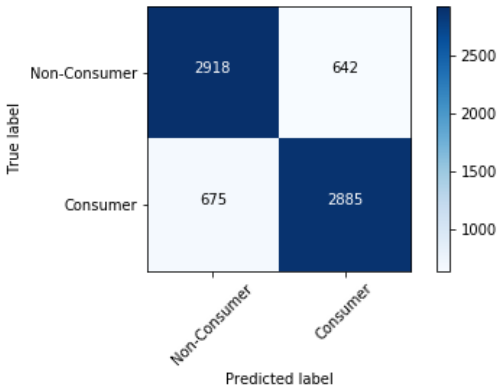
(d) TC13.

Figure 4.2: Confusion matrices for the task of predicting self-reported alcohol use during each TC based on documents constructed from all texting communication, with 50 random messages sampled.

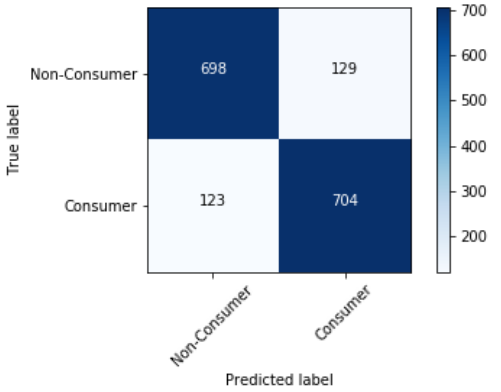
4.2 for documents comprised of 50 sent and received messages, Figure 4.3 for documents of 50 sent messages only, and Figure 4.4 for documents of weekly correspondence. As is evidenced by the cases in which the model predicted incorrectly, represented by the upper right and lower left quadrants of the confusion matrices, none of the models are focusing all of their predictions on one classification. Because the number of predictions made for each "Consumer" and "Non-Consumer" are very similar across all TCs, we can be confident that our models are not overfitting to our training data.

(a) TC10.

(b) TC11.

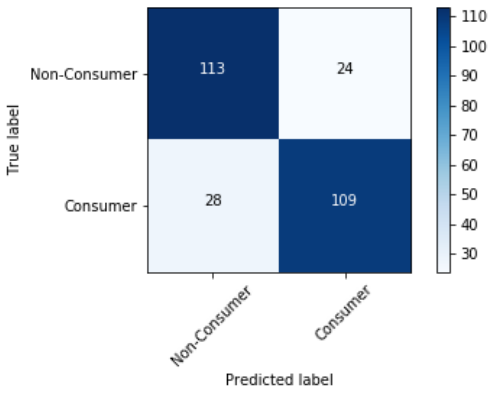


(c) TC12.

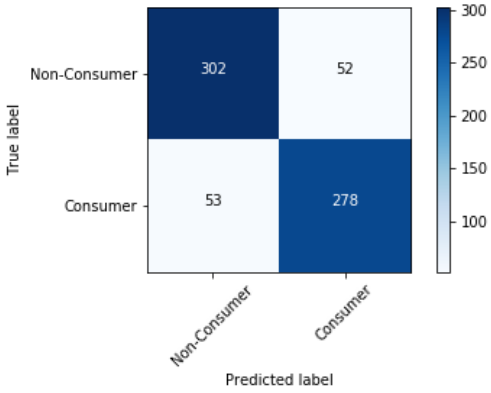


(d) TC13.

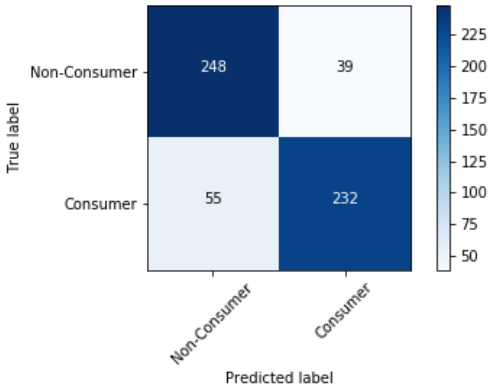
Figure 4.3: Confusion matrices for the task of predicting self-reported alcohol use during each TC based on documents constructed from only sent messages, with 50 random messages sampled.



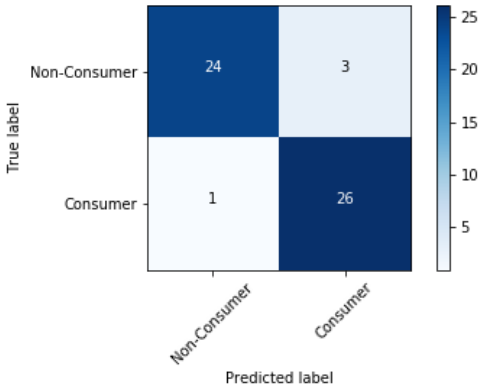
(a) TC10.



(b) TC11.



(c) TC12.



(d) TC13.

Figure 4.4: Confusion matrices for the task of predicting self-reported alcohol use during each TC based on documents constructed from weekly messages.

Chapter 5

Significance and Future Work

This thesis augments existing research efforts focused on predicting risky behaviors based on text messaging data. Its contributions are two-fold. First, it develops and applies machine learning techniques to text messaging data in order to develop a semantic model of content which can be fed to a classifier for behavior prediction. Secondly, by leveraging these techniques, we are able to reduce the time required to complete such analyses by several orders of magnitude compared to the manual approaches employed to date. Based on these contributions, we discuss the applicability of the methods detailed herein towards SMS-based intervention strategies for adolescent alcohol consumption.

5.1 Reduction of Necessary Analysis Resources

5.1.1 Leveraging doc2vec Embeddings to Subsample Data

The doc2vec model outlined in this work can be trained in a matter of hours to produce reliable inputs for prediction tasks related to the behavioral patterns of participants. Given

that manually coding the text messaging data took the original research group multiple years, utilization of this machine learning method could at the very least supplement existing methods to reach results more quickly. One such application could be a pre-screening process to produce candidates likely of engaging in risk behaviors. The messages of these specific individuals could then be manually examined to ascertain precise indicators of harmful habits. Researchers would thus be relieved of the onus of scrutinizing every text message in the corpus.

The unsupervised nature of doc2vec provides another compelling rationale for this approach to processing text message data. The algorithm engineers representations of the natural text without need for distinct labels or additional categorization. With a dataset of this volume and level of noise, it is paramount to the efficiency of investigation to be able to screen for importance and relevance of certain instances over others. The vector encodings described in this thesis provide interested researchers with a way to evaluate samples of an individual's communication for their connection to both other samples of text and the behaviors that participants reported on during the TCs, as we showed with our classification task. That the context in which certain words and topics are used is leveraged by the algorithm to develop the document vectors is reassuring of the trustworthiness of the resulting vectors' portrayal of the raw data. Manual coding of the data is not necessary, as the algorithm itself draws on underlying message structure and co-occurring topics of conversation to develop its outputs. Our experiments show that the vectors for the communication excerpts carry some signal regarding the habits of the involved participants. This opens up a wealth of potential for future classification tasks and concept models on the data without the need for laborsome preprocessing techniques. In particular, we hope to leverage the doc2vec embeddings developed in this word to predicting mental health issues among the participants, including anxiety and depression.

5.1.2 Augmenting Existing Text Classification Methodologies with word2vec Embeddings

As evidenced by the results presented in this thesis, the word2vec algorithm provides a machine learning approach to discovering similar words within a text corpus, even when the vocabulary is irregular and atypical. This outcome carries the potential for supplementing existing coding methodologies in a dataset-specific manner. The BlackBerry Project original study relied on the Linguistic Inquiry and Word Count (LIWC) [31] to categorize the topics of conversation among participants. LIWC is a computer program that references dictionaries of words that are known to fall under certain categories to read through text documents and calculate the percentages of the words in the documents that pertain to each category. Given the breadth of knowledge represented in its dictionaries, LIWC provides an extensive way to model the concepts and topics at play in a text. However, it does have the setback of being relatively static and cumbersome to update. Word candidates for each category must be manually selected, and as the original developers note, must ignore context. Their example is the word "mad", which is coded as an anger word irrespective of its common usage as a synonym for "crazy". Furthermore, the BlackBerry Project team had to pre-process the texting transcripts before inputting them to the LIWC software to create a more cohesive vocabulary to the one represented within LIWC. Any time a common abbreviation was used, the researchers replaced it with its full phrase, e.g. "laughing my ass off" being substituted for "lmao" [2].

The word2vec results presented in this paper reveal the embeddings learned by the algorithm as a logical next step to wholly representing the meaning intrinsic in a vocabulary. Supplementing the LIWC dictionaries with specific term sets compiled from the "most similar" words for each word within a given corpus would allow researchers to circumvent the shortcomings of the manually-compiled LIWC dictionaries and avoid the need for pre-processing the text. In fact, as referenced within the following chapter, many researchers have found

success in classifying texts as concerning a given topic by searching for the presence of keyword sets augmented with similar words found via word2vec representations. The word2vec model employed can be trained on either a researcher's specific corpus or an external, related one. This clearly broadens the scope of any text analytics task, as the success found from learning word2vec vectors on external corpora relieves a project of the limitations of training dataset size. Specifically for projects such as this one, where the vocabulary is widely varied and contains many misspellings and uncommon terms, word2vec represents a very promising methodology for expanding upon existing coding strategies.

5.2 Practical Applications for Screening and Intervention Strategies

Privacy is a key issue in the treatment of personal data such as text messaging correspondence. Every parent wants to make sure their child is conducting him or herself in a safe and appropriate manner online. However, it can be challenging to maintain surveillance over adolescents' communication with others while still preserving important boundaries and fostering a sense of independence. An automated intervention methodology based on screening incoming and outgoing communication from a teenager's mobile device through a method similar to the one proposed in this thesis could provide security to parents while avoiding invasion of privacy. This method could simply trigger intervention strategies when a teen's communication indicates substance abuse patterns or other risky behaviors, without parents directly monitoring their child's texts. Much in the same way that advertisements are targeted towards internet users based on their online activity, informative articles or warning messages could be sent directly to adolescents indicated by the tool as at risk for engagement in harmful activities.

There are two notable anticipated benefits from such a system. Firstly, active SMS intervention has been shown to reduce frequency and degree of binge drinking in young adults [32]. Establishing a program where this methodology would flag students for enrollment in a texting intervention program could provide a mechanism for mitigating binge drinking in high schools. Secondly, this provides an opportunity to improve health literacy. Approximately 80 million Americans have low health literacy, and as a consequence experience more hospitalizations and greater reliance on emergency care, among other health risks [33]. An intervention strategy based on simply providing pertinent statistics and other important health-related information to adolescents likely of engaging in risky behaviors could prepare younger generations to become health-conscious adults.

Moreover, due to the fact that the doc2vec algorithm abstracts actual SMS messaging content into vectors, this system would avoid storing private data which could be vulnerable to exploitation. Rather, representations of that data which retain important contextual information would be used. This approach could provide a means of monitoring and triggering intervention techniques without direct observation by automatically processing texts.

Chapter 6

Related Work

6.1 Analysis of Alcohol Use Patterns

Analysis of adolescents' online activities provides unique insight into their social lives and habits, including behaviors associated with health risks such as substance use, particularly due to the near ubiquitous use of social media and cellular devices by teenagers [34]. Many studies have been done on the effects of social media use on the drinking habits of young people. [35] applied longitudinal social media analysis to study the effects of alcohol consumption during the early years of college. This study found that college students who tweeted about alcohol beginning in the early years of college (the Alcohol group) went on to mention alcohol with greater frequency during the later years of college than their counterparts (the Control group) who had not tweeted about drinking early on. The researchers also found that sex topic mentions also occurred with greater frequency amongst the Alcohol group than the Control group, whereas the Alcohol group tweeted less about work/employment throughout college. These findings indicate that social media can be leveraged to show that those adolescents who begin drinking at an earlier phase of a shared experience, like college,

are more likely to engage in other risky behaviors during that shared experience. Likewise, our analysis of text messages provides a means for tracking the longitudinal behaviors of students over time, although we focus on high school instead of college, where there are more mitigating factors at play, like the fact that the participants still live at home with their families.

[36], on the other hand, focuses on the influence of social media on alcohol use among adolescents, rather than using it to gain evidence of self-reported alcohol use. This study is in agreement with [35] that "...adolescents who display one health-risk behavior on social media are more likely to also display other behaviors." The paper highlights the social aspect of social media as threatening to the spread of risky behaviors amongst social networks; an adolescent is more likely to engage in alcohol abuse if their immediate friends do as well. Therefore, studying even more private avenues of communication between friends is likely to reveal how adolescents' communication about risky behaviors influences the adoption of these behaviors amongst their social groups. The text messaging data described in this thesis represents open communication between the participants, which contains similar levels of profane language and sexual themes as the proportions observed in un-monitored online chat rooms [37]. The work by [2] highlights the importance of developing automatic approaches for analyzing the data, as the conclusions drawn in that work are a result of taking just a two-day sample from the entire dataset of texting correspondence across four years.

To our knowledge, this thesis presents the first effort towards predicting substance use amongst adolescents using machine learning analysis of raw text messaging data. However, this type of research has been applied to analyze alcohol consumption patterns based on social media activities [16, 14, 15, 35, 36]. Existing efforts rely directly on alcohol-related keyword lists and specific metrics to filter posts from sites such as Twitter and Facebook. [15] monitors the rate of alcohol consumption in the UK by developing a novel Social Media Alcohol Index, based on counts of six key-terms relating to alcohol, and predict levels

of consumption consistent with the estimates from the Health and Social Care Information Centre in the UK, where the study was performed. The work of [16] addresses the variability in drinking-related language, particularly in an informal social media setting, by specifically consulting with a group of college students to augment their list of pre-defined keywords with common alternatives and misspellings. This highly tailored approach achieves 100% accuracy in classifying each tweet as alcohol-related or not. [10] further highlights the need to preprocess conversational, noisy texts such as tweets and text messages. The authors perform a variety of steps before classification, including truncating multiple occurrences of letters and splitting the messages into tri-gram linguistic feature sets.

In this thesis, we propose word2vec as an alternative to these types of manually curated coding schemes and preprocessing. The algorithm has been shown to develop similar vectors for similar words, and is, by design, capable of relating misspelled words, including those with repeated letters, to their correct counterparts without interference, as we show in Chapter 4. Furthermore, the application of doc2vec removes the necessity of extracting N-grams from the data. An entire text excerpt can be passed to the doc2vec algorithm to infer a vector representation, and concatenation of specific linguistic feature sets is not necessary, nor is the addition of any annotations or preprocessing [11]. This thesis shows that machine learning methods can be leveraged to analyze large amounts of text messages very quickly towards the goal of predicting self-reported alcohol use.

Research methods to study drinking behaviors beyond self-report surveys, particularly amongst adolescents, have been to-date focused primarily on social media usage. While young people do overwhelmingly use social media applications and share many details of their lives online, they are also more likely to portray themselves in a positive light in a public setting [38]. Applying machine learning techniques to text message data captured throughout a subject's high school years will complement existing methods by providing a window into less-filtered communication by these individuals. It will further provide the opportunity to study holistic

communication as it applies to drinking activities and enable latent features indicative of alcohol consumption to be capitalized upon.

6.2 Natural Language Processing of Short Texts

Since the advent of the Turing Test in 1950, machine learning researchers have been interested in the field of natural language processing (NLP): teaching a computer to understand and generate text in the way a human would to the extent that it is impossible to distinguish reliably between conversation generated between a human and a program [39]. In recent years, the study of NLP has been heavily influenced by deep learning techniques such as Long Short Term Memory networks [40] and the Skip-Gram model [26]. These methods are able to be trained to predict latent topics in text, conduct sentiment analysis, and generate text translations, among many other applications. However, performing text classification on short, unstructured texts such as the text messaging data that forms our corpus poses a unique challenge that has yet to be fully mastered.

One very common application of natural language processing methods to short text is the analysis of tweets. It is very easy to download vast quantities of text data from Twitter via their API, and the social media platform has become a popular source of data for text processing as a result. Tweets have been used to track a myriad of interesting societal and cultural phenomena, from real-time HIV-risk analysis to the spread of the Influenza virus [41, 42]. These studies show that short, informal texts can be used in classification tasks to predict the topics present in the samples. In particular, studies involving tweets are applicable to the work of this thesis based on the syntactic similarity between tweets and text messages. A 2013 paper comparing the linguistics of tweeting versus SMS messaging found that in both cases, people tend to use more conversational vernacular and specific slang language topical to their regions as well as temporal references [38]. In addition, the 150-

character limit imposed on tweets often forces people to use acronyms or improper grammar to express their ideas. This resembles adolescent's tendency to engage in "SMS" language [1]. However, the authors show that tweeting actually conforms more greatly to proper grammar rules than SMS messaging. Of particular note are the statistics they present on lexical density (LD), finding that SMS falls at the low end of the spectrum of all digital communication media. A low LD indicates less frequent usage of information-carrying words within a textual body. This finding is indeed corroborated by our dataset; many of the text messages carry no meaningful content at all, but rather represent reactions to other messages, such as laughter or surprise. This negatively effects the ability of text processing algorithms to derive meaning from text messaging data, as was discussed in Chapter 3 of this thesis.

Previous research reveals that training any sort of text analytics model on short texts is difficult without developing methods for expanding or augmenting each training example or developing novel frameworks. Most of the methods rely largely on the development of domain-specific keyword lists that are used to filter training examples, and discussed previously [43, 38, 44, 16, 14, 15]. Others augment tweets with additional keywords drawn from computing similarity between words involved in the original tweet and a set of curated additional text. In [45], the authors extract entities, defined as words or phrases, from the tweet and then map these to a concept set. An external database of concepts that links related terms allows this mapping to occur. A popular strategy is to leverage pre-trained word2vec vectors to compute similarity between words native to the tweet and words from large external corpuses. In some cases, the vectors trained from Wikipedia and provided by the original authors of the word2vec paper are used to generally augment given data with contextual knowledge [46]. In others, a specific word2vec model is trained on a domain-specific external corpus. [43] augments billing records data from a Health Insurance company by using a word2vec model learned from a corpus of medical articles. This paper predicts patient outcomes based on billing codes by computing the similarity between the word2vec embed-

[1] https://en.wikipedia.org/wiki/SMS_language

dings for the description of the billing code and the description of the outcome, respectively, and using this similarity metric to scale the billing codes' influence on the prediction task. The works described above all offer compelling reasons to aggregate all short text examples into longer documents representative of their generating source, i.e. to create "logs" of all communication by a specific participant instead of attempting to classify on a text-by-text basis.

A previous application of doc2vec itself to tweet data in [44] used the resulting embeddings to cluster users with similar communication patterns and recommend tweets of particular interest for a user. These authors explored the doc2vec model applied to "user history" constructed by aggregating all tweets by a single user into a document. We follow a similar approach to building a "texting history" for the participants of the Blackberry Project study, both by taking weekly snapshots of their correspondence and by randomly sampling from their overall communication. With this thesis, we find that doc2vec works well for the task of text transformation for classification, when individual text messages are concatenated to form representations of overall communication. By applying doc2vec directly to the communication excerpts of our subjects, instead of leveraging word2vec similarity on specific keywords, our approach gives the model the freedom to learn latent patterns in the communication of the students.

Chapter 7

Conclusion

Given the near constant engagement by young people with their electronic devices, and the proclivity to discuss social behaviors such as drinking over text message, the content generated by adolescents' online activities presents a wealth of information prime for largescale data mining and machine learning efforts. Within this thesis, we explore a machine learning approach to vectorizing text message data based on contextual and semantic structure for use as input to a logistic regression binary classification algorithm to predict substance use patterns.

We draw from a database of 214 individuals' text messages, captured directly from their mobile devices, to generate vector embeddings of correspondence excerpts using the doc2vec algorithm. These original excerpts are constructed through the selection of both all weekly communication and groups of 50 randomly sampled messages to experiment with the differences in embeddings for cohesive, linear communication versus randomly selected messages. Based on prediction accuracies achieved by models built for each time period over which alcohol consumption information was reported for participants, distributed vector representations of weekly communication portray the latent characteristics in an individuals'

communication with the greatest applicability towards our task. These classifiers achieve above 80% accuracy for each time period, with a maximum of 84.0% accuracy during TC13. Moreover, analysis of the vector representations of words within our text messaging corpus reveals the ability of the machine learning techniques presented herein to overcome problems of vocabulary inconsistency to draw meaningful connections between similar terms. These results validate our approach as a useful methodology for analyzing text message data in an efficient, unsupervised manner with machine learning.

This study lays the foundation for future studies into how the contextual patterns of an individual's text messaging communication reveal his or her wider behavioral patterns. Given the success attained at classifying self-reported alcohol consumption using these doc2vec embeddings as "communication profiles" for participants, we hope to apply this methodology to predicting other risky behaviors and wider mental health issues. We hope this work and future efforts will provide a basis for development of non-intrusive intervention techniques to provide targeted resources to youth based on indicators in their mobile communication.

REFERENCES

[1] Ling R. Campbell S. Purcell K. Lenhart, A. Teens and mobile phones. `http://pewinternetorg/Reports/2010/Teens-and-Mobile-Phones.aspx`, 2010. Accessed: 2019-04-14.

[2] David More Samuel E. Enhrenreich Marion K. Underwood, Lisa H. Rosen and Joanna K. Gentsch. The blackberry project: capturing the content of adolescents' text messaging. *Developmental psychology*, 48(2):295–302, 2011.

[3] P. Greenfield and Z. Yan. Children, adolescents, and the internet: A new field of inquiry in developmental psychology. *Developmental psychology*, 42:391–394, 2006.

[4] Sanders-Jackson A. Smallwood A. M. K Bryant, J. A. Iming, text messaging, and adolescent social networks. *Journal of Computer- Mediated Communication*, 11:577592, 2006.

[5] R Ling. Texting as a life phase phenomenon. *Journal of Computer-Mediated Communication*, 15:277292, 2010.

[6] R Ling. *Mobile communications vis-a-vis teen emancipation, peer group integration, and deviance*, chapter The inside text: Social, cultural, and design perspectives on SMS, pages 175–193. Springer, Dordrecht, the Netherlands, 2005.

[7] T. Pierce. Social anxiety and technology: Face-to-face communication versus technological communication among teens. *Computers in Human Behavior*, 25:13671372, 2009.

[8] Reid J. M. Reid, D. J. Text or talk? social anxiety, loneliness, and divergent preferences for cell phone use. *CyberPsychology Behavior*, 10:424435, 2007.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, G.s Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 10 2013.

[10] Georgiana Dinu Marco Baroni and German Kruszewski. Dont count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. page 238247, 06 2014.

[11] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. pages 78–86, 07 2016.

[12] O Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. *Advances in Neural Information Processing Systems*, 3:2177–2185, 01 2014.

[13] Tomas Mikolov Quoc Le. Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on Machine Learning*, 32, 2014.

[14] Nabil Hossain, Tianran Hu, Roghayeh Feizi, Ann Marie White, Jiebo Luo, and Henry Kautz. Inferring fine-grained details on user activities and home location from social media: Detecting drinking-while-tweeting patterns in communities. 03 2016.

[15] Patrick Stacey Daniel Kershaw, Matthew Rowe. Towards tracking and analysing regional alcohol consumption patterns in the uk through the use of social media. pages 220–228, 06 2014.

[16] Saleem Alhabash, Courtland Vandam, Pang-Ning Tan, Sandi W. Smith, Gregory Viken, Duygu Kanver, Liang Tian, and Luiz Figueira. 140 characters of intoxication: Exploring the prevalence of alcohol-related tweets and predicting their virality. *SAGE Open*, 8, 10 2018.

[17] Marion Underwood, Samuel Ehrenreich, David More, Jerome S. Solis, and Dawn Brinkley. The blackberry project: The hidden world of adolescents' text messaging and relations with internalizing symptoms. *Journal of Research on Adolescence*, 25, 12 2013.

[18] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[19] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent dirichlet allocation for tag recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 61–68, New York, NY, USA, 2009. ACM.

[20] M. Lienou, H. Maitre, and M. Datcu. Semantic annotation of satellite images using latent dirichlet allocation. *IEEE Geoscience and Remote Sensing Letters*, 7(1):28–32, Jan 2010.

[21] Seshadri Tirunillai and Gerard J. Tellis. Mining marketing meaning from online chatter: Strategic brand analysis of big data using latent dirichlet allocation. *Journal of Marketing Research*, 51:463–479, 08 2014.

[22] Bing Liu, Lin Liu, Anna Tsykin, Gregory J. Goodall, Jeffrey E. Green, Min Zhu, Chang Hee Kim, and Jiuyong Li. Identifying functional miRNAmRNA regulatory modules with correspondence latent dirichlet allocation. *Bioinformatics*, 26(24):3105–3111, 10 2010.

[23] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. `http://mallet.cs.umass.edu.2002`.

[24] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[25] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[26] Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. 09 2013.

[27] Geoffrey Hinton and Terrence J. Sejnowski. *Unsupervised Learning: Foundations of Neural Computation (Computational Neuroscience)*. MIT Press, May 1999.

[28] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162, 09 2013.

[29] Laurens van der Maaten and Geoffrey Hinton. Viualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 11 2008.

[30] Martin Wattenberg, Fernanda Vigas, and Ian Johnson. How to use t-sne effectively. *Distill*, 2016.

[31] Yla R. Tausczik and James Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29:24–54, 03 2010.

[32] Brian Suffoletto, Jeffrey Kristan, Clifton Callaway, Kevin H. Kim, Tammy Chung, Peter Monti, and Duncan B. Clark. A text message alcohol intervention for young adult emergency department patients: a randomized clinical trial. *Annals of emergency medicine*, 64 6:664–72.e4, 2014.

[33] Nancy Berkman, Stacey L Sheridan, Katrina Donahue, David J Halpern, and Karen Crotty. Low health literacy and health outcomes: An updated systematic review. *Annals of internal medicine*, 155:97–107, 07 2011.

[34] Reich S. M. Waechter N. Espinoza G. Subrahmanyam, K. Online and off-line social networks: Use of social networking sites by emerging adults. *Journal of Applied Developmental Psychology*, 29:420–433, 2008.

[35] Emre Kiciman, Scott Counts, and Melissa Gasser. Using longitudinal social media analysis to understand the effects of early college alcohol use. In *ICWSM*, 2018.

[36] Megan A Moreno and Jennifer M. Whitehill. Influence of social media on alcohol use in adolescents and young adults. In *Alcohol research : current reviews*, 2014.

[37] Smahel D. Subrahmanyam, K. and P. Greenfield. Connecting developmental constructions to the internet: Identity presentation and sexual exploration in online teen chat rooms. *Developmental Psychology*, 42:395–406, 2006.

[38] Y Hu, K Talamadupula, and S Kambhampati. Dude, srsly?: The surprisingly formal nature of twitter's language. *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013*, pages 244–253, 01 2013.

[39] Alan Turing. Computing machinery and intelligence. *Mind*, 49:433–460, 1950.

[40] Sepp Hochreiter and Jrgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[41] Wenchao Yu Sean Young and Wei Wang. Toward automating hiv identification: Machine learning for rapid identification of hiv-related social media data. *Journal of Acquired Immune Deficiency Syndromes*, 74, 02 2017.

[42] Alberto Maria Segre Alessio Signorini and Phillip M. Polgreen. The use of twitter to track levels of disease activity and public concern in the u.s. during the influenza a h1n1 pandemic. *PloS one*, 6(5), 05 2011.

[43] Yun Liu, Collin M. Stultz, John V. Guttag, Kun-Ta Chuang, Fu-Wen Liang, and Huey-Jen Su. Transferring knowledge from text to predict disease onset. In *MLHC*, 2016.

[44] Abdullah Shobi Mojtaba Zahedi Amiri. A link prediction strategy for personalized tweet recommendation through doc2vec approach. *Research in Economics and Management*, 2(4), 2017.

[45] Fang Wang, Zhongyuan Wang, Zhoujun Li, and Ji-Rong Wen. Concept-based short text classification and ranking. pages 1069–1078, 11 2014.

[46] Peng Wang, Bo Xu, Jiaming Xu, Guanhua Tian, Cheng-Lin Liu, and Hongwei Hao. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*, 174, 10 2015.

Appendix A

Sample Daily Log File

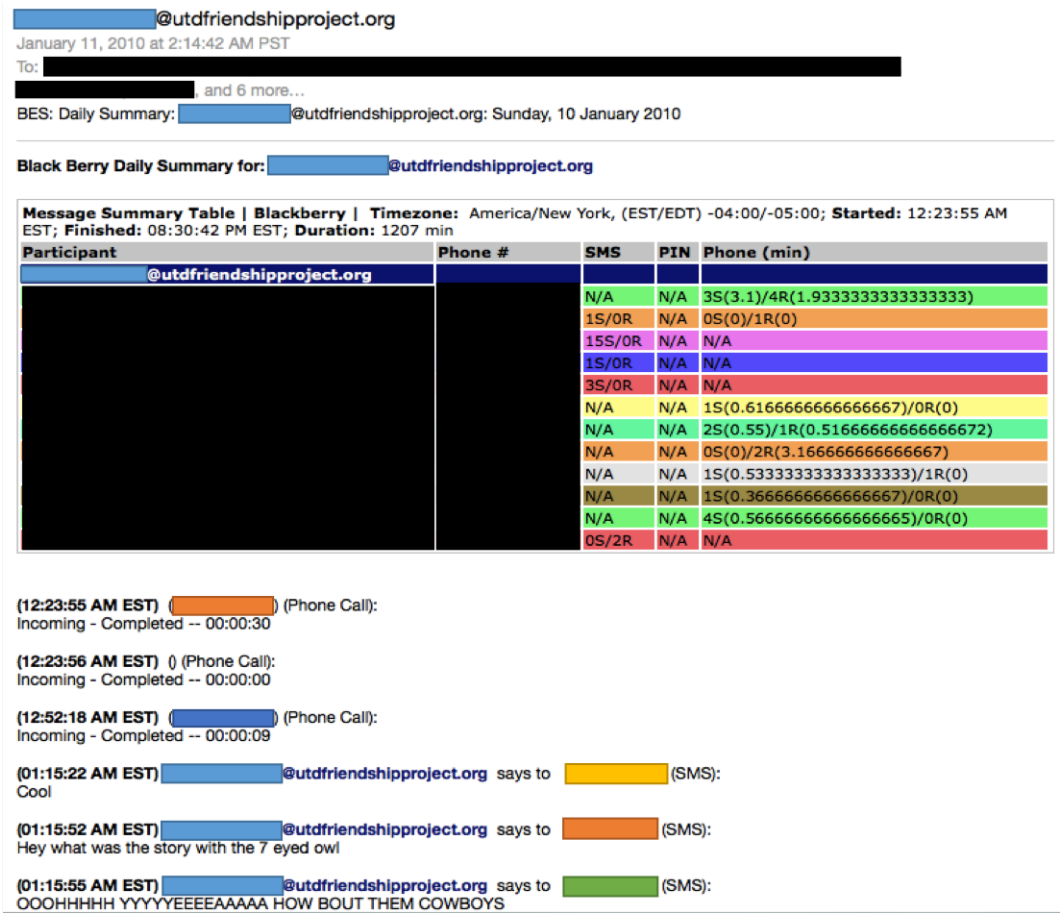


Figure A.1: Excerpt from a daily log file. Log files were provided in .eml format and contained all incoming and outgoing messages captured from a participant's device during one 24 hour period, along with phone call records and a table of contact information from the participant's device.

Appendix B

Source Code Sample for t-SNE Visualization

```
from gensim.models import Doc2Vec
from sklearn.manifold import TSNE
import numpy as np, matplotlib.pyplot as plt, matplotlib.cm as cm

def cluster_words_embeddings(model, keys, n):
    embedding_clusters, word_clusters = [], []
    for word in keys:
        embeddings, words = [], []
        for similar_word, _ in model.wv.most_similar([word], topn=n):
            words.append(similar_word)
            embeddings.append(model[similar_word])
        embedding_clusters.append(embeddings)
        word_clusters.append(words)
    return word_clusters, embedding_clusters
```

```
def tsne_2d_dimensionalityreduction(embedding_clusters, p):
    embedding_clusters = np.array(embedding_clusters)
    n, m, k = embedding_clusters.shape
    tsne_model_en_2d = TSNE(perplexity=p, n_components=2, init='pca',
        n_iter=3500, random_state=32)
    embeddings_en_2d = np.array(tsne_model_en_2d.fit_transform(
        embedding_clusters.reshape(n * m, k))).reshape(n, m, 2)
    return embeddings_en_2d




def tsne_plot_similar_words(title, labels,
    embedding_clusters, word_clusters, a, filename=None):

    % matplotlib inline
    plt.figure(figsize=(16, 9))
    colors = cm.rainbow(np.linspace(0, 1, len(labels)))
    for label, embeddings, words, color in zip(labels,
        embedding_clusters, word_clusters, colors):
        x = embeddings[:, 0]
        y = embeddings[:, 1]
        plt.scatter(x, y, c=color, alpha=a, label=label)
        for i, word in enumerate(words):
            plt.annotate(word, alpha=0.5, xy=(x[i], y[i]), xytext=(5, 2),
                         textcoords='offset points', ha='right', va='bottom', size=8)
    plt.legend(loc=4)
    plt.title(title)
    plt.grid(True)
    if filename:
        plt.savefig(filename, format='png', dpi=150, bbox_inches='tight')
    plt.show()

words = [list of target words to visualize/compare]
model = Doc2Vec.load("model path")
word_clusters, embedding_clusters = cluster_words_embeddings(model,
    words, num_neighbors)
embeddings_2d = tsne_2d_dimensionalityreduction(embedding_clusters, 10)
tsne_plot_similar_words("Plot Title", words, embeddings_2d,
    word_clusters, 0.7, "figure.png")
```

Figure B.1: t-SNE plotting source code.

Appendix C

Additional word2vec-Based Word Similarity Visualizations

Taken in aggregate, Figures C.1, C.2, and C.3 demonstrate the differences between the underlying word2vec models for doc2vec models learned with different input document types. With the weekly model, all terms cluster more closely together, showing this model's focus on the topical similarity between them. The model has learned to relate drinking with alcohol and partying to a greater extent than the models based on 50 random messages. Those latter models rely more heavily on pure context, and separate 'party' very starkly from 'alcohol' and 'drinking'.

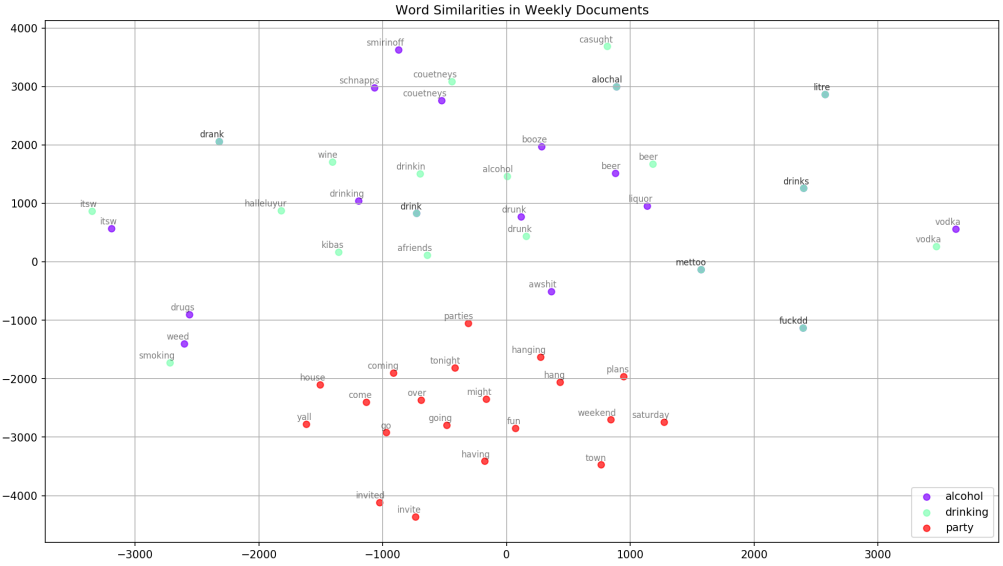


Figure C.1: Clustering for alcohol consumption-related terms' word embeddings from model using documents of weekly correspondence.

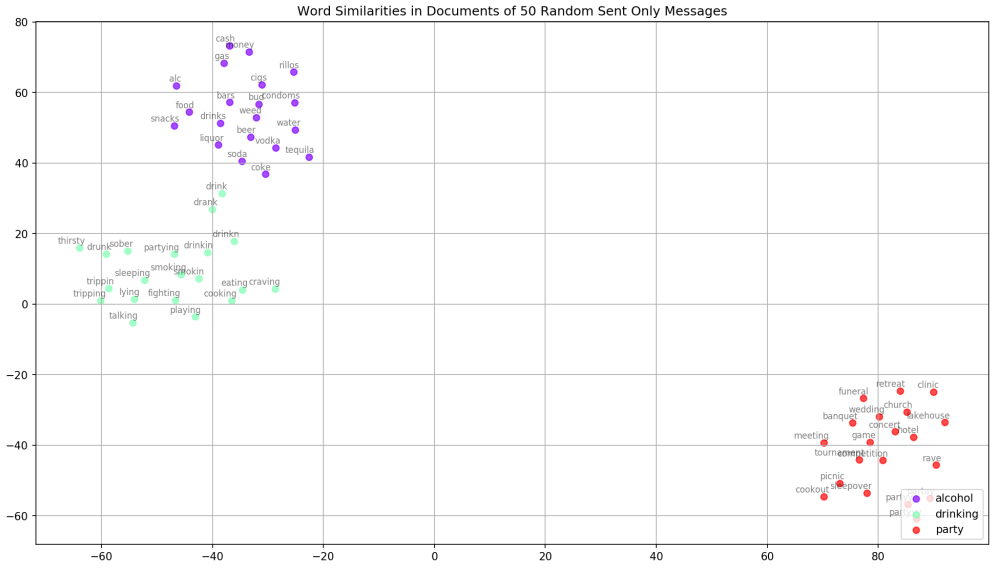


Figure C.2: Clustering for alcohol consumption-related terms' word embeddings from model using documents of sent messages only.

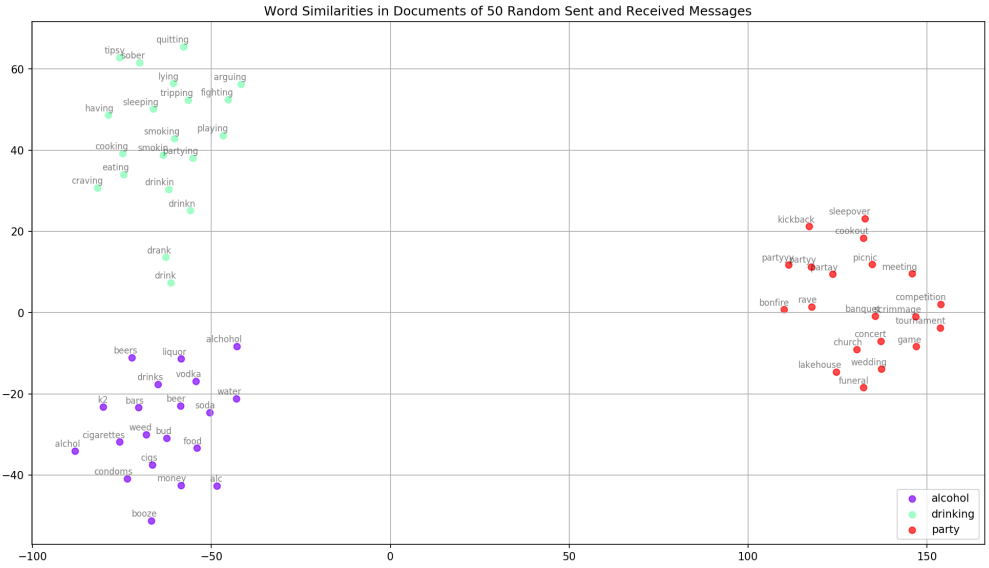


Figure C.3: Clustering for alcohol consumption-related terms' word embeddings from model using documents of sent and received messages.

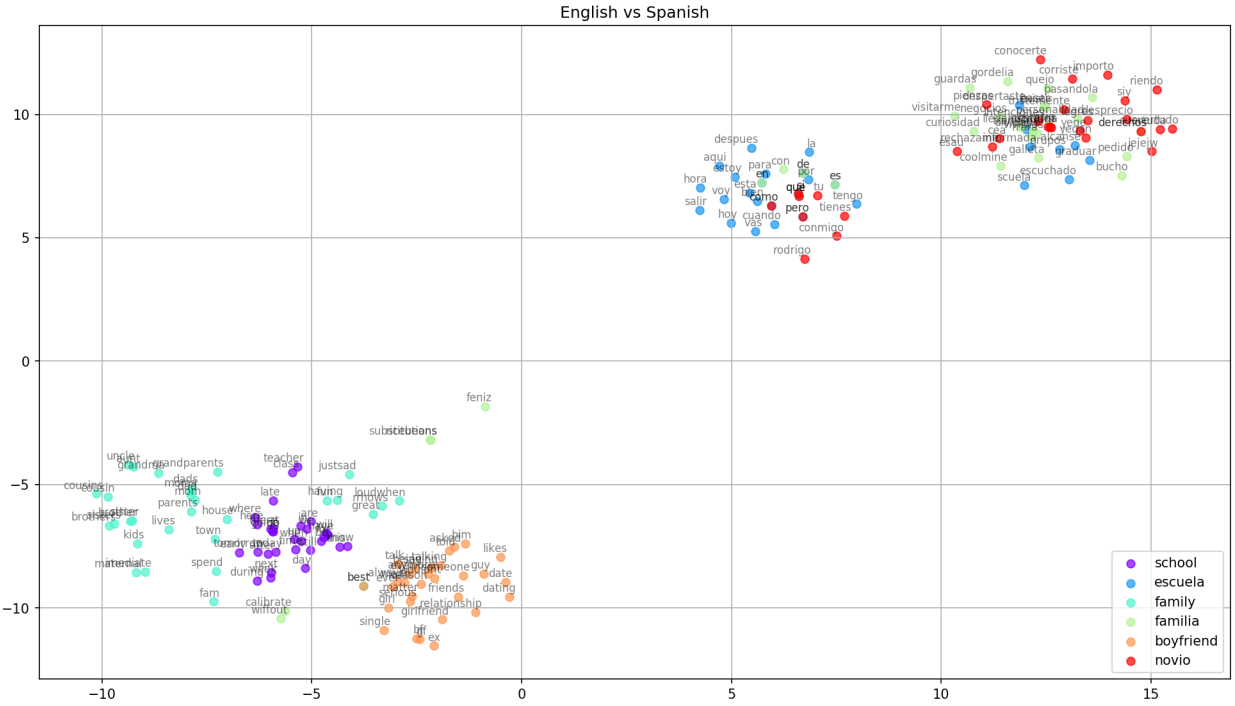


Figure C.4: Clustering of the most similar terms for a selection of English words and their Spanish translations, showing that model generates similar vectors for words based on language. The model draws on context to group all Spanish words together and all English words together.

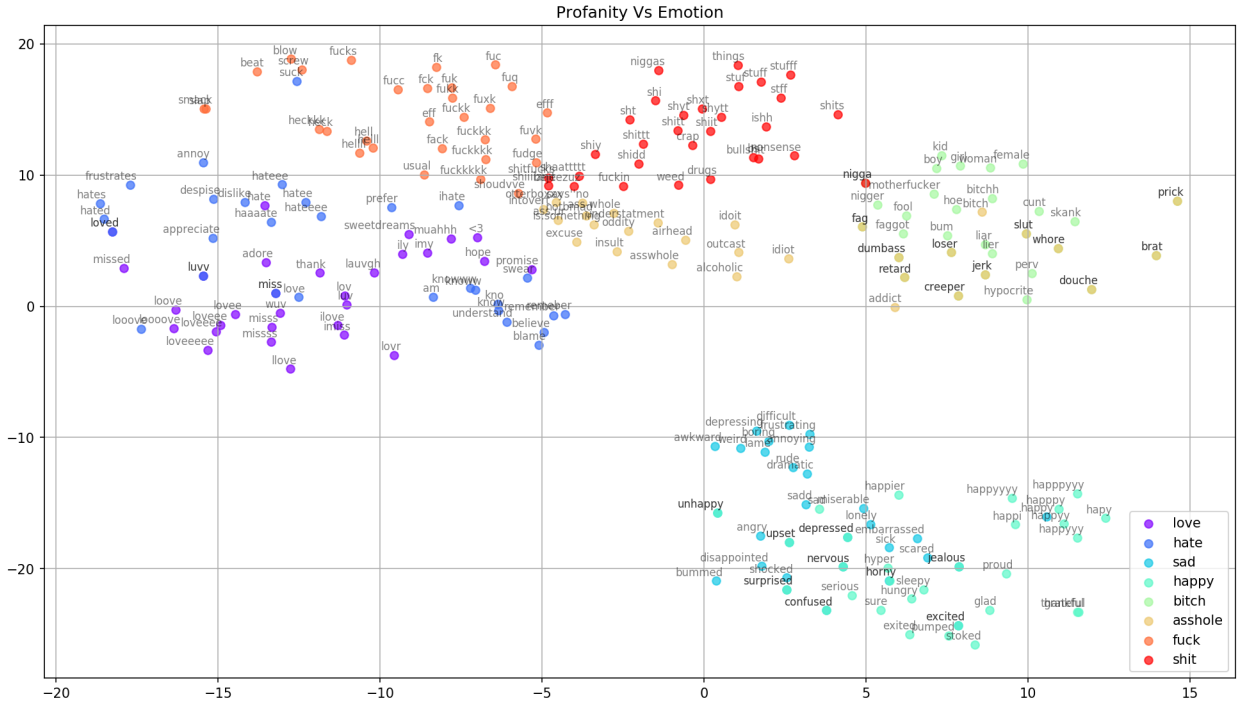


Figure C.5: Clustering of terms related to emotion: “love”, “hate”, “sad”, and “happy”, along with profanity: “fuck”, “shit”, “bitch”, and “asshole”. This visualization reveals an interesting segregation of “happy” and “sad” from the remaining terms, which intermingle. “Love” and “hate” appear linearly separable from the profane terms included, but neighbor those clusters. This indicates a contextual relevance to the “dramatic” tendencies of texting. It is possible that the model has learned to associate the more intense emotions of “love” and “hate” with more intense language.

Appendix D

Additional Word Embedding Cosine Similarity Results

Original Word	Most Similar Word	Cosine Similarity Metric
"beer"	"alcohol"	0.69
	"drinks"	0.69
	"vodka"	0.68
	"weed"	0.67
	"cigs"	0.63
"drinking"	"drinkin"	0.80
	"smoking"	0.73
	"partying"	0.62
	"smokin"	0.59
	"fighting"	0.56
"booze"	"alcohol"	0.64
	"beer"	0.58
	"drinks"	0.56
	"alc"	0.55
	"weed"	0.54
"vodka"	"beer"	0.68
	"rum"	0.65
	"liquor"	0.64
	"wine"	0.64
	"whiskey"	0.64
"drunk"	"stoned"	0.75
	"tipsy"	0.72
	"horny"	0.68
	"paranoid"	0.68
	"pregnant"	0.63
"hungover"	"exhausted"	0.72
	"sleepy"	0.68
	"sore"	0.66
	"sick"	0.65
	"depressed"	0.63

Table D.1: Word embedding similarities for terms relating to drinking from model trained on all sent and received messages. These are example of potential augmentations to a keyword set that could be leveraged to flag messages as pertaining to alcohol-related activities.