LREC 2018 Workshop

TA-COS 2018:
2nd Workshop on Text Analytics for Cybersecurity and Online Safety

PROCEEDINGS

Edited by

Els Lefever, Bart Desmet, Guy De Pauw

ISBN: 979-10-95546-27-6
EAN: 9791095546276

12 May 2018

Proceedings of the LREC 2018 Workshop
TA-COS 2018 – 2nd Workshop on Text Analytics for Cybersecurity and Online Safety

12 May 2018 – Miyazaki, Japan

Edited by Els Lefever, Bart Desmet, Guy De Pauw

http://ta-cos.org

Organising Committee

- Els Lefever, LT3 – Ghent University, Belgium
  https://www.lt3.ugent.be/people/els-lefever/
  els.lefever@ugent.be
- Bart Desmet, LT3 – Ghent University, Belgium
  https://www.lt3.ugent.be/people/bart-desmet/
  bart.desmet@ugent.be
- Guy De Pauw, Textgain – University of Antwerp, Belgium
  https://www.textgain.com/
  guy@textgain.com

Programme Committee

- Walter Daelemans (chair), CLiPS – University of Antwerp, Belgium
- Veronique Hoste (chair), LT3 – Ghent University, Belgium
- Fabio Crestani, University of Lugano, Switserland
- Maral Dadvar, Twente University, The Netherlands
- Guy De Pauw, Textgain, Belgium
- Bart Desmet, Ghent University, Belgium
- Chris Emmery, Tilburg University, The Netherlands
- Lee Gillam, University of Surrey, UK
- Jose Maria Gomez Hidalgo, Pragsis Technologies, Spain
- Els Lefever, Ghent University, Belgium
- Eva Lievens, Ghent University, Belgium
- Haji Mohammad Saleem, McGill University, Canada
- Karolien Poels, University of Antwerp, Belgium
- Stephan Tulkens, University of Antwerp, Belgium
- Cynthia Van Hee, Ghent University, Belgium
- Anna Vartapetiance, University of Surrey, UK
- Zeerak Waseem, University of Sheffield, UK
- Shomir Wilson, Carnegie Mellon University, USA
- Yanan Sun, University of Maryland, Baltimore County, USA

Preface

Text analytics technologies are being widely used as components in Big Data applications, allowing for the extraction of different types of information from large volumes of text, including purely factual information ("traditional" text mining), subjective information (sentiment mining) and even metadata (e.g. author profiling). A growing number of research efforts is now investigating the applicability of these techniques for cybersecurity purposes. Many applications are using text analytics techniques to provide a safer and more pleasant online experience, by detecting unwanted content and behavior on the Internet. Other text analytics approaches attempt to detect illegal activity on online networks or monitor social media against the background of real-life threats.

The second run of the workshop on Text Analytics for Cybersecurity and Online Safety (TA-COS 2018) aims to bring together researchers that have an active interest in the development and application of text analytics systems in the broad context of cybersecurity. We were interested in research papers on text analytics and text mining approaches that (1) help reduce the exposure to harmful content on the Internet, (2) detect illegal online activities and (3) monitor user-generated content in the context of real-life security threats.

Following our call for papers, we received papers on a wide range of topics and with the help of our varied team of reviewers were able to select the most relevant and most interesting contributions. We are very pleased to present a wide variety of topics of the accepted papers for the workshop. Two papers deal with identifying hate speech on social media: Sirihattasak et al. present an annotated corpus and classification experiments for toxic messages in Thai tweets, while Isbister et al. present a case study on monitoring targeted hate in Swedisch online text. Alshehri et al. present a dataset of adult content in Arabic Twitter and provide an in-depth analyses of this data. The fourth paper deals with targeted email attacks. Das and Verma propose a system for advanced email masquerading attacks using Natural Language Generation techniques. We are furthermore very pleased to be able to kick off our workshop with a keynote lecture by Pierre Lison, who is a Senior Research Scientist at Norsk Regnesentral (Norwegian Computing Center), a contract-funded research institute located in Oslo, Norway. He will present research on data-driven models of reputation in cyber-security.

We are sure that the presentations at TA-COS 2018 will trigger fruitful discussions and will help foster the awareness of the increasingly important role text analytics can play in cybersecurity applications.

Els Lefever, Bart Desmet, Guy De Pauw — May 2018

Programme

Keynote

14.00 – 14.10 Introduction

14.10 – 15.00 Pierre Lison
Data-driven models of reputation in cyber-security (invited talk)

Workshop Papers I

15.00 – 15.30 Sugan Sirihattasak, Mamoru Komachi, Hiroshi Ishikawa
Annotation and Classification of Toxicity for Thai Twitter

15.30 – 16.00 Tim Isbister, Magnus Sahlgren, Lisa Kaati, Milan Obaidi, Nazar Akrami
Monitoring Targeted Hate in Online Environments

Break

16.00 – 16.30 Coffee break

Workshop Papers II

16.30 – 17.00 Ali Alshehri, El Moatez Billah Nagoudi, Hassan Alhuzali, Muhammad Abdul-Mageed
Think Before Your Click: Data and Models for Adult Content in Arabic Twitter

17.00 – 17.30 Avisha Das, Rakesh Verma
Automated email Generation for Targeted Attacks using Natural Language

Table of Contents

Annotation and Classification of Toxicity for Thai Twitter

Sugan Sirihattasak, Mamoru Komachi, Hiroshi Ishikawa
Tokyo Metropolitan University
6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan
sirihattasak-sugan@ed.tmu.ac.jp, {komachi, hiroshi-ishikawa}@tmu.ac.jp

Abstract

In this study, we present toxicity annotation for a Thai Twitter Corpus as a preliminary exploration for toxicity analysis in the Thai language. We construct a Thai toxic word dictionary and select 3,300 tweets for annotation using the 44 keywords from our dictionary. We obtained 2,027 and 1,273 toxic and non-toxic tweets, respectively; these were labeled by three annotators. The result of corpus analysis indicates that tweets that include toxic words are not always toxic. Further, it is more likely that a tweet is toxic, if it contains toxic words indicating their original meaning. Moreover, disagreements in annotation are primarily because of sarcasm, unclear existing target, and word sense ambiguity. Finally, we conducted supervised classification using our corpus as a dataset and obtained an accuracy of 0.80, which is comparable with the inter-annotator agreement of this dataset. Our dataset is available on GitHub.

Keywords: toxicity, corpus, Thai, Twitter

1. Introduction

With the rise of social media in Thailand, it has become an integral part of the daily lives of Thai people, providing various opportunities for education, relationships, and career development. Despite these benefits, online toxicity is not only becoming harsher, but also difficult to control. Furthermore, the victims of toxic messages are not always the intended targets of those messages. According to Wang et al. (2011), many people regret their negative posts because of problems they face later, such as being terminated from employment and losing other opportunities. The instances of bullying or any similar toxic behavior are not easy to delete once they are posted publicly. In particular, any post shared on social media can potentially spread widely across an entire community with a considerably small possibility of deleting it and undoing its effects.

Consequently, there have been many research efforts among various fields such as social science, psychology, and natural language processing, to improve the quality of online conversion while considering the right to freedom of speech. For example, the Google Jigsaw Team launched the Perspective API[1] to identify toxic comments.

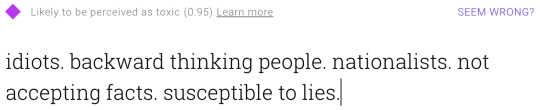

Figure 1: Example of toxicity evaluation from Perspective API.

One of the challenges in studying toxicity in online communication is a clear common definition of toxicity in the case of language. Toxic comments are often sarcastic and indicate aggressive disagreement; in Kolhatkar and Taboada (2017), the relationship between constructiveness and toxicity including toxicity levels in news comments was studied. In our study, we define toxicity with a more general perspective to include any messages that can imply toxic behavior (Kwak and Blackburn, 2014), antisocial behavior (Cheng et al., 2017), online harassment (Yin et al., 2009), hate speech (Davidson et al., 2017), cyberbullying (Van Hee et al., 2015), and any type of offensive language (Razavi et al., 2010). In particular, a toxic message is any message that may hurt or harm an individual or a generalized group, may challenge the societal norms, or negatively affect the entire community. In terms of toxic words, we consider any negative words, such as those associated with profanity and obscenity, or those which are offensive.

Though there is an increase in the studies related to toxicity, open resources related to it are still limited. There are several corpora for major languages like English, including a harassment dataset (Kennedy et al., 2017), hate speech Twitter annotation corpus (Waseem and Hovy, 2016), and personal attacks comment corpus (Wulczyn et al., 2017). Unfortunately, researches related to this topic do not include minor languages, such as the Thai language. To our best knowledge, there is no public Thai resource related to online toxicity. Furthermore, text analysis in Thai language is complicated due to ambiguity in segmentation (Cooper, 1996); for example, "ปลาตากลมตัวนี้น่ารัก (This round-eyes (ตา ∣ กลม) fish is cute.)" and "ขอเดินออกไปตากลม (Let me go out to have some fresh air (ตาก ∣ ลม))." Likewise, sentence boundary detection is difficult (Zhou et al., 2016) because the space which is used for differentiating sentences is not appropriate in some cases such as in "โอ๊ย! เจ็บ (Ouch! it hurts)."

Some toxic tweets that are typical in the case of bullying messages, such as "ไอ้ห่า! ไปตายซะ คนไร้ประโยชน์ แกก็เหมือนพ่อแก" (Damn you! Just go to die. You are useless just like your father.), may not only affect an individual, but also his or her family. Thus, we present annotation and classification of toxicity on Twitter in the Thai language as a preliminary exploration for toxicity analysis in the Thai language in general. The main contributions of this study are as follows:

1. We construct a dictionary of Thai toxic words that we use as keywords for annotation.

[1] http://www.perspectiveapi.com/

2. We build a toxicity corpus based on Twitter messages or tweets, because these messages represent the daily-life conversations of the Thai people.

3. We used our abovementioned dataset to conduct supervised classification and obtained an accuracy of 0.80 for it.

Our dictionary and corpus are available on GitHub[2].
The remainder of this paper is organized as follows. Section 2 introduces the definition of toxicity and describes some difficulties with respect to Thai tweet analysis. Section 3 explains our corpus construction and annotation process including the construction of our dictionary of Thai toxic words. Then, Section 4 presents the analysis of the resulting corpus, while Section 5 provides classification results and discussion. Finally, Section 6 presents the conclusions of our study and indicates future work.

2. Toxicity and Thai Language

Many social media platforms and websites use embedded keyword-based approaches to automatically filter out toxic messages. However, it is possible for individuals who are close friends to casually communicate using toxic words without intending any harm (Nand et al., 2016). Likewise, the factors used to identify politeness in Thai male conversation depend on the situational context such as the relationship between the speaker and listener, and the location at which the conversation takes place, rather than the linguistic aspects (Mekthawornwathana, 2011).
Moreover, the keyword-based approach does not seem flexible for a non-segmenting language like the Thai language. The following two examples contain a toxic word "หอก[3]" (The original meaning is "spear"; however, the slang meaning is an insulting phrase, "Damn, Bitch.")

(a) นักการเมืองหอกเลวมากสมควรตาย
นักการเมือง (politician) | หอก (damn) | เลว (bad) | มาก (very) | สมควร (deserve) | ตาย (die)
The damn Politician deserves to die.
(This is a toxic message.)

(b) ที่หอกล้องวงจรปิดเยอะจึงไม่มีหัวขโมย
ที่ (at) | หอ (dormitory) | กล้องวงจรปิด (security camera) | เยอะ (many) | จึง (therefore) | ไม่ (no) | มี (have) | หัวขโมย (thief/thieves)
There are no thieves because there are a lot of security cameras at the dormitory.
(This is a non-toxic message.)

Therefore, not only ambiguity in segmenting as shown above, but also word variations and homonyms are inevitable obstacles in Thai tweet analysis. For example, the toxic word "เหี้ย" has several homonyms including the following examples presented below.

(a) นักกีฬาประเทศนี้เหี้ยโกงตลอด
นักกีฬา (athlete) | ประเทศ (country) | นี้ (this) | เหี้ย (damn/bad) | โกง (cheat) | ตลอด (always)
An athlete from this country always cheats.
(This is a toxic message.)

(b) อากาศร้อนเหี้ย
อากาศ (weather) | ร้อน (hot) | เหี้ย (damn/very)
The weather is very hot.
(This is a non-toxic message.)

(c) เหี้ยเป็นสัตว์เลื้อยคลาน
เหี้ย (varanus salvator) | เป็น (is) | สัตว์เลื้อยคลาน (reptile)
Varanus salvator is a reptile.
(This is a non-toxic message.)

Thus, the classification of toxicity should not only depend on a word, but also the context in which it is used. In order to achieve this, we need to apply a data-driven approach because a keyword-based approach is insufficient (Saleem et al., 2016); we do this by creating a corpus that contains a variety of examples of toxicity in the Thai language.

3. Dataset Construction and Annotation

3.1. Keyword Dictionary Construction

Because toxic posts often contain toxic words, we used toxic words as the keywords to retrieve the data for our dictionary. We selected some toxic words from the Conceptual Metaphor of Thai Curse Words (Orathai Chinakarapong, 2014) and rechecked spelling using the Royal Institute Dictionary[4]. Then, we added some well-known variations of these toxic words such as "สัส," which is a spelling variation of "สัตว์" (The original meaning of this word is animal and its slang meaning is similar to "damn."). Finally, we included a few negative words, for example, "ฆ่า" (kill) and "แช่ง" (curse), into the set. In total, we included 44 keywords in this dictionary, which are shown in Figure 2.

3.2. Data Collection

We used the public Twitter Search API to collect 9,819 tweets from January–December 2017 based on our keyword dictionary. Then, we selected 75 tweets for each keyword. In total, we collected 3,300 tweets for annotation.
To ensure quality of data, we set the following selection criteria.

1. All tweets are selected by humans to prevent word ambiguity. (The Twitter API selected the tweets based on characters in the keyword. For example, in the case of "บ้า(crazy)," the API will also select "บ้านนอก" (countryside)" which is not our target.)

[2] https://github.com/tmu-nlp/ThaiToxicityTweetCorpus/

[3] This paper contains several inappropriate, impolite, and harsh words in both the Thai and English languages. We rewrite some English toxic words using "*" for some characters or replacing these words with appropriate substitutes. However, we could not rewrite such words for the Thai language because that may lead to an ambiguous word.

[4] http://www.royin.go.th/dictionary

2. The length of the tweet should be sufficiently long to discern the context of the tweet. Hence, we set five words as the minimum limit.

3. The tweets that contain only extremely toxic words, (for example: "damn, retard, bitch, f*ck, slut!!!") are not considered.

4. In addition, we allowed tweets with English words if they were not critical elements in the labeling decision, for example, the word "f*ck." As a result, our corpus contains English words, but they are less than 2% of the total.

All hashtags, re-tweets, and links were removed from these tweets. However, we did not delete emoticons because these emotional icons can imply the real intent of the post owners. Furthermore, only in the case of annotation, some entries such as the names of famous people were replaced with a tag <ไม่ขอเปิดเผยชื่อ>, for anonymity to prevent individual bias.

3.3. Annotation

We manually annotated our dataset with two labels: Toxic and Non-Toxic. We define a message as toxic if it indicates any harmful, damage, or negative intent based on our definition of toxicity. Furthermore, all the tweets were annotated by three annotators to identify toxicity; the conditions used for this identification are presented in the following list.

- A toxic message is a message that should be deleted or not be allowed in public.
- A message's target or consequence must exist. It can either be an individual or a generalized group based on a commonality such as religion or ethnicity, or an entire community.
- Self-complain is not considered toxic, because it is not harmful to anyone. However, if self-complain is intended to indicate something bad, it will be considered as toxic.
- Both direct and indirect messages including those with sarcasm are taken into consideration.

We strictly instructed all the annotators about these concepts and asked them to perform a small test to ensure they understood these conditions. The annotation process was divided into two rounds. We asked the candidates to annotate their answers in the first round to learn our annotation standard. Then, we asked them to annotate a different dataset and selected the ones who obtained a full-score for the second round as an annotator. From among these annotators, 20% of the annotators failed the first round and were not involved in the final annotation.

4. Corpus Analysis

As previously mentioned, the corpus consists of 3,300 tweets divided into 2,027 toxic tweets and 1,273 non-toxic tweets. The labels are assigned based on majority decisions. The numbers of tweets with perfect agreement, referred to as gold standard tweets, are 1,692 and 1,093 for toxic and non-toxic cases, respectively. The inter-annotator agreement (Fleiss' Kappa) (Carletta, 1996) is 0.78, which shows that the agreement is significant.

There are three primary reasons for disagreement. First, more than 35% of tweets that annotators disagreed upon are difficult to judge as toxic or non-toxic because of sarcasm. Second, it is ambiguous whether a message owner is self-complaining or referring to someone else or some group by cunning to avoid defamation. Lastly, there are some cases where word sense ambiguity is affected by the annotation. For example; "Damn it, I want to commit arson on the university," which can imply that he/she is very stressed out and just wants to complain. This kind of sarcastic expression is quite common in Thailand. However, there is a possibility that the owner of the comment really intends to commit such a crime.

The distribution of toxic and non-toxic tweets is shown in Figure 2. Interestingly, the tweets that contain toxic words related to animals are less likely to be toxic than the rest except in the cases of "แมงดา" (pimp/horseshoe crabs) and "ควาย" (stupid/buffalo). Most of the non-toxic cases for "แมงดา" refer to one of Thailand's popular dish that is made from horseshoe crabs while "ควาย" seems to be rarely used for its literal meaning of buffalo. Moreover, the words that related to bottom like "ต่ำ" (low) and "ส้นตีน" (heel) are commonly used in a toxic manner because they are antonyms to the words "top" or "high" which Thai people believe indicate a sacred position like a head. The word "โง่" (stupid) seems to be used in a non-toxic manner rather than for toxic purposes. Based on the non-toxic tweets from our corpus, we found that people tend to use the word "stupid" whenever they want to blame themselves. Moreover, as part of everyday conversation, people use the word "หมา" (dog) not only as an insult, but also to refer to a pet or as an adorable joke. Surprisingly, the usage of the word "ชั่ว" (wicked) is not limited as a toxic word, but we found that, in everyday conversation, like in the case of teaching or reporting a situation, it is used in a non-toxic manner as well. Finally, the word "สัตว์" (animal) is used by people for its original non-toxic meaning. This is in contrast to its variations such as "สัส" and "สัด," which are more likely to be used in a toxic manner.

In the case of toxic tweets, we found that a word, "ควย," which refers to f*ck or genitalia, is highly toxic and unpleasant regardless of the level of contextual toxicity.

Some tweets are difficult to label leading to inconsistency in annotation as shown in Table 1. Moreover, Thai people often use metaphors in their conversations as indicated in the example below.

กินกะหรี่ปั๊บอร่อยไม่เหมือนกินกะหรี่

กิน (eat) | กะหรี่ปั๊บ (curry puff) | อร่อย (yummy/delicious) | ไม่ (not) | เหมือน (similar to) | กิน (eat) | กะหรี่ (curry? whore?)

Eating curry puff is yummy not like eating curry (whore?).

In such cases, it is difficult to ascertain the meaning of the

Table 1: Top three conflicts in annotation agreement.

Keywords (original/toxic meaning)	Disagreement of tweets (%)
กะหรี่ (curry/whore)	22.7
ห่า (damn) หอก (spear/bitch) ฉิบหาย (woeful)	21.3
ตอแหล (lie) เห็บ (tick/parasite) ปลวก (termite/ugly) ประสาท (nerve/insane) ส้นตีน (heel) ดัดจริต (pretentious) แช่ง (curse) จัญไร (beastly)	20.0

word "กะหรี่"; thus, its purpose is vague and could either indicate a warning or be an attack against someone. These types of tweets are common in Thai Twitter because people avoid mentioning the target of the message directly to prevent legal repercussions or other issues.

5. Classification Experiment

5.1. Data

Aside from the steps performed for annotation, we conduct further tweet data cleaning after we have segmented the tweets into tokens using the Deepcut library version 0.6[5].

1. We normalized repetitive letters, for example, "มากกก" to "มาก" and "5555..." to "555." The pronunciation in Thai for number 5 is "Ha," therefore, people always use it as a substitute for the laugh sound.
2. We removed stopwords and punctuation marks except "?" and "!" because they may be related to some emotions.
3. We removed non-Thai words.

In order to make a fair comparison, the training data is created by selecting equal number of toxic and non-toxic instances from the corpus; in particular, we selected 1,888 tweets with 944 toxic tweets and 944 non-toxic tweets. All of these tweets were selected randomly. Furthermore, each keyword must have an equal number of tweets for both labels and the maximum number of tweets per label is 30. For test data, we used 176 tweets from among the gold standard tweets with 2 toxic tweets and 2 non-toxic tweets per keyword.

5.2. Setting

For classification, we use the CountVectorizer method from the scikit-learn library version 0.19[6] to create bag-of-word features and set the threshold to 10 for minimum document frequency. From the same library, we tuned hyper-parameters for the LogisticRegression method using the GridSearchCV method. We setup the hyper-parameters as follows.

1. C value: 0.001, 0.01, 0.1, 1, 10.
2. Fit intercept: True or False.
3. Penalty: L1 or L2.

Finally, our baseline is to set all predictions of toxic tweets according to the keyword-based approach, because all tweets contain toxic keywords.

[5] https://github.com/rkcosmos/deepcut

[6] https://github.com/scikit-learn/scikit-learn

Table 2: Classification result.

Method	Precision	Recall	F1-Score
Logistic Regression	0.87	0.70	0.78
Keyword Baseline	0.50	1.00	0.67

5.3. Results and Discussion

Table 2 shows the experimental results. The best accuracy is 0.80, when the hyper-parameters are C = 0.1, Fit intercept = True, and Penalty = L2. We obtained 9 false negatives and 26 false positives, as can be seen in Figure 3. Compared with the keyword baseline method, our classification results are better in terms of precision and F1-score.

Although the keyword-based approaches are popular for performing this type of classification, it failed to correctly classify some tweets, as in the following example, which is a Thai-English translated tweet: "Damn, just finished laundry and it's raining." In contrast, our approach correctly classified it as non-toxic.

Furthermore, in our approach, the primary reason for an error in the case of a false positive is complaining in a tweet, examples of which are given in Table 3. The cases of false negatives are primarily because of the following two reasons.

1. Tweets that contain both toxic words and positive words such as "good" or "beautiful."
2. Tweets that contain unknown or low document frequency words in our model.

The examples of false negatives are shown in Table 4.

Because our corpus is small, surface features are insufficient for abbreviation, slang, and unknown words; thus, we need to increase the size of our dictionary to let the model learn more words. In addition, we are aware that using only bag-of-word features is not sufficient for tweet classification; therefore, we will explore more efficient approaches in a future study.

Furthermore, we admit that the auto-segmentation is not perfect, which affects the classification. For example, a tweet that includes a wrong word segmentation like "อะอีดอก" gets incorrectly predicted as non-toxic. The right segmentation should be "อะ (affix) | อี (impolite prefix) | ดอก (bitch)" and with this, the prediction is toxic.

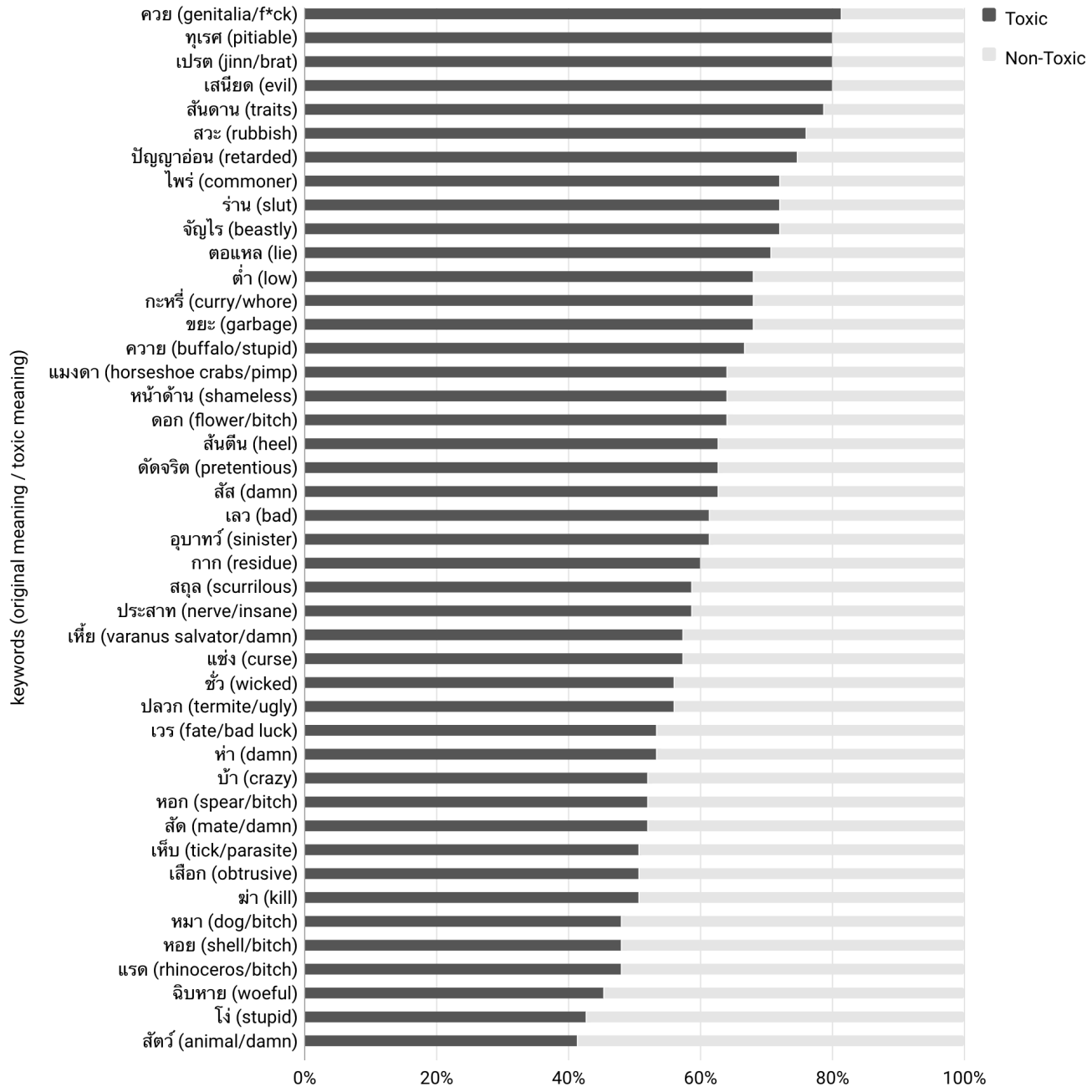

Figure 2: Distribution of toxic and non-toxic tweets based on keywords.

Despite some errors, our auto-segmentation method is considerably effective referring to the examples below.

(a) ถึงคุณรวยล้นฟ้าแต่ไร้น้ำใจก็ยากที่คนจะศรัทธา (Despite of being a millionaire, but without kindness, nobody will respect you.) which auto-segmentation and human-segmentation are same.
ถึง (to/although) | คุณ (you) | รวย (rich) | ล้น (overflow) | ฟ้า (sky) | แต่ (but) | ไร้ (without) | น้ำใจ (kindness) | ก็ (then) | ยาก (hard) | ที่ (at/that) | คน (person/people) | จะ (will) | ศรัทธา (faith).

(b) คนเห็นแก่ตัวที่ไม่เคยเห็นใจคนอื่น (A selfish person who never care for others.)
auto-segmentation: คน (person/people) | เห็น (see) | แก่ (for) | ตัว (self) | ที่ (at/that) | ไม่ (no) | เคย (ever) | เห็นใจ (sympathetic) | คน (person/people) | อื่น (another)
human-segmentation: คน (person/people) | เห็นแก่ตัว (selfish) | ที่ (at/that) | ไม่เคย (never) |

Table 3: Examples of false positives.

Tweet text (English translation)	Toxic keyword	True label	Predicted label
Since this morning, the dormitory internet is damn and even now, it is still damn.	damn	Non-toxic	Toxic
I want to shout f*ck but all I can say is yes sir.	f*ck	Non-toxic	Toxic

Table 4: Examples of false negatives.

Tweet text (English translation)	Toxic keyword	True label	Predicted label
You damn, Just go to die for better.	damn	Toxic	Non-toxic
Damn, you're annoying. You are just pretty but stupid.	damn, stupid	Toxic	Non-toxic

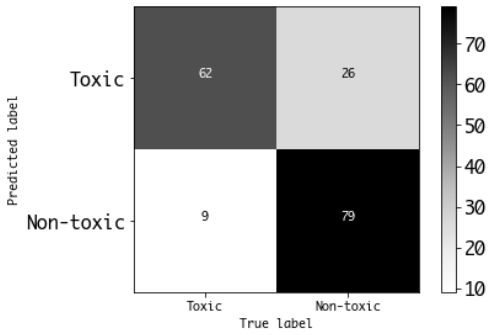

Figure 3: Confusion matrix of toxicity classification.

เห็นใจ (sympathetic) | คนอื่น (others)

6. Conclusions and Future work

With the increasing popularity of social media in Thailand, the growth of toxicity in online conversation is a growing concern. To the best of our knowledge, there is no public Thai resource related to online toxicity. In this study, we present toxicity annotation for a Thai Twitter Corpus along with a supervised classification method as a preliminary exploration for toxicity analysis in the Thai language.

In the future, we plan to not only enhance the classification method, but also improve our model and use streaming data for the dataset to eliminate bias involved with using keywords. Our improved model will be used to extend the volume of the Thai toxicity corpus.

Furthermore, aside from the corpus, we intend to increase, both, the size and content of our dictionary to include various other language entities, such as word variations and abbreviations by applying semantic orientation (Turney, 2002). Our dictionary will not only provide the English translation for Thai toxic words, but also examples for each word. We hope to enlarge our corpus with this new dictionary to make it a sufficient and reliable resource for Thai language analysis in the future. Finally, we might consider using other content such as re-tweets or previous conversations to provide a better understanding regarding the intentions of the messages in a future study.

7. Acknowledgements

This research was (partly) supported by Grant-in-Aid for Research on Priority Areas, Tokyo Metropolitan University, Research on social bigdata.

8. Bibliographical References

Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254.

Cheng, J., Bernstein, M., Danescu-Niculescu-Mizil, C., and Leskovec, J. (2017). Anyone Can Become a Troll: Causes of Trolling Behavior in Online Discussions. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 1217–1230, Portland, OR, USA, February. Association for Computing Machinery.

Cooper, D. (1996). Ambiguous (((Par(t)(it))((ion))(s))(in)) Thai Text. In *Proceedings of the 11th Pacific Asia Conference on Language, Information and Computation*, pages 109–118, Seoul, South Korea, December. Association for Computational Linguistics.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the 11th International Conference on Web and Social Media*, Montreal, Canada, May. Association for the Association for the Advancement of Artificial Intelligence.

Kennedy, G., McCollough, A., Dixon, E., Bastidas, A., Ryan, J., Loo, C., and Sahay, S. (2017). Technology Solutions to Combat Online Harassment. In *Proceedings of the First Workshop on Abusive Language Online*, pages 73–77, Vancouver, BC, Canada, August. Association for Computational Linguistics.

Kolhatkar, V. and Taboada, M. (2017). Constructive Language in News Comments. In *Proceedings of the First Workshop on Abusive Language Online*, pages 11–17, Vancouver, BC, Canada, August. Association for Computational Linguistics.

Kwak, H. and Blackburn, J. (2014). Linguistic Analysis of Toxic Behavior in an Online Video Game. In *Pro-*

ceedings of the 1st Exploration on Games and Gamers Workshop, EGG 2014.

Mekthawornwathana, T. (2011). The Factors used for Identifying "Politeness" in Male and Female Conversations among Thai Undergraduate Students. *NIDA Development Journal*, 51(3):142–166.

Nand, P., Perera, R., and Kasture, A. (2016). "How Bullying is this Message?": A Psychometric Thermometer for Bullying. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 695–706, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Orathai Chinakarapong. (2014). Conceptual Metaphor of Thai Curse Words. *Journal of Humanities Naresuan University*, 11(2):57–76, August.

Razavi, A., Inkpen, D., Uritsky, S., and Matwin, S. (2010). Offensive Language Detection Using Multi-level Classification. In *Proceedings of the 23rd Canadian conference on Advances in Artificial Intelligence*, pages 16–27, Ottawa, Canada, June. Canadian Conference on Artificial Intelligence 2010.

Saleem, H. M., Dillon, K. P., Benesch, S., and Ruths, D. (2016). A Web of Hate: Tackling Hateful Speech in Online Social Spaces. In *Proceedings of the First Workshop on Text Analytics for Cybersecurity and Online Safety (TA-COS 2016)*, Portorož, The Republic of Slovenia, May.

Turney, P. (2002). Thumbs Up or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Van Hee, C., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., De Pauw, G., Daelemans, W., and Hoste, V. (2015). Detection and Fine-grained Classification of Cyberbullying Events. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 672–680, Hissar, Bulgaria, September. Association for Computational Linguistics.

Wang, Y., Norcie, G., Komanduri, S., Acquisti, A., Leon, P. G., and Cranor, L. F. (2011). "I regretted the minute I pressed share": A Qualitative Study of Regrets on Facebook. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, page 10. Association for Computing Machinery.

Waseem, Z. and Hovy, D. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Student Research Workshop*, pages 88–93, San Diego, California, June. Association for Computational Linguistics.

Wulczyn, E., Thain, N., and Dixon, L. (2017). Ex Machina: Personal Attacks Seen at Scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399, Perth, Australia, April. International World Wide Web Conference 2017.

Yin, D., Xue, Z., Hong, L., Davison, B. D., Kontostathis, A., and Edwards, L. (2009). Detection of Harassment on Web 2.0. In *Proceedings of the Content Analysis in the WEB*, volume 2, pages 1–7, Madrid, Spain, April. International World Wide Web Conference 2009.

Zhou, N., Aw, A., Lertcheva, N., and Wang, X. (2016). A Word Labeling Approach to Thai Sentence Boundary Detection and Pos Tagging. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, pages 319–327, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Monitoring Targeted Hate in Online Environments

Tim Isbister[1], Magnus Sahlgren[1], Lisa Kaati[1], Milan Obaidi[2], Nazar Akrami[2]
[1]Swedish Defense Research Agency (FOI), [2]Uppsala University
[1]164 90 Stockholm, Sweden, [2]Box 256, 751 05 Uppsala, Sweden
{Tim.Isbister, Magnus.Sahlgren, Lisa.Kaati}@foi.se, {Milan.Obaidi, Nazar.Akrami}@psyk.uu.se

Abstract

Hateful comments, swearwords and sometimes even death threats are becoming a reality for many people today in online environments. This is especially true for journalists, politicians, artists, and other public figures. This paper describes how hate directed towards individuals can be measured in online environments using a simple dictionary-based approach. We present a case study on Swedish politicians, and use examples from this study to discuss shortcomings of the proposed dictionary-based approach. We also outline possibilities for potential refinements of the proposed approach.

1. Introduction

Digital environments provide an enormously large and accessible platform for people to express a broad range of behavior — perhaps even broader than what can be expressed in real world environments, due to the lack of social accountability in many digital environments. Hate and prejudice are examples of such behaviors that are overrepresented in digital environments. Hate messages in particular are quite common, and have increased significantly in recent years. In fact, many, if not most, digital newspapers have closed down the possibility to comment on articles since the commentary fields have been overflowing with hate messages and racist comments (Gardiner et al., 2016). To many journalists, politicians, artists, and other public figures, hate messages and threats have become a part of daily life. A recent study on Swedish journalists showed that almost 3 out of 4 journalists received threats and insulting comments through emails and social media (Nilsson, 2015).

Several attempts to automatically detect hate messages in online environments have been made. For example, Warner and Hirschberg (2012) use machine learning coupled with template-based features to detect hate speech in user-generated web content with promising results. Wester et al. (2016) examine the effects of various types of linguistic features for detecting threats of violence in a corpus of YouTube comments, and find promising results even using simple bag-of-words representations. On the other hand, Ross et al. (2016) examine the reliability of annotations of hate speech, and find that the annotator agreement is very low, indicating that hate speech detection is a very challenging problem. The authors suggest that hate speech should be seen as a continuous rather then as a binary problem, and that detailed instructions for the annotators are needed to improve the reliability of hate speech annotation. Waseem and Hovy (2016) examine the effect of various types of features on hate speech detection, and find that character n-grams and gender information provide the best results. Davidson et al. (2017) argues that lexical methods suffer from low precision and aims to separate hate speech from other instances of offensive language. Their results show that while racist and homophobic content are classified as hate speech, this is not the case for sexist content, which illustrates the challenge in separating hate speech from other instances of offensive language.

The apparent lack of consensus regarding the difficulty of the hate speech detection problem suggests that the problem of hate speech detection deserves further study. This paper contributes to the discussion in two ways. Firstly, we provide a psychological perspective on the concept of hate. Secondly, we present a study of the advantages and disadvantages of using the arguably simplest possible approach to hate speech detection: that of counting occurrences of keywords based on dictionaries of terms related to hate speech. The main goal of this paper is to provide a critical discussion about the possibility of monitoring targeted hate in online environments.

This paper is outlined as follows. Section 2 discusses the psychological aspects of hate and how hate messages can have various level of severity. Section 3 presents a dictionary-based approach to measure hate directed towards individuals. Section 4 provides a case study where we analyze hate speech targeted towards 23 Swedish politicians on immigration-critical websites, and discuss challenges and directions for future work. Finally, Section 5 provides some concluding remarks.

2. On hate

In the psychological literature hate is thought to be a combination to two components: one cognitive and one emotional (Sternberg and Sternberg, 2008). The cognitive component can be threat perceptions caused for example by out-group members, but it can also involve devaluation or a negative view of others. The emotional component on the other hand involves emotions such as contempt, disgust, fear, and anger that are generally evoked by the cognitive component. Defined in this way, hates shares much with prejudice, which is defined as negative evaluations or devaluations of others based on their group membership. Like hate, prejudice is argued to be consisting of a cognitive component (stereotypes about others), an emotional component (dislike of others) and a behavioral component (acting in accordance with the emotional and cognitive component (Allport, 1954)). Hate, like prejudice, functions as the motivational force when people behave in harmful ways toward others.

Category	Sample terms (ENG)	Sample terms (SWE)	Normalized frequency per category
Swearword	fuck, shit, gay	fan, skit, bög	0.00137
Anger	is crazy, idiot, enemy	är galen, idiot, fiende	0.00106
Naughtiness	clown, is an idiot, stupid	clown, är en idiot, knäpp	0.00076
General threat	kidnap, be followed, hunt	kidnappa, bör förföljas, jaga	0.00068
Death threat	should be killed, ruin, bomb	borde dödas, utrota, bomba	0.00031
Sexism	whore, bitch, should be raped	hora, subban, borde våldtas	0.00005

Table 1: Different categories of hate with representative terms and normalized frequency.

Hate is commonly directed toward individuals and groups but it is also expressed toward other targets in the social world. For example, it is common that hate is expressed toward concepts (e.g. communism) or countries (e.g. USA). It is important to note however that there is some disagreement about not only the definition but also the behavioral outcomes of hate. For example, while some see hate leading to behavioral tendencies such as withdrawal caused by disgust or fear, others see hate as the manifestation of anger or rage, which lead one to approach, or attack, the object of hate (Edward et al., 2005).

Dealing with digital environments, the disagreement about behavioral tendencies might seem less relevant. Specifically, withdrawal caused by disgust or fear in the real world is not the same in digital environment where withdrawal would not be necessary — or approach would not be a direct threat to wellbeing. Acknowledging the disagreements noted above, we aim to examine hate messages with various level of severity varying between swearwords directed to individuals to outright death threats.

3. Monitoring hate

This work focuses on detecting hate messages and expressions directed towards individuals. The messages can have various level of severity with respect to individual integrity and individual differences in perception of threat. More specifically, we examine six different categories: anger, naughtiness, swearwords, general threats, and death threats. While the two categories naughtiness and anger may overlap in some aspects, they were aimed to capture different expressions and causes of hate speech, with naughtiness indicating to the speaker's tendency to misbehave and generally express naughtiness toward others, and anger being an emotional state triggered by something in the surrounding and leading to the expression of anger (and/or naughtiness) towards a person. We also include sexism (degradation of women), since it is commonly used for devaluative purposes. Each category is represented by a dictionary of terms, as exemplified in Table 1. Our study focuses on Swedish data, but to ease understanding we have translated some of the words to English. Note that the dictionaries may contain both unigrams and multiword expressions.

The dictionaries are constructed in a manner similar to Tulkens et al. (2016b; 2016a); human experts (psychologist and computer scientist) manually study a large number of posts from the text domain of interest (see further Section 4.1.) and record significant words and phrases. In order to improve the recall of the dictionaries, a word embedding is then used to suggest other relevant terms to the experts. This is done by simply computing the 15 nearest neighbors in the embedding space to each term in the dictionaries. For each term suggestion, the expert has the choice to either include or reject the term suggestion. We note that it is also possible to cast the term suggestion task as an active learning problem, in which a classifier is iteratively refined to identify useful term suggestions based on the expert's feedback (Gyllensten and Sahlgren, 2018).

As embedding, we use Gensim's (Řehůřek and Sojka, 2010) implementation of the Continuous Bag of Words (CBOW) model (Mikolov et al., 2013), which builds word vectors by training a 2-layer neural network to predict a target word based on a set of context words. The network learns two sets of vectors, one for the target terms (the embedding vectors), and one for context terms. The objective of the network is to learn vectors such that their dot product correspond to the log likelihood of observing word pairs in the training data. We use default parameters for the embeddings, with a window size set to 5. The embeddings are trained on a collection of immigration-critical websites, further discussed in Section 4.1.. Note that the embedding method does not handle multiword units in any special way; if multiword units are to be included in the analysis, they need to be incorporated in the data as a preprocessing step.

The expanded dictionaries are used to detect and monitor hate by simple frequency counting; if a term from one of the dictionaries occurs in the vicinity of a mention of a target individual, we increment the count for that category. This is arguably the simplest possible approach to hate speech monitoring, and many types of refinements are possible, such as weighting of the dictionary entries (Eisenstein, 2017), handling of negation (Reitan et al., 2015), and scope detection. We will return to a more detailed discussion of problems with the proposed approach in Section 4.3.. At this point, we note that one possible advantage of using such a simple approach is its transparency; it is easy to understand a simple frequency counter for a non-technical end user.

Of course, transparency and comprehensibility are useless if the method generates an excessive amount of false positives. The only way for us to control the precision of the frequency counting is to delimit the context within which occurrences of dictionary terms are counted; a narrow context window spanning something like one to three words around a target individual's name will reduce the probability that a term from one of the dictionaries refers to something other than the target name. In the following case study, we opt for the most conservative approach and use a context of only one term on each side of the target name.

Website	# comments	# words
avpixlat.info	2 904 933	99 472 281
nordfront.se	89 495	3 125 218
nyatider.nu	2 176	124 949
motgift.nu	1 380	68 992
nordiskungdom.com	117	6 530

Table 2: The websites included in our study.

4. Case study

To exemplify the dictionary-based approach, we have examined the expression of the different categories of hate toward 23 national-level politicians (10 males and 13 females). Studying national-level politicians in Sweden is timely as we are approaching the Swedish parliament election in September 2018. There have also been recent alarms on politicians threatening to leave politics because of an increasing amount of hate being expressed in recent years. Our analyses are based on text from commentary fields on immigration critical websites from September 2014 to December 2017. The time period was chosen to cover a single electoral period in the Swedish parliament.

As target names, we use the full names of the politicians. This is obviously a crude simplification that severely affects the recall of the approach, since people are often referred to using only their first name, a pronoun, or, in the data we studied, some negative nickname or slur. As an example, the Swedish prime minister, Stefan Löfven, is often referred to in online discussions as "svetsarn" (the welder), or using negative nicknames such as "Röfven", which is a paraphrase of "röven" (in English "the ass").

4.1. Data

In Sweden, as well as in several other European countries, there has been a recent surge in activity and formation of movements that are critical of immigration. These immigration-critical groups show a high interactivity on social media and on websites. In Sweden, there are several digital immigration-critical milieus with a similar structure: articles published by editorial staff and user-generated comments. The commentary fields are not moderated, which makes the comments an important scene to express hate toward journalists, politicians, artists, and other public figures. The comment section allows readers to respond to an editorial article instantly. The editorial articles generally focuses on topics such as crimes, migration, politics and societal issues. The websites that we have studied are listed in Table 2. For each website, we have downloaded all comments between 2014/09/01 to 2017/10/01. Note that the embeddings used for term suggestions are also trained on this data.

4.2. Results

Table 3 shows the how many times each minister is mentioned in the comments with his or hers full name during the given time period. Obviously, the Prime Minister Stefan Löfven is the most frequently mentioned politician, with more than 10,000 mentions during the analyzed period. The second most mentioned politician in the studies data is Morgan Johansson, the Swedish Minister of Justice and Home Affairs, and the third most mentioned minister is Margot Wallström, Minister for Foreign Affairs.

Name	Mentions
Stefan Löfven	10 663
Morgan Johansson	3 142
Margot Wallström	2 681
Magdalena Andersson	1 931
Ylva Johansson	1 524
Gustav Fridolin	1 113
Alice Bah Kuhnke	567
Peter Eriksson	248
Peter Hultqvist	228
Isabella Lövin	184
Mikael Damberg	169
Ardalan Shekarabi	158
Åsa Regnér	136
Ann Linde	128
Annika Strandhäll	98
Ibrahim Baylan	61
Per Bolund	48
Anna Ekström	36
Heléne Fritzon	36
Helene Hellmark Knutsson	14
Karolina Skog	11
Sven-Erik Bucht	8

Table 3: Number of times each Swedish minister is mentioned in the comments during the time period.

Figure 1 (next page) shows the amount of hate towards the Swedish ministers. The left figure shows simple frequency counts of hate terms in the immediate vicinity of each target name, while the right figure shows the proportions of targeted hate toward the Swedish ministers, calculated as the frequency of each hate category in the context of each politician, divided by the total number of mentions for that politician. In both figures, it is obvious that naughtiness (in purple) is the most frequent category for the politicians as a group, followed by anger (in red), swearwords (in yellow) and general threat (in gray). We do not see any sexism and no explicit death threats in our data, most likely due to the very narrow context used in these experiments.

Figure 1 shows that the most frequently mentioned ministers are also those who receive the most hate in the data we have studied. However, when looking at the proportions of hateful comments for each minister, we see that the most mentioned politician (Stefan Löfven) is not the minister with the proportionally most hateful comments. This is instead Mikael Damberg, the Minister for Enterprise and Innovation. However, Damberg is only mentioned 169 times in the data, and a mere 1.18% of these contain hate; that is, only 2 mentions of 169. It is a similar situation with Ann Linde, the Minister for EU Affairs and Trade, who has the proportionally most general threats in her mentions, but this is based on only 1 mention out of 128. Isabella Lövin, the Minister for International Development Cooperation, is the target of the proportionally most naughtiness, but also in this case, this is only 1 mention out of 184.

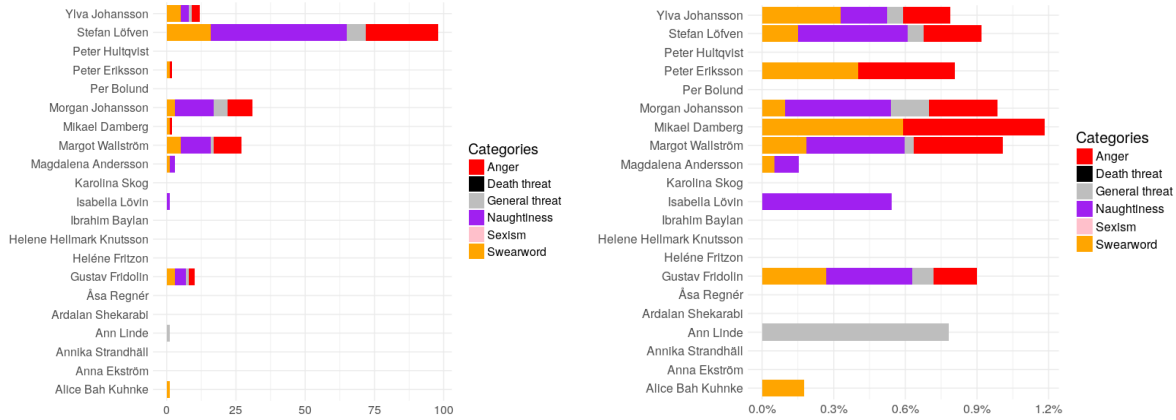

Figure 1: Amount of hate contexts for Swedish ministers (using only the preceding and succeeding terms). The left figure shows simple frequency counts of hate terms, while the right figure shows proportions (i.e. counts divided by the total number of mentions).

4.3. Discussion

The results in Figure 1 demonstrate that even with such a simple and naïve method as the one used in this paper, it is possible to do a general and rudimentary form of threat assessment based on mentions in social media data. The method is sufficiently simple to be adaptable to many different scenarios, and sufficiently transparent for end-users to understand. However, we do pay a price for the simplicity.

As we noted in the last section, expressions of hate seem to correlate with frequency of mention (at least in the data we have studied). This makes the left part of Figure 1 less interesting. On the other hand, counting proportions, as we do in the right part of the figure, risks overestimating the significance of very rare events. A perhaps more useful measure might be to calculate deviations from the expected amount of hateful comments for each minister. As an example, Morgan Johansson is mentioned 3 142 by his full name in our data. Based on the normalized category frequencies in Table 3, we should expect that 4 of these mentions contain swearwords, 3 contain anger, 2 contain naughtiness, and 2 contain general threat. Looking at the actual frequency counts, we see that 3 mentions contain swearwords, 8 contain anger, 14 contain naughtiness, and 5 contain general threat. For the last three categories, the actual counts are much higher than would be expected, indicating that these are significant measurements.

Table 4 (next page) shows the deviations from expected counts per category for each minister. The deviation is computed as the actual counts minus the expected counts:

$$\#(m,c) - \left(\frac{\#(c)}{T} \cdot \#(m)\right) \quad (1)$$

where $\#(m,c)$ is the actual co-occurrence count of a minister and a category, $\frac{\#(c)}{T}$ is the relative frequency of a category in the data $\#(c)$ is the frequency of the category and T is the total number of words in the data), and $\#(m)$ is the frequency of mention of a minister.

This is a obviously a severely oversimplified probabilistic model, but it does provide useful information. We note that the columns for death threats and sexism only contain negative or zero values, which indicates that no significant death threats or sexism is being expressed towards the ministers in the data. Two ministers have higher general threats than can be expected, and a few more have higher swearwords and anger, but the deviations for these categories in our data are not very large. The highest deviation in our study is the naughtiness category for the prime minister, which indicates that he is the subject of a significant amount of negative remarks in the data we have studied. Another potentially interesting observandum is the combination of categories that have positive deviations for the different ministers. To take two examples, Morgan Johansson has positive deviations for anger, naughtiness and general threat, while Ylva Johansson has positive deviations for swearwords, anger and naughtiness. One might hypothesize that the combination of anger and general threat deserves more attention than the combination of swearwords and naughtiness.

The perhaps most obvious drawback of the approach used in this paper is that it will only detect hate in direct relation to a full name, but not in relation to pronouns or slang expressions referring to the person in question; i.e. the approach suffers from a lack of coreference resolution. This will obviously affect the recall of the method, which is a serious shortcoming that risks missing critical mentions. In the present analysis, we have no idea whether the lack of death threats in our results is due to an actual absence of death threats in the data, or whether it is due to omissions in the analysis.

Although we delimit the context as much as possible to only include the preceding and succeeding terms, our results are still affected by false positives. There are three basic error types for false positives in our analysis. One is negated statements, such as (hate term in boldface):

Person	Swearword	Anger	Naughtiness	General threat	Death threat	Sexism
Stefan Löfven	0.98	3.29	**16.49**	−2.65	−3.15	−0.46
Morgan Johansson	−1.16	2.82	2.77	2.32	−0.93	−0.14
Margot Wallström	1.5	2.32	3.12	−1.41	−0.79	−0.12
Magdalena Andersson	−1.56	−1.96	0.63	−1.03	−0.57	−0.08
Ylva Johansson	2.95	1.43	1.9	−0.83	−0.46	−0.07
Gustav Fridolin	1.51	−0.14	2.2	−0.6	−0.33	−0.05
Alice Bah Kuhnke	0.24	−0.58	−0.4	−0.3	−0.17	−0.02
Peter Eriksson	0.67	0.74	−0.18	−0.13	−0.08	−0.01
Peter Hultqvist	−0.29	−0.22	−0.15	−0.12	−0.06	−0.01
Isabella Lövin	−0.24	−0.18	0.87	−0.1	−0.05	−0.01
Mikael Damberg	0.77	0.83	−0.12	−0.09	−0.05	−0.01
Ardalan Shekarabi	−0.21	−0.16	−0.11	−0.08	−0.05	−0.01
Åsa Regnér	−0.18	−0.14	−0.1	−0.07	−0.04	−0.01
Ann Linde	−0.17	−0.13	−0.09	0.93	−0.04	−0.01
Annika Strandhäll	−0.13	−0.1	−0.07	−0.05	−0.03	0
Ibrahim Baylan	−0.08	−0.06	−0.04	−0.03	−0.02	0
Per Bolund	−0.06	−0.05	−0.03	−0.02	−0.01	0
Anna Ekström	−0.05	−0.04	−0.03	−0.02	−0.01	0
Heléne Fritzon	−0.01	−0.01	−0.01	−0.01	0	0
Helene Hellmark Knutsson	−0.02	−0.01	−0.01	−0.01	0	0
Karolina Skog	−0.01	−0.01	−0.01	−0.01	0	0

Table 4: Deviation from expected counts per category for each minister. Positive scores indicate that the actual count is higher than the expected count.

`jag tror inte Stefan Lövfen är dum`
(I don't think Stefan Lövfen **is stupid**)

Handling negations is a well-known issue in both information retrieval and sentiment analysis, and one could think of several different ways to deal with negations. The perhaps most simple method is to use a *skip* or *flip* function that skips a sequence of text when having encountered a negation, or simply flips the sentiment of the negated text (Choi and Cardie, 2009). It is of course also necessary to determine the scope of the negation, which is a non-trivial problem in itself (Lazib et al., 2016).

Another error type in our analysis is quotes, such as:

`vi har varit naiva [sa] Stefan Löfven`
(we have been **naive** [said] Stefan Löfven)

The "said" is implicit, and is signaled by quotation marks and punctuation in the original data. However, when using aggressive tokenization, such punctuation is normally removed, which leads to the above type of errors. Retaining punctuation would obviously be one way to prevent such errors. Another possibility is to use a dependency parse of the data, which would rearrange the context according to the dependency structure. "Naive" would then be closer to "we" than to "Stefan Löfven".

A third error type that is related to the previous one is misinterpreting (or ignoring) the semantic roles of the proposition. Consider the following examples:

`låt regeringen med Stefan Lövfen hota med nyval`
(let the government with Stefan Lövfen **threaten** with new election)

`vi skiter i om du blir förbannad Stefan Lövfen`
(we don't care if you get **upset** Stefan Lövfen)

Stefan Löfven is not the target of hate in neither of these cases. Instead, he (or in the first case, he and the Swedish government) is the *agent* of the predicates "threatened" vs. "upset". In order to resolve agency of the predicates, we would need to do semantic role labeling, which assigns a semantic role to each participant of a proposition. Identifying the agent of the predicate becomes even more important when increasing the context size, since it will also increase the number of false positives when only counting occurrences of hate terms.

5. Conclusion

In this paper, we have aimed to measure how online hate is directed toward national-level politicians in Sweden. This is an important and timely endeavor because the expression of online hate is becoming increasingly pervasive in online forums, especially toward this specific group. The expression of hate has shown to have downstream consequences not only for individuals who are targeted, but also for our democratic society and core liberal values. Recent studies show that the frequent exposure to hate speeches does not only lead to increased devaluation and prejudice (Soral et al., 2017), but may also increase dehumanization of the targeted group (Fasoli et al., 2016). Dehumanization in return makes the targeted groups or individuals seem less than human, legitimizing and increasing the likelihood of violence (Rai et al., 2017). Moreover, online hate does not only play a significant role in shaping people's attitudes and beliefs toward certain groups, but it also have far-reaching consequences for societies in general, such as increasing

tendency to violating social norms and threatening democratic core values.

As we mentioned in the introduction, many digital newspapers in Sweden and other countries have closed down the possibility to comment on articles due to the degree of hate expressed by some users. This is a clear example of how online hate restricts and threatens one of the core values of democracy. That is the freedom to express your views and opinions. To prevent such harmful effects it is important to monitor and measure how and toward whom hate is expressed online.

The second aim of this study was to address some of the gaps in the field. As noted in the introduction, the contemporary approaches to measuring online hate are marked by the apparent lack of consensus regarding the difficulty of the hate speech detection. The approach for monitoring targeted hate that we have described in this work is a simple yet powerful way to understand hate messages directed toward individuals. The strength of this method lies in its simplicity and transparency, and perhaps also for having more conservative criteria that reduces the number of false positives. We have also identified a number of ways to improve the method, including the use of **coreference resolution**, handling of **negation**, context refinement using **dependency parsing**, and agency detection using **semantic role labeling**.

The trade-off between complexity and performance, and between recall and precision, are challenging dilemmas for law enforcement and other end users of hate monitoring tools. Acknowledging these dilemmas, future improvements of hate monitoring should be directed toward the optimal cut-off where usefulness for law enforcement can meet ease of conduct when it comes to analyzing data.

6. References

Allport, G. (1954). *The Nature of Prejudice*. Reading, MA: Addison-Wesley.

Choi, Y. and Cardie, C. (2009). Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of EMNLP*, EMNLP '09, pages 590–598, Stroudsburg, PA, USA. Association for Computational Linguistics.

Davidson, T., Warmsley, D., Macy, M. W., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International Conference on Web and Social Media*, pages 512–515.

Edward, B., McCauley, C., and Rosin, P. (2005). From plato to putnam: Four ways to think about hate. in the psychology of hate. pages 3–36.

Eisenstein, J. (2017). Unsupervised learning for lexicon-based classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3188–3194.

Fasoli, F., Paladino, M., Carnaghi, A., Jetten, J., Bastian, B., and Bain, P. (2016). Not "just words": Exposure to homophobic epithets leads to dehumanizing and physical distancing from gay men. *European Journal of Social Psychology*, 46:237–248.

Gardiner, B., Mansfield, M., Anderson, I., Holder, J., Louter, D., and Ulmanu, M. (2016). The web we want: The dark side of guardian comments. *The Guardian*, 12, April.

Gyllensten, A. C. and Sahlgren, M. (2018). Distributional term set expansion. In *Accepted for publication in the Proceedings of LREC 2018*.

Lazib, L., Zhao, Y., Qin, B., and Liu, T. (2016). Negation scope detection with recurrent neural networks models in review texts. In Wanxiang Che, et al., editors, *Social Computing*, pages 494–508, Singapore. Springer Singapore.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Nilsson, M. L. (2015). Hot och hat mot svenska journalister. *Nordicom-information*, 37(3-4):31–56.

Rai, T., Valdesolo, P., and Graham, J. (2017). Dehumanization increases instrumental violence, but not moral violence. *Pnas*, 114(32):8511–8516.

Řehůřek, R. and Sojka, P. (2010). 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. ELRA.

Reitan, J., Faret, J., Gambäck, B., and Bungum, L. (2015). Negation scope detection for twitter sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 99–108, Lisboa, Portugal, September.

Ross, B., Rist, M., Carbonell, G., Cabrera, B., Kurowsky, N., and Wojatzki, M. (2016). Measuring the reliability of hate speech annotations: The case of the european refugee crisis. In *Proceedings of NLP4CMC III*. Bochumer Linguistische Arbeitsberichte.

Soral, W., Bilewicz, M., and Winiewski, M. (2017). Exposure to hate speech increases prejudice through desensitization. *Aggressive Behaviour*.

Sternberg, R. and Sternberg, K. (2008). *The nature of hate*. New York: Cambridge University Press.

Tulkens, S., Hilte, L., Lodewyckx, E., Verhoeven, B., and Daelemans, W. (2016a). The automated detection of racist discourse in dutch social media. *Computational Linguistics in the Netherlands Journal*, 6:3–20, 12/2016.

Tulkens, S., Hilte, L., Lodewyckx, E., Verhoeven, B., and Daelemans, W. (2016b). A dictionary-based approach to racism detection in dutch social media. In *First Workshop on Text Analytics for Cybersecurity and Online Safety (TA-COS 2016)*.

Warner, W. and Hirschberg, J. (2012). Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, LSM '12, pages 19–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Waseem, Z. and Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016*, pages 88–93.

Wester, A., Øvrelid, L., Velldal, E., and Hammer, H. L.

(2016). Threat detection in online discussions. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 66–71, San Diego, USA.

Think Before Your Click: Data and Models for Adult Content in Arabic Twitter

Ali Alshehri[1], **El Moatez Billah Nagoudi**[2], **Hassan Alhuzali**[3], **Muhammad Abdul-Mageed**[3]
[1] SUNY at Buffalo
[2] Laboratoire d'Informatique et de Mathmatique LIM, Amar Telidji University
[3] Natural Langauge Processing Lab, The University of British Columbia
alimoham@buffalo.edu, e.nagoudi@lagh-univ.dz, halhuzali@alumni.ubc.ca, muhammad.mageeed@ubc.ca

Abstract

Given the widespread use of social media and their increasingly impactful role in our lives today, there is a pressing need to ensure their safety of use. In particular, various social groups view the spread of adult content in social networks as undesirable. This content may even pose a serious threat to other vulnerable groups (e.g. children). In this work, we develop a unique, large-scale dataset of adult content in Arabic Twitter and provide in-depth analyses of the data. The dataset enables us to study the scope and distribution of adult content in the Arabic version of the network, thus possibly uncovering target phic locales. In addition, computationally exploit the data to learn a large lexicon specific to the topic and detect spreaders of adult content on the microblogging platform. Our models achieve promising results, reaching 79% accuracy on the task (24% higher than a competitive baseline).

1. Introduction

Social media continues to play an increasingly important role in our lives, making it necessary to keep these platforms safe and free from 'undesirable' content. Undesirable postings come in many forms, including deceptive (Westerman et al., 2014), hateful (Williams and Burnap, 2015), abusive (Mubarak et al., 2017), dangerous (Fuchs, 2017; Sikkens et al., 2017), and adult content (Abozinadah, 2015). Identification of spreaders of unsolicited content is beneficial not only for user satisfaction, but also for the safety of individuals and communities alike.

In the Arab world, social media are widely used (Lenze, 2017). This is especially the case for the Twitter platform where, according to some estimates (Salem, 2017), the number of monthly active users was expected to be 11.1 million as of March 2017. These Arab users send 27.4 million tweets per day, almost doubling up from 5.8 million in 2014 (Salem, 2017). Twitter has also been a very influential tool in the Arab world, as is evident from its role in the waves of uprisings the region. In the contexts of the political and social transformations the Arab world has witnessed, activists have heavily used the platform for disseminating views antagonistic to several Arab governments (Khondker, 2011; Gerbaudo, 2012). Similarly, governments themselves are increasingly using Twitter to spread content supporting their causes (i.e., propaganda) (Mejova, 2017).

Twitter prohibits the promotion of adult or sexual products, services, and content, whether in images, videos, or text. [1]However, spreaders of undesirable content are exploiting Twitter's popularity, and it is not uncommon to even witness advertising and adult content hashtags trending (Herzallah et al., 2017).

Popular search engines such as Google and Yahoo provide "safe search" options to filter out unwanted content. Social media platforms (e.g. Twitter, Facebook, YouTube) also offer similar options, yet seem to be fighting a more difficult battle. Efforts to combat unsolicited content, however, does not seem to be very successful thus far, as we will show. Depending on manually curated lists of words for use in filtering out adult content is no longer sufficient since language and techniques employed by spreaders of these content are constantly evolving. For example, spreaders of adult content often intentionally employ misspelled and/or slang words. Misspellings can be as simple as replacing the letter 'o' with the digit '0' in a word, which can enable these users to bypass Twitter's algorithmic filters.

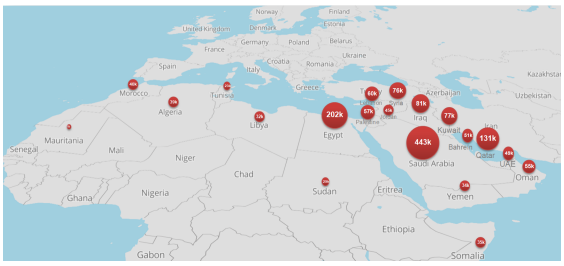

Figure 1: Geographical distribution of adult content in the Arab world.

Filtering out adult content is perhaps especially valuable in the Arab world, due to religious and cultural sensitivities. In this work, we seek to alleviate this bottleneck for Arabic social media. We make the following contributions: (1) we build a large-scale dataset of Arabic adult content; (2) we learn large-scale lexica (based on hashtags, unigrams, and bigrams) correlated with adult content from the data; (3) we perform an in-depth analysis of the data, thus affording a better understanding of the dynamics of adult content sharing and the behavior of its users on Twitter; and (4) we develop successful predictive models for detecting spreaders of adult content.

The remainder of the paper is organized as follows: In Section 2, we review related literature. We describe our dataset in Section 3, we perform several textual analyses of the data and describe learning a lexicon of adult content in Section 4. In Section 5, we describe our models for detecting adult content. Section 6 concludes the paper with our main find-

[1] https://support.twitter.com/articles/20170427?lang=en

ings.

2. Related work

Unsolicited Content on Twitter. Undesirable content can be prevalent in Twitter. The network is indeed vulnerable to misuse through posting of undesirable content such as spams, racist content, hateful speech, threats, and adult content. This is due to the fact that creating and maintaining an account on Twitter is fairly easy. Unlike Facebook, where anonymity is at least theoretically not possible, anonymity is easier on Twitter. This possibly translates to more undesirable content. The work of Grier et al. (2010) is relevant to the scope of unsolicited or spam content on Twitter. The authors studied 25 million URLs posted on Twitter and found that 8% of content in these URLs are spam. Analyzing the click-through rate of those spam tweets, they found that around 0.13% of them generate a site visit. This rate is much higher than the click-through rate reported for spam emails (Kanich et al., 2008). This implies that the number of spammers on Twitter is increasing over time.

Racist and Hateful Speech. A number of studies have attempted to investigate racists and hateful speech in the web as well as Twitter. For example, Burnap and Williams (2014) look at the manifestation and diffusion of hate speech and antagonistic content in social media in relation to events that could be classified as 'trigger' events for hate crimes. Their dataset consists of $450k$ tweets collected a two weeks window in the immediate aftermath of Drummer Lee Rigbys murder in Woolwich, UK. Using n-gram and type-dependency features, they implemented probabilistic, rulebased, and spatial classifiers. The authors reported a best F-score of 0.77 using the probabilistic classifier. Similarly, Davidson et al. (2017) created a hate speech lexicon based on a list of phrases and words provided by *Hatebase.org*. Using this list, they crawled a set of $85m$ tweets containing terms from the lexicon. Then, a random set of $25k$ tweets were manually annotated by CrowdFlower users on three categories: hate speech, offensive, and neither. They used Logistic Regression and a dictionary to construct a predictive hate and offensive language model, which achieved an F1-score of 90%.

Adult Content. Some studies were also devoted to investigating and detecting adult content online. For example, Coletto et al. (2016) analyzed 169 million data points on Tumblr and Flickr and found that although the community of adult content producers is small, adult content is spread widely in the networks. While producers of adult content are clustered in semi-isolated communities on these platforms, they are linked with the rest of the network with a very high number of what Coletto et al. (2016) called "consumers" (users who do not post new adult content but follow producers of such content, share and like their posts). The authors maintained that, due to the fact that users in the network are enabled to see what other users 're-post' or 'like,' over a quarter of the all Tumblr users were unintentionally exposed to adult content. The case is no different in Twitter where users are able to see recently liked tweets by users they follow. Singh et al. (2016) estimated at least 10 million accounts tweeting and spreading adult content according as of May 2015.

Singh et al. (2016) employ graph- and content-based features extracted from $74k$ tweets posted by $18k$ Twitter users on the same task, reporting 91.96% accuracy. Their analysis shows that adult content users fulfill the characteristics of spammers as stated by the rules and guidelines of Twitter[2]. These pioneering works, however, focused on detecting adult or spam content, without providing analyses of the content itself. Our work exploits a much bigger dataset (e.g., our dataset is about eight times bigger than (Abozinadah, 2015)), and pays attention to especially the geographical distribution of targets of the adult content.

Twitter Spam. What increases Twitter users's exposure to pornographic tweets is also the fact that trending hashtags are usually exploited by spammers (Abozinadah, 2015; El-Mawass and Alaboodi, 2016). This vulnerability of Twitter users has recently led to a number of studies focusing on analyzing and detecting Twitter 'spams' (e.g. (Lin and Huang, 2013; Yang et al., 2013; Wahsheh et al., 2012b; Wahsheh et al., 2013; Herzallah et al., 2017; Chu et al., 2012; Grier et al., 2010; El-Mawass and Alaboodi, 2016; Singh et al., 2016)). A few of these studies were dedicated to spam detection in Arabic social media (e.g. (Wahsheh et al., 2012a; Wahsheh et al., 2012b)).

Adult Content in Arabic. Early work on Arabic social media has focused on developing corpora and systems for detecting sentiment (Abdul-Mageed and Diab, 2012; Abdul-Mageed and Diab, 2011; Abdul-Mageed et al., 2014), aided by automatic processing tools developed for the language like ASMA (Abdul-Mageed et al., 2013), and later emotion (Abdul-Mageed et al., 2016). More related to our work is research by Abozinadah (2015) and Singh et al. (2016) who focused on detecting adult content on Arabic and English Twitter, respectively. Abozinadah (2015) and Abozinadah and Jones (2017) built a dataset of $1,000,300$ tweets comprising the most recent 50 tweets of 255 users as well as the most recent 50 tweets of users in their network. The authors then develop a machine learning classifier using different feature sets. They found that lexical features yield the best performance. As feature input to their classifiers, the authors extracted basic statistical measures from each tweet (e.g., average, minimum, maximum, standard deviation, and the total number of URLs, hashtags, picture, mentions, and characters). They reported 96% accuracy of adult content detection.

3. Dataset

We collect a large dataset of tweets with adult content. In addition, we identify a large network of adult content producers (who are also usually spreaders). We explain our data collection methods in terms of the following steps[3]:

1. **Hashtag seeds:** We start by collecting a list of hashtags[4]. associated with adult content by manually in-

[2] https://support.twitter.com/articles/64986.

[3] Due to the nature of this work, in various places of the paper, we provide examples that involve language that are related to adult content. Although we use academic norms to present the content in appropriate way, reader discretion is advised.

[4] This list can be downloaded from: https://goo.gl/Qcc1wW.

specting several relevant tweets. We iteratively expand the list by adding co-occurring hashtags that clearly communicate adult content. Our final list is composed of 100 hashtags that we manually judge as highly connected to adult content. Example hashtags from this list include سكس (Eng. "sex"), مولعه (Eng. "horny"), and مومس (Eng. "prostitute").

2. **Tweet-level dataset:** We use both the Twitter rest and streaming APIs to crawl tweets employing items from this list of 100 manually developed hashtags described above. Using these crawlers, we acquired a dataset of $\sim$ 27 million tweets. We refer to this dataset as **`main`**. After filtering out retweets and duplicates, we ended up with a total of $200K$ tweets. We refer to this dataset as **`unique`**.

3. **User-level dataset:** We extract all the users who posted one or more of the tweets in the **`main`** dataset and acquire a total of $20,621$ users. We then crawl the timelines of these users, possibly fetching up to $3,200$ tweets from each user. We are able to obtain the timelines of $11,648$ of these users, making the total number of tweets from these timelines around 8.6 million. We could not fetch the tweets of the remaining $8,973$ users for a number of reasons: First, $2,456$ users were suspended during the period between crawling the **`main`** dataset and the timelines. These users represents $\sim$ 11% of all users. Second, 629 users were not found at the time of user data crawling at all. These users most likely have deleted their accounts. The remaining $5,888$ users were found active, but our crawlers failed to fetch their data due either to (a) their accounts being protected[5] or (b) have no tweets at the time of crawling. We call this dataset **`timelines`**. See Table 1 for a summary of the datasets and Table2 for a summary of users in our datasets.

Dataset	Size (tweets)
Main	27 M
Unique	200 K
Timelines	8.6 M

Table 1: Datasets in the study. **Main:** All the tweets we have initially crawled. **Unique:** Tweets from main after deduplication and removal of retweets. **Timelines:** Tweets from our list of unique list of $11,648$ users' timelines.

4. Understanding Adult Content

We use our dataset as a basis for understanding adult content in various ways. First, we build lexica of adult content in the form of hashtags and n-grams (unigrams and bigrams). These can provide a summary of what the involved lexical content is like, but can also be used for collecting adult content in the future for building classifiers. Related results are presented in Section 4.1. Second, we study the posting behaviors of adult content users by aggregating important frequencies from their content. We also present a description of their network structure via simple follower-followee statistics (Section 4.2.). The types of media employed in adult content is another significant aspect of sharing pornography online and hence we also study this aspect of content in Section 4.4. Adult content users also seem to have specific practices as to choosing their screen names on the network. In an attempt to understand these practices, we analyze a sample from our data in Section 4.3. Finally, a question that arises is related to the locales this particular type of business might be targeting and/or most thriving in. In Section 4.5., we perform an analysis that answers this exact question. We now turn to describing our findings related to each of these user and content attributes.

Type of user	Freq.	%
Active (collected)	$11,648$	56.5%
Active (not collected)	$5,888$	28.5%
Suspended	$2,456$	11.9%
Not found	629	3.1%

Table 2: Types, counts, and percentages of users in our **`timelines`** datasets.

4.1. Lexica of Adult Content

4.1.1. Hashtags

We extract all the hashtags with frequency $>$ 20 in the dataset, acquiring a total of $21,907$ hashtags. A sample from the extracted hashtags is in Table 3. The range of hashtags are related to descriptions of explicit content that may be accessible via a shared URL in a tweet, a range of pornographic activities, and references to individuals with different sexual orientations. The lexicon can be used as a basis for monitoring online adult content and collecting even larger data for detecting pornography.

4.1.2. N-grams

We also extract all unigrams and bigrams with frequencies $>$ 20 from the dataset, acquiring a total of $128,625$ unigrams and $243,953$ bigrams. Table 3 shows a sample of each of these types[6]. Similar to the hashtag lexicon, the N-gram lexicon exposes a range of activities related to adult content, but also clickbait where users are asked to click on a link to watch adult video or see an explicit photo. This clearly paints a picture of adult content marketing as a business.

4.2. User Timelines

For a deeper understanding of the behaviour of adult content spreaders, we calculate several measures based on our **`timelines`** dataset. These measures include the average, median, and mode of (1) total tweets posted per user, (2) total pornographic hashtags employed by a user, (3) avergae

[5] Protected users can only be crawled when the authenticated user crawling the data either "owns" the timeline or is an approved follower of the owner. None of these applied to us.

[6] The lists of all hashtags, unigrams and bigrams with their frequencies can be downloaded from: https://goo.gl/LVig9g.

Hashtag		Uingram		Bigram	
AR	**EN**	**AR**	**EN**	**AR**	**EN**
#سكس	#sex	هنا	here	هنا #سكس	#sex here
#ز*ك	#f*ck	ز*ك	f*ch	الفيلم كامل	full movie
#فحل	#bull	سكس	sex	شاهد وحمل	watch and download
#ممحونه	#divorced	الفيلم	movie	الفيلم هنا	movie here
#ديوث	#cuckold	اضغط	click	كامل هنا	full here
#محارم	#incest	شاهد	watch	ثم اضغط	then click
#افلام_سكس	#sex_movies	خاص	private	#روابط_سكس #سكس	#sex #sex_links
#ز*	#pe*is	كامل	full	اضغط الرابط	ckick the link
#سالب	#bottom	الرابط	link	اضغط على	click on
#ق*ه	#b*tch	ين*ك	fu*king	#سكس #ز*ك	#sex #fu*k

Table 3: A Sample of our Adult Content Lexica. Hashtags (left), unigrams (middle), and bigrams (right).

hashtags used per tweet, and (4) number of friends and followers per user. As Table 4 shows, an average adult content user posts ~ 914 tweets, uses 1.45 hashtags per tweets, and has $\sim 7,489$ friends and 850 followers in their network. These statistics show that spreaders of adult content not only employ hashtags as a mechanism of reaching wider audiences, but also as a way to adhere to Twitter regulation about pornographic content. The analysis also reveals that these users are not silos in the network, but rather have friends and followers.

	Mean	**Median**	**Mode**
Total tweets	914.20	235	10
Total hashtags used	1,370.91	525.50	28
Hashtags per tweet	1.45	0.35	0
Friends	7,488.70	252	0
Followers	850.30	72	0

Table 4: Descriptive statistics of adult content and user network in our data.

4.3. Screen Names Analysis

We wish to investigate screen names used by adult content users. To do so, we first randomly sampled 100 adult users and manually analyzed their screen names. We found out a number of interesting patterns. As shown in Table 5, the most common screen name pattern consists of one or more (e.g., age, physical) attributes. For example, in عشريني وسيم (EN: "a handsome twenties aged guy") there are two adjectives describing both the age and physical attributes of the user. For another example, in المتغطرس (EN: "the arrogant one"), the user chooses to describe his psychological attributes that imply power and pride. In addition, about 60% of those include more pronounced physical attributes with clear sexual meanings and an indication of user gender. Examples include جادة ممحونة (EN: "horny and serious female"), مربرب مشعر (EN: "chubby and hairy male"), and فحل عنيف (EN: "violent and potent male"). Other common screen names are person names, some of which also contain attributes such as أمل ك* مفتوح (EN: "Amal open vag*na") and مجودي بوث (EN: "Majoodi bisexual"). It is also not uncommon for screen names to have city or country names such as سالب مصرى القاهره (EN: "Egyptian bottom Cairo") and سالب الرياض (EN: "bottom from Riyadh"). Some users use their email, phone, or social media account addresses as their screen names. Finally, some screen names do not seem to follow any specific patterns. Instead, they contain numbers, commas, underscores, symbols or mixture of these without any apparent meaning such as '-' and '//-//'. To further analyze adult users screen names, we extract unigrams, bigrams and emoji from all screen names. Table 6 provides a list of the top 10 unigrams, bigrams, and emoji employed by these users. It is clear from the Table that adult content users tend to employ screen names with sexual connotations. We also investigated which exact language is used in screen names. We found that about 66% of these names consist of either Arabic alphabet exclusively or a mixture of Arabic and Roman alphabet. About 29% employ Roman alphabet only. The rest 5% consists of emojis, numbers, symbols, or/and alphabet other than Arabic and Roman.

4.4. Tweet media

We also analyze the use of media in the tweets posted by adult content spreaders. This helps us answer questions like: "What is the rate of tweets that contain URLs?" and "Which is the most URL type (web page, photo or videos) used?". Table 7 summarizes the results of this analysis. We have noticed that many of the adult content tweets contain links, many of which do not actually lead to what they are advertised to be, specifically adult content (%59.68), but rather other sites but such as news sites or ones related to health and beauty content (e.g., http://healthwabeauty.com/). Interestingly, some links lead to blogs that do not seem to originate from the Arab world. For example, the blog

Type	percentage	Example	English
Attribute	34%	فحل عنيف	Violent and potent
Attribute + city/country	9%	سالب الرياض	Botton (in) Riyadh
Email address	2%	a-sa**@**.com	–
Emoji	19%	👅🔥	–
Hashtag	1%	#مقاطع	#clips
Person name	25%	خالد	Khalid
Person name + attribute	5%	مجودي بوث	Majoodi bisexual
Others	19%	//-//	–

Table 5: Types of screen names in a sample of 100 pornographic users

Uingram	EN	Bigram	EN	Emoji
سكس	Sex	ك* مفتوح	Open pus*y (unvirgin)	🔥
مطلقه	Divorced (F)	طي* كبيرة	Big As*	🔞
مفتوح	Open	افلام سكس	Sex movies	🖤
متحرره	Emancipated (F)	من المغرب	From Morocco	✨
هايجة	Horny (F)	سكس محارم	Incest sex	👅
مولعة	Horny (F)	سكس عربي	Arab sex	❤️
فحل	potent (M)	سكس في	Sex in	🫧
طي*	As*	مقاطع سكس	Sex clips	🌈
كبيرة	Big (F)	سكس فون	Phone sex	💋
افلام	Movies	وسيط زواج	Marriage broker	💘

Table 6: Top 10 unigram, bigram, and emojis in screen names used by users (**F:** female; **M:** male).

	Count	%
Web link URLs	6.754M	59.68%
URLs refer to photo	3.166M	27.98%
URLs refer to video	1.310M	11.57%
URLs refer to animated gif	86.973M	0.77%
Total URLs (web link+media)	**11.318M**	**100%**

Table 7: Types of media in tweet URLs in the data.

at https://ecoinsnews.blogspot.com/ focuses on Bitcoin and the encryption market mostly likely directed to English speaking-audience. We also observed that only a small fraction of these sites are ones that solicit subscriptions for one or another of a sex 'service' or sexual content.

4.5. Geographical Distribution

Using our dataset, we analyze the geographical distribution of adult content across the Arab world. For the purpose, we follow a simple method:

1. Initially, we automatically generate a list of Arab countries and cities (we refer to the list as **autocities**) from Google map API[7]. The **autocities** list pertains 22 countries and has a total of 361 cities. **autocities** had several errors (e.g., names in English and Hebrew, neighborhood names instead of the a specific city name, GPS coordinates cities).
2. For this reason, we manually correct this list using the following procedure: For each country in the **autocities**, we keep only Arabic city names and

Country	Freq.	City	Freq.
KSA	443,112	Riyadh	89,232
Egypt	202,795	Jeddah	66,944
Qatar	131,707	Amman	27,651
Iraq	81,517	Makkah	16,133
Kuwait	81,517	Qassim	14,344
Syria	76,948	Dammam	14,251
Lebanon	76,290	Madinah	10,365
Palestine	57,029	Jerusalem	9,345
Oman	55,735	Tabuk	8,690
Bahrain	51,956	Gaza	8,256

Table 8: Top 10 Arab countries and cities matched in the adult content.

[7] https://developers.google.com/maps/?hl=fr.

				regular_content			adult_content		
BOW	**#data_points**	**acc**	**avg-f**	**prec**	**rec**	**f**	**prec**	**rec**	**f**
	10	0.54	0.42	0.64	0.07	0.12	0.54	0.97	0.69
	50	0.54	0.41	0.77	0.04	0.08	0.54	0.99	0.70
	100	0.55	0.43	0.83	0.06	0.12	0.54	0.99	0.70
	250	0.53	0.38	0.50	0.01	0.02	0.53	0.99	0.69
	500	0.53	0.38	1.00	0.01	0.02	0.53	1.00	0.69
				regular_content			**adult_content**		
BOM	**#data_points**	**acc**	**avg-f**	**prec**	**rec**	**f**	**prec**	**rec**	**f**
	10	0.76	0.76	0.69	0.92	0.79	0.90	0.63	0.74
	50	0.77	0.77	0.69	0.94	0.80	0.92	0.63	0.75
	100	0.78	0.78	0.70	0.94	0.80	0.92	0.64	0.76
	250	**0.79**	**0.78**	**0.70**	**0.93**	**0.80**	**0.92**	**0.65**	**0.76**
	500	0.78	0.78	0.70	0.94	0.80	0.92	0.64	0.76

Table 9: Results from our models for detecting spreaders of adult content on Twitter. We use SVMs in our experiments. **BOW:** bag-of-words models. **BOF:** bag-of-means models.

manually add other cities (replacing, e.g., the English and Hebrew names with Arabic counterparts, and substituting GPS co-oridnates with corresponding cities). For this step, we use Wikipedia[8]. We also search Wikipedia for Arabic city names that are not in the original **`autocities`** list and add cities we find. The new list covers 22 countries and a total of 488 cities. We call this list **`goldcities`**[9].

3. Finally, we use **`goldcities`** to identify the names of countries and cities targeted in the adult dataset, based on simple matching between our goldcities and tweets' unigrams. This allows for identifying the most targeted Arab countries and cities by adult content users. Figure 1 maps the geographical distribution of targets in adult content by country. Table 8 shows the top 10 Arab countries as well as top 10 cities matched in the data. The top two countries are *KSA*[10] and *Egypt*. The city list in Table 8 contains "Qassim" which is a KSA province rather than a city. Observably, 7 cities out of the 10 top mentioned cities are KSA cities. This shows very heavy targeting of KSA cities. The findings about KSA and Egypt is not surprising as these two countries have large Twitter populations, although there may be other reasons these countries are targeted most. Any such potential reasons are outside the scope of our current work, but form the basis of important research questions.

5. Classification

We build supervised models for detecting adult users exploiting the data of these users. For the purpose, we identify $2,500$ users in the adult data such that each has at least 500 tweets. For the negative class (i.e., regular users), we use an equal number of users' data where each user has at least 500 tweets.

5.1. Pre-processing, Data splits, and settings

We randomize the user data from both the positive and the negative classes and remove all the hashtag seeds used to collect the data. For this work, we choose our hyper-parameters beforehand from a small set of choices as we describe next. To facilitate replication and future work under more sophisticated conditions, we split the data into 80% training, 10% development, and 10 % testing so that development data can be used to tune parameters with more advanced experiments. We employ simple SVM classifiers with a fixed vocabulary size of $20K$ words, under two classification conditions:

Bag-of-Words: Where each vector simply represents each word existing in a tweets with a binary value (0 or 1).

Bag-of-Means: We build a word embedding model (Mikolov et al., 2013) exploiting a large in-house dataset of Arabic tweets totaling $> 100m$ data points. For this purpose, we adopt the pre-processing pipeline of (Zahran et al., 2015; Abdul-Mageed et al., 2018), in that we remove any non-unicode characters, normalize *Alif maksura* to *Ya*, reduce all *hamzated Alif* to plain *Alif*, remove all non-Arabic characters. To clean noise, we reduce all letter repetition of > 2 characters to only 2. We build a skip-gram model with 300 dimensions, a minimal word count $= 100$ words, and a window size of 5 words on each side of a target word. For vectorization, we average the word vectors of each tweet, acquiring a 300-dimension bag of means for each data point.

Settings: We develop the classifiers under a number of conditions, pertaining the number of tweets exploited from each user. We use numbers of tweets according to values from the set {*10, 50, 100, 250, 500*}. For these simple classifiers, we use the **`scikit learn`** [11] SVC implementation.

5.2. Evaluation:

We report in terms of accuracy (acc), precision (prec), recall (rec), and F-score (f). We use a random baseline of 50%, which is also equal to each of the two classes in the data,

[8] https://en.wikipedia.org/wiki/Arab_world.

[9] The **`goldcities`** list can be downloaded from: https://goo.gl/s3xzpB

[10] Kingdom of Saudi Arabia

[11] http://scikit-learn.org/stable.

given that the two classes are balanced. We first performed the experiments on both Dev and Test under the same conditions, but only report on Test here. As mentioned earlier, we choose to set aside a development set for future replicability and comparisons under more sophisticated experimental conditions.

Table9 presents the results of our model. As the Table shows, the **BOM** conditions perform better, with best accuracy reaching 79% with 250 tweets, significantly (i.e., $p< 0.03$) exceeding the random baseline of 50%. The best **BOM** (250 tweets) classifier reaches 92% of precision on the adult/positive class, with a reasonable recall of 65%. These results show the utility of the simple SVM **BOM** classifier on this task, as opposed to a **BOW**. Even with 10 tweets, the **BOM** classifier performs at 76% acc, reaching a high precision of 90% on the adult users class.

6. Conclusion

In this work, we described a method for collecting a large-scale dataset of adult content in Arabic Twitter. We also described the data we acquired using this method and used the data to understand the tweeting behavior in this safety-important area of online behavior. We also extracted three lexica involving hashtags, unigrams, and bigrams, which we also make available to the community. Analyzing our data also gave us an opportunity to identify the geographical distribution of targets of adult content, which may lead to future important discoveries about the dynamics and market of adult content production and spread. We finally developed simple, yet quite successful, models for detecting spreaders of adult contents on the microblogging platform. Our models achieve 79% accuracy on the task. In the future, we plan to improve our classification models and further investigate the network structure of the adult content spreaders.

7. Acknowledgement

This research was enabled in part by support provided by WestGrid (`https://www.westgrid.ca/`) and Compute Canada (`www.computecanada.ca`).

8. Bibliographical References

Abdul-Mageed, M. and Diab, M. T. (2011). Subjectivity and sentiment annotation of modern standard arabic newswire. In *Proceedings of the 5th LAW*, pages 110–118. ACL.

Abdul-Mageed, M. and Diab, M. T. (2012). Awatif: A multi-genre corpus for modern standard arabic subjectivity and sentiment analysis. In *LREC*, pages 3907–3914.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2013). Asma: A system for automatic segmentation and morpho-syntactic disambiguation of modern standard arabic. In *Proceedings of RANLP 2013*, pages 1–8.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2014). Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Abdul-Mageed, M., AlHuzli, H., and DuaaAbu Elhija, M. D. (2016). Dina: A multi-dialect dataset for arabic emotion analysis. In *The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media*, page 29.

Abdul-Mageed, M., Alhuzali, H., and Elaraby, M. (2018). You tweet what you speak: A city-level dataset of arabic dialects. In *LREC*.

Abozinadah, E. A. and Jones, Jr., J. H. (2017). A statistical learning approach to detect abusive twitter accounts. In *Proceedings of the International Conference on Compute and Data Analysis*, ICCDA '17, pages 6–13, New York, NY, USA. ACM.

Abozinadah, A., M. A. a. J. J. (2015). Detection of abusive accounts with arabic tweets. *International Journal of Knowledge Engineering*, Vol. 1, No. 2.

Burnap, P. and Williams, M. L. (2014). Hate speech, machine classification and statistical modelling of information flows on twitter: Interpretation and communication for policy decision making.

Chu, Z., Widjaja, I., and Wang, H., (2012). *Detecting Social Spam Campaigns on Twitter*, pages 455–472. Springer Berlin Heidelberg, Berlin, Heidelberg.

Coletto, M., Aiello, L. M., Lucchese, C., and Silvestri, F. (2016). Pornography consumption in social media. *CoRR*, abs/1612.08157.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*.

El-Mawass, N. and Alaboodi, S. (2016). Detecting arabic spammers and content polluters on twitter. In *2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC)*, pages 53–58, April.

Fuchs, C. (2017). *Social media: A critical introduction*. Sage.

Gerbaudo, P. (2012). *Tweets and the streets: Social media and contemporary activism*. Pluto Press.

Grier, C., Thomas, K., Paxson, V., and Zhang, M. (2010). @spam: The underground on 140 characters or less. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 27–37, New York, NY, USA. ACM.

Herzallah, W., Faris, H., and Adwan, O. (2017). Feature engineering for detecting spammers on twitter: Modelling and analysis. *Journal of Information Science*, page 0165551516684296.

Kanich, C., Kreibich, C., Levchenko, K., Enright, B., Voelker, G., Paxsonf, V., and Savage, S. (2008). Spamalytics: An empirical analysis of spam marketing conversion. In *Proceedings of the ACM Conference on Computer and Communications Security*.

Khondker, H. H. (2011). Role of the new media in the arab spring. *Globalizations*, 8(5):675–679.

Lenze, N. (2017). Social media in the arab world: Communication and public opinion in the gulf states. *European Journal of Communication*, 32(1):77–79.

Lin, P.-C. and Huang, P.-M. (2013). A study of effective features for detecting long-surviving twitter spam accounts. In *2013 15th International Conference on*

Advanced Communications Technology (ICACT), pages 841–846, Jan.

Mejova, Y. (2017). Seminar users in the arabic twitter sphere. In *Social Informatics: 9th International Conference, SocInfo 2017, Oxford, UK, September 13-15, 2017, Proceedings*, volume 10539, page 91. Springer.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mubarak, H., Darwish, K., and Magdy, W. (2017). Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

Salem, F. (2017). The arab social media report 2017: Social media and the internet of things: Towards data-driven policymaking in the arab world. Vol. 7.

Sikkens, E., van San, M., Sieckelinck, S., Boeije, H., and de Winter, M. (2017). Participant recruitment through social media: Lessons learned from a qualitative radicalization study using facebook. *Field Methods*, 29(2):130–139.

Singh, M., Bansal, D., and Sofat, S. (2016). Behavioral analysis and classification of spammers distributing pornographic content in social media. *Social Network Analysis and Mining*, 6(1):41, Jun.

Wahsheh, H., Alsmadi, I., and Al-Kabi, M. (2012a). Analyzing the popular words to evaluate spam in arabic web pages. *IJJ: The Research Bulletin of JORDAN ACM–ISWSA*, 2(2):22–26.

Wahsheh, H. A., Al-kabi, M. N., and Alsmadi, I. M. (2012b). Evaluating arabic spam classifiers using link analysis. In *Proceedings of the 3rd International Conference on Information and Communication Systems*, ICICS '12, pages 12:1–12:5, New York, NY, USA. ACM.

Wahsheh, H. A., Al-Kabi, M. N., and Alsmadi, I. M. (2013). Spar: A system to detect spam in arabic opinions. In *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Dec.

Westerman, D., Spence, P. R., and Van Der Heide, B. (2014). Social media as information source: Recency of updates and credibility of information. *Journal of Computer-Mediated Communication*, 19(2):171–183.

Williams, M. L. and Burnap, P. (2015). Cyberhate on social media in the aftermath of woolwich: A case study in computational criminology and big data. *British Journal of Criminology*, 56(2):211–238.

Yang, C., Harkreader, R., and Gu, G. (2013). Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Transactions on Information Forensics and Security*, 8, Aug.

Zahran, M. A., Magooda, A., Mahgoub, A. Y., Raafat, H. M., Rashwan, M., and Atyia, A. (2015). Word representations in vector space and their applications for arabic. In *CICLing (1)*, pages 430–443.

Automated email Generation for Targeted Attacks using Natural Language

Avisha Das, Rakesh Verma
Department of Computer Science
University of Houston, Houston, Texas
{adas5, rverma}@uh.edu

Abstract

With an increasing number of malicious attacks, the number of people and organizations falling prey to social engineering attacks is proliferating. Despite considerable research in mitigation systems, attackers continually improve their modus operandi by using sophisticated machine learning, natural language processing techniques with an intent to launch successful targeted attacks aimed at deceiving detection mechanisms as well as the victims. We propose a system for advanced email masquerading attacks using Natural Language Generation (NLG) techniques. Using legitimate as well as an influx of varying malicious content, the proposed deep learning system generates *fake* emails with malicious content, customized depending on the attacker's intent. The system leverages Recurrent Neural Networks (RNNs) for automated text generation. We also focus on the performance of the generated emails in defeating statistical detectors, and compare and analyze the emails using a proposed baseline.

Keywords: natural language generation, email masquerading, deep learning

1. Introduction

The continuous adversarial growth and learning has been one of the major challenges in the field of Cybersecurity. With the immense boom in usage and adaptation of the Internet, staggering numbers of individuals and organizations have fallen prey to targeted attacks like phishing and pharming. Such attacks result in digital identity theft causing personal and financial losses to unknowing victims. Over the past decade, researchers have proposed a wide variety of detection methods to counter such attacks (e.g., see (Verma and Hossain, 2013; Thakur and Verma, 2014; Verma and Dyer, 2015; Verma and Rai, 2015; Verma and Das, 2017), and references cited therein). However, wrongdoers have exploited cyber resources to launch newer and sophisticated attacks to evade machine and human supervision. Detection systems and algorithms are commonly trained on historical data and attack patterns. Innovative attack vectors can trick these pre-trained detection and classification techniques and cause harm to the victims.

Email is a common attack vector used by phishers that can be embedded with poisonous links to malicious websites, malign attachments like malware executables, etc (Drake et al., 2004). Anti-Phishing Working Group (APWG) has identified a total of 121,860 unique phishing email reports in March 2017. In 2016, APWG received over 1,313,771 unique phishing complaints. According to sources in IRS Return Integrity Compliance Services, around 870 organizations had received W-2 based phishing scams in the first quarter of 2017, which has increased significantly from 100 organizations in 2016. And the phishing scenario keeps getting worse as attackers use more intelligent and sophisticated ways of scamming victims.

Fraudulent emails targeted towards the victim may be constructed using a variety of techniques fine-tuned to create the perfect deception. While manually fine-tuning such emails guarantees a higher probability of a successful attack, it requires a considerable amount of time. Phishers are always looking for automated means for launching fast and effective attack vectors. Some of these techniques include bulk mailing or spamming, including action words and links in a phishing email, etc. But these can be easily classified as positive warnings owing to improved statistical detection models.

Email masquerading is also a popular cyberattack technique where a phisher or scammer after gaining access to an individual's email inbox or outbox can study the nature/content of the emails sent or received by the target. He can then synthesize targeted malicious emails masqueraded as a benign email by incorporating features observed in the target's emails. The chances of such an attack being detected by an automated pre-trained classifier is reduced. The malicious email remain undetected, thereby increasing the chances of a successful attack.

Current Natural Language Generation (NLG) techniques have allowed researchers to generate natural language text based on a given context. Highly sophisticated and trained NLG systems can involve text generation based on predefined grammar like the Dada Engine (Baki et al., 2017) or leverage deep learning neural networks like RNN (Yao et al., 2017) for generating text. Such an approach essentially facilitates the machine to learn a model that emulates the input to the system. The system can then be made to generate text that closely resembles the input structure and form.

Such NLG systems can therefore become dangerous tools in the hands of phishers. Advanced deep learning neural networks (DNNs) can be effectively used to generate coherent sequences of text when trained on suitable textual content. Researchers have used such systems for generating textual content across a wide variety of genres - from tweets (Sidhaye and Cheung, 2015) to poetry (Ghazvininejad et al., 2016). Thus we can assume it is not long before phishers and spammers can use email datasets - legitimate and malicious - in conjunction with DNNs to generate deceptive malicious emails. By masquerading the properties of a legitimate email, such carefully crafted emails can deceive pre-trained email detectors, thus making people and organizations vulnerable to phishing scams.

In this paper, we address the new class of attacks based on

automated fake email generation. We start off by demonstrating the practical usage of DNNs for fake email generation and walk through a process of fine-tuning the system, varying a set of parameters that control the content and intent of the text. The key contributions of this paper are:

1. A study of the feasibility and effectiveness of deep learning techniques in email generation.
2. Demonstration of an automated system for generation of 'fake' targeted emails with a *malicious* intent.
3. Fine-tuning synthetic email content depending on training data - intent and content parameter tuning.
4. Comparison with a baseline - synthetic emails generated by Dada engine (Baki et al., 2017).
5. Detection of synthetic emails using a statistical detector and investigation of effectiveness in tricking an existing spam email classifier (built using SVM).

2. Related Works

Phishing detection is one of the widely researched areas of cybersecurity. Despite the development of a large number of phishing detection tools, many victims are still falling prey to these attacks. Researchers in (Drake et al., 2004) explicitly break down the structure of a phishing email, describing in detail the *modus operandi* of a phisher or scammer. In this section, we review previous research in areas of text generation using natural language and the use of deep learning in generation of phishing based attacks and detection.

Textual Content Generation. Natural language generation techniques have been widely popular for synthesizing unique pieces of textual content. NLG techniques proposed by (Reiter and Dale, 2000; Turner et al., 2010) rely on templates pre-constructed for specific purposes. The fake email generation system in (Baki et al., 2017) uses a set of manually constructed rules to pre-define the structure of the fake emails. Recent advancements in deep learning networks have paved the pathway for generating creative as well as objective textual content with the right amount of text data for training. RNN-based language models have been widely used to generate a wide range of genres like poetry (Ghazvininejad et al., 2016; Xie et al., 2017), fake reviews (Yao et al., 2017), tweets (Sidhaye and Cheung, 2015), geographical information (Turner et al., 2010) and many more.

The system used for synthesizing emails in this work is somewhat aligned along the lines of the methodology described in (Chen and Rudnicky, 2014a; Chen and Rudnicky, 2014b). However, our proposed system has no manual labor involved and with some level of post processing has been shown to deceive an automated supervised classification system.

Phishing email Detection. In this paper, we focus primarily on generation of fake emails specifically engineered for phishing and scamming victims. Additionally, we also look at some state-of-the-art phishing email detection systems. Researchers in (Basnet et al., 2008) extract a large number of text body, URL and HTML features from emails, which are then fed into supervised (SVMs, Neural Networks) as well as unsupervised (K-Means clustering) algorithms for the final verdict on the email nature. The system proposed in (Chandrasekaran et al., 2006) extracts 25 stylistic and structural features from emails, which are given to a supervised SVM for analysis of email nature. Newer techniques for phishing email detection based on textual content analysis have been proposed in (Verma et al., 2012; Verma and Hossain, 2013; Verma and Aassal, 2017; Yu et al., 2009). Masquerade attacks are generated by the system proposed in (Baki et al., 2017), which tunes the generated emails based on legitimate content and style of a famous personality. Moreover, this technique can be exploited by phishers for launching email masquerade attacks, therefore making such a system extremely dangerous.

3. Experimental Methodology

The section has been divided into four subsections. We describe the nature and source of the training and evaluation data in Section 3.1. The pre-processing steps are demonstrated in Section 3.2. The system setup and experimental settings have been described in Section 3.3.

3.1. Data description

To best emulate a benign email, a text generator must learn the text representation in actual legitimate emails. Therefore, it is necessary to incorporate benign emails in training the model. However, as a successful attacker, our main aim is to create the perfect deceptive email - one which despite having malign components like poisoned links or attachments, looks legitimate enough to bypass statistical detectors and human supervision.

Primarily, for the reasons stated above, we have used multiple email datasets, belonging to both legitimate and malicious classes, for training the system model and also in the quantitative evaluation and comparison steps. For our training model, we use a *larger ratio* of malicious emails compared to legitimate data (approximate ratio of benign to malicious is 1:4).

Legitimate dataset. We use three sets of legitimate emails for modeling our legitimate content. The legitimate emails were primarily extracted from the outbox and inbox of real individuals. Thus the text contains a lot of named entities belonging to PERSON, LOC and ORGANIZATION types. The emails have been extracted from three different sources stated below:

- 48 emails sent by Sarah Palin (**Source 1**) and 55 from Hillary Clinton (**Source 2**) obtained from the archives released in (The New York Times, 2011; WikiLeaks, 2016) respectively.
- 500 emails from the Sent items folder of the employees from the Enron email corpus (**Source 3**) (Enron Corpus, 2015).

Malicious dataset. The malicious dataset was difficult to acquire. We used two malicious sources of data mentioned below:

- 197 Phishing emails collected by the second author - called Verma phish below.

- 3392 Phishing emails from Jose Nazario's Phishing corpus [1] (Source 2)

Evaluation dataset. We compared our system's output against a small set of automatically generated emails provided by the authors of (Baki et al., 2017). The provided set consists of 12 emails automatically generated using the Dada Engine and manually generated grammar rules. The set consists of 6 emails masquerading as Hillary Clinton emails and 6 emails masquerading as emails from Sarah Palin.
Tables 1 and 2 describe some statistical details about the legitimate and malicious datasets used in this system. We define length (L) as the number of words in the body of an email. We define Vocabulary (V) as the number of unique words in an email.

Dataset	Count	Avg. L	Avg. V
Clinton	48	32	21
Palin	55	33	26
Enron	500	91	53
Total	603	81	48

Table 1: Legitimate Data Statistics

Dataset	Count	Avg. L	Avg. V
Verma Phish	197	153	99
Nazario Phish	3392	210	129
Total	3589	207	127

Table 2: Phishing Data Statistics

A few observations from the datasets above: the malicious content is relatively more verbose than than the legitimate counterparts. Moreover, the size of the malicious data is comparatively higher compared to the legitimate content.

3.2. Data Filtering and Preprocessing

We considered some important steps for preprocessing the important textual content in the data. Below are the common preprocessing steps applied to the data:

- Removal of special characters like @, #, $, % as well as common punctuations from the email body.
- emails usually have other URLs or email IDs. These can *pollute* and confuse the learning model as to what are the more important words in the text. Therefore, we replaced the URLs and the email addresses with the <LINK> and <EID> tags respectively.
- Replacement of named entities with the <NET> tag. We use Python NLTK NER for identification of the named entities.

On close inspection of the training data, we found that the phishing emails had incoherent HTML content which can pollute the training model. Therefore, from the original data (in Table 2), we carefully filter out the emails that were not in English, and the ones that had all the text data was embedded in HTML. These emails usually had a lot of random character strings - thus the learning model could be *polluted* with such random text. Only the phishing emails in our datasets had such issues. Table 3 gives the details about the filtered phishing dataset.

Dataset	Count	Avg. L	Avg. V
Verma Phish	127	50	34
Nazario Phish	2148	115	71
Total	2275	112	70

Table 3: Phishing Data Statistics after filtering step

3.3. Experimental Setup

We use a deep learning framework for the Natural Language Generation model. The system used for learning the email model is developed using Tensorflow 1.3.0 and Python 3.5. This section provides a background on a Recurrent Neural Network for text generation.
Deep Neural Networks are complex models for computation with deeply connected networks of neurons to solve complicated machine learning tasks. Recurrent Neural Networks (RNNs) are a type of deep learning networks better suited for sequential data. RNNs can be used to learn character and word sequences from natural language text (used for training). The RNN system used in this paper is capable of generating text by varying levels of granularity, i.e. at the character level or word level. For our training and evaluation, we make use of Word-based RNNs since previous text generation systems (Xie et al., 2017), (Henderson et al., 2014) have generated coherent and readable content using word-level models. A comparison between Character-based and Word-based LSTMs in (Xie et al., 2017) proved that for a sample of generated text sequence, word level models have lower perplexity than character level deep learners. This is because the character-based text generators suffer from spelling errors and incoherent text fragments.

3.3.1. RNN architecture

Traditional language models like N-grams are limited by the history or the sequence of the textual content that these models are able to look back upon while training. However, RNNs are able to retain the long term information provided by some text sequence, making it work as a "memory"-based model. However while building a model, RNNs are not the best performers when it comes to preserving long term dependencies. For this reason we use Long Short Term Memory architectures (LSTM) networks which are able to learn a better language/text representation for longer sequences of text.
We experiment with a few combinations for the hyper-parameters- number of RNN nodes, number of layers, epochs and time steps were chosen empirically. The input text content needs to be fed into our RNN in the form of word embeddings. The system was built using 2 hidden LSTM layers and each LSTM cell has 512 nodes. The input data is split into mini batches of 10 and trained for

[1] `http://monkey.org/~jose/wiki/doku.php` (2004), Deprecated now

100 epochs with a learning rate of 2×10^{-3}. The sequence length was selected as 20. We use $cross - entropy$ or $softmax$ optimization technique (Goodfellow et al., 2016) to compute the training loss, $Adam$ optimization technique (Kingma and Ba, 2014) is used for updating weights. The system was trained on an Amazon Web Services EC2 Deep Learning instance using an Nvidia Tesla K80 GPU. The training takes about 4 hours.

3.3.2. Text Generation and Sampling

The trained model is used to generate the email body based on the nature of the input. We varied the sampling technique of generating the new characters for the text generation.

Generation phase. Feeding a word ($\widehat{w_0}$) into the trained LSTM network model, will output the word most likely to occur after $\widehat{w_0}$ as $\widehat{w_1}$ depending on $P(\widehat{w_1} \mid \widehat{w_0})$. If we want to generate a text body of n words, we feed $\widehat{w_1}$ to the RNN model and get the next word by evaluating $P(\widehat{w_2} \mid \widehat{w_0}, \widehat{w_1})$. This is done repeatedly to generate a text sequence with n words: $\widehat{w_0}$, $\widehat{w_1}$, $\widehat{w_2}$, ..., $\widehat{w_n}$.

Sampling parameters. We vary our sampling parameters to generate the email body samples. For our implementation, we choose *temperature* as the best parameter. Given a sequence of words for training, w_0, w_1, w_2, ..., w_n, the goal of the trained LSTM network is to predict the best set of words that follow the training sequence as the output ($\widehat{w_0}$, $\widehat{w_1}$, $\widehat{w_2}$, ..., $\widehat{w_n}$).

Based on the input set of words, the model builds a probability distribution $P(w_{t+1} \mid w_{t' \leq t}) = softmax(\widehat{w_t})$, here $softmax$ normalization with *temperature* control (Temp) is defined as:

$$P(softmax(\widehat{w_t^j})) = \frac{K(\widehat{w_t^j}, Temp)}{\sum_{j=1}^{n} K(\widehat{w_t^j}, Temp)}, \quad \text{where}$$

$$K(\widehat{w_t^j}, Temp) = e^{\frac{\widehat{w_t^j}}{Temp}}$$

The novelty or eccentricity of the RNN text generative model can be evaluated by varying the Temperature parameter between $0 < Temp. \leq 1.0$ to generate samples of text (the maximum value is 1.0). We vary the nature of the model's predictions using two main mechanisms - deterministic and stochastic. Lower values of $Temp.$ generates relatively deterministic samples while higher values can make the process more stochastic. Both the mechanisms suffer from issues, deterministic samples can suffer from repetitive text while the samples generated using the stochastic mechanism are prone to spelling mistakes, grammatical errors, nonsensical words. We generate our samples by varying the temperature values to 0.2, 0.5, 0.7 and 1.0. For our evaluation and detection experiments, we randomly select 25 system generated samples, 2 samples generated at a temperature of 0.2, 10 samples at temperature 0.5, 5 samples at a temperature of 0.7 and 8 samples at temperature 1.0.

3.3.3. Customization of Malicious Intent

One important aspect of malicious emails is their harmful intent. The perfect attack vector will have malicious elements like a poisonous link or malware attachment wrapped in legitimate context, something which is sly enough to fool both a state-of-the-art email classifier as well as the victim. One novelty of this system training is the procedure of **injecting** malicious intent during training and **generating** malicious content in the synthetic emails.

We followed a *percentage based influx* of malicious content into the training model along with the legitimate emails. The training models were built by varying the percentage (5%, 10%, 30% and 50%) of phishing emails selected from the entire phishing dataset along with the entire legitimate emails dataset. We trained separate RNN models on all these configurations. For studying the varying content in emails, we generate samples using temperature values at 0.2, 0.5, 0.7 and 1.0.

3.4. Detection using Existing Algorithms

We perform a simple quantitative evaluation by using three text-based classification algorithms on our generated emails. Using the Python SciKit-Learn library, we test three popular text-based filtering algorithms - Support Vector Machines (Maldonado and L'Huillier, 2013), Naive Bayes (Witten et al., 2016) and Logistic Regression (Franklin, 2005).

The training set was modeled as a *document-term matrix* and the *word count vector* is used as the feature for building the models. For our evaluation, we train models using Support Vector Machines (SVM), Naive Bayes (NB) and Logistic Regression (LR) models on a training data of 300 legitimate emails from WikiLeaks archives[2] and 150 phishing emails from Cornell PhishBowl (IT@Cornell, 2018). We test the data on 100 legitimate emails from WikiLeaks archives that were not included in the training set and 25 'fake' emails that were generated by our natural language generation model.

4. Analysis and Results

We discuss the results of the generative RNN model in this section. We give examples of the email text generated with various training models and varying temperatures. We also provide the accuracy of the trained classifiers on a subset of these generated email bodies (after slight post processing). We try to provide a qualitative as well as a quantitative review of the generated emails.

4.1. Examples of Machine generated emails

(A) *Training only on Legitimates and varying sampling temperatures*

We show examples of emails generated using models trained on legitimate emails and sampled using a temperature of 1.0.

Example I at Temperature = 1.0:

Dear <NME> The article in the <NME> offers promotion should be somewhat changed for the next two weeks. <NME> See your presentation today. <NME>

[2] `https://wikileaks.org/`

Example II Example I at Temperature = 0.7:

Sir I will really see if they were more comments tomorrow and review and act upon this evening <NET>. The engineer I can add there some <LINK> there are the issues <NET>. Could you give me a basis for the call him he said

The example above shows that while small substrings make some sense. The sequence of text fragments generated make very little sense when read as a whole. When comparing these with the phishing email structure described in (Drake et al., 2004), the generated emails have very little malicious content. The red text marks the incongruous text pieces that do not make sense.

(B) ***Training on Legitimates + 5% Malicious content:***

In the first step of intent injection, we generate emails by providing the model with all the legitimate emails and 5% of the cleaned phishing emails data (Table 3). Thus for this model, we create the input data with 603 legitimate emails and 114 randomly selected phishing emails. We show as examples two samples generated using temperature values equal to 0.5 and 0.7.

Example I at Temperature = 0.5:

Sir Here are the above info on a waste of anyone, but an additional figure and it goes to <NET>. Do I <NET> got the opportunity for a possible position between our Saturday <NME> or <NET> going to look over you in a presentation you will even need <NET> to drop off the phone.

Example II at Temperature = 0.7:

Hi owners <NET> your Private <NET> email from <NET> at <NET> email <NET> Information I'll know our pending your fake check to eol thanks <NET> and would be In maintenance in a long online demand

The model thus consists of benign and malicious emails in an approximate ratio of 5:1. Some intent and urgency can be seen in the email context. But the incongruent words still remain.

(C) ***Training on Legitimates + 30% Malicious content:***

We further improve upon the model proposed in (B). In this training step, we feed our text generator all the legitimate emails (603 benign) coupled with 30% of the malicious emails data (683 malicious). This is an almost balanced dataset of benign and phishing emails. The following examples demonstrate the variation in text content in the generated emails.

Example I at Temperature = 0.5:

Sir we account access will do so may not the emails about the <NET> This <NET> is included at 3 days while when to <NET> because link below to update your account until the deadline we will received this information that we will know that your <NET> account information needs

Example II at Temperature = 1.0:

Dear registered secur= online, number: hearing from This trade guarded please account go to pay it. To modify your Account then fill in necessary from your notification preferences, please PayPal account provided with the integrity of information on the Alerts tab.

A good amount of text seems to align with the features of malicious emails described in (Drake et al., 2004) have malicious intent in it. We choose two examples to demonstrate the nature of text in the generated emails. We include examples from further evaluation of steps.

(D) ***Training on Legitimates + 50% Malicious content:***

In this training step, we consider a total of 50% of the malicious data (1140 phishing emails) and 603 legitimate emails. This is done to observe whether training on an unbalanced data, with twice the ratio of malign instances than legitimate ones, can successfully incorporate obvious malicious flags like poisonous links, attachments, etc. We show two examples of emails generated using deep learners at varying sampling temperatures.

Example I at Temperature = 0.7:

If you are still online. genuine information in the message, notice your account has been frozen to your account in order to restore your account as click on CONTINUE Payment Contact <LINK> UK.

Example IT at Temperature = 0.5:

Hi will have temporarily information your account will be restricted during that the Internet accounts and upgrading password An data Thank you for your our security of your Account Please click on it using the <NET> server This is an new offer miles with us as a qualified and move in

The generated text reflects malicious features like URL links and tone of urgency. We can assume that the model picks up important cues of malign behavior. The model then learns to incorporate such cues into the sampled data during training phase.

4.2. Evaluation using Detection Algorithm

We train text classification models using Support Vector Machines (SVM), Naive Bayes (NB) and Logistic Regression (LR) models on a training data of 300 legitimate emails from WikiLeaks archives[3] and 150 phishing emails from Cornell PhishBowl (IT@Cornell, 2018). We test the data on 100 legitimate emails from WikiLeaks archives that were not included in the training set and 25 'fake' emails that were generated by our natural language generation model trained on a mix of legitimate and 50% malicious emails. We randomly select the emails (the distribution is: 2 samples generated at a temperature of 0.2, 10 samples at temperature 0.5, 5 samples at a temperature of 0.7 and 8 samples at temperature 1.0) for our evaluation.

We use the Scikit-Learn Python library to generate the *document-term matrix* and the *word count vector* from a given sample of email text body used as a feature for train-

[3] https://wikileaks.org/

ing the classification models. The Table 4 reports the accuracy, precision, recall, and F1-scores on the test dataset using SVM, Naive Bayes and Logistic Regression classifiers.

Classifier	Accuracy	Precision	Recall	F1-score
SVM	71	72	85	78
NB	78	91	75	82
LR	91	93	95	94

Table 4: Classification metrics of generated emails

Despite the incoherent nature of the generated emails, the text-based classifiers do not achieve a 100% accuracy as well as F1-scores.

4.3. Comparison of emails with another NLG model

The authors in (Baki et al., 2017) discuss using a Recursive Transition Network for generating fake emails similar in nature to legitimate emails. The paper discusses a user study testing the efficacy of these fake emails and their effectiveness in being used for deceiving people. The authors use only legitimate emails to train their model and generate emails similar to their training data - termed as 'fake' emails. In this section, we compare a couple of examples selected randomly from the emails generated by the Dada Engine used in (Baki et al., 2017) and the outputs of our Deep Learning system generated emails.

Generated by the RNN (Example I):

Hi will have temporarily information your account will be restricted during that the Internet accounts and upgrading password An data Thank you for your our security of your Account Please click on it using the < *NET* > server This is an new offer miles with us as a qualified and move in

Generated by the RNN (Example II):

Sir Kindly limit, it [IMAGE] Please contact us contained on this suspension will not be = interrupted by 10 product, or this temporary cost some of the

Generated by the Dada Engine:

Great job on the op-ed! Are you going to submit? Also, Who will be attending?

The examples provide evidence that emails generated by the RNN are more on the lines of phishing emails than the emails generated by the Dada Engine. Of course, the goal of the email generated by the Dada engine is masquerade, not phishing. Because of the rule-based method employed that uses complete sentences, the emails generated by the Dada engine have fewer problems of coherence and grammaticality.

5. Error Analysis

We review *two types of errors* observed in the evaluation of our RNN text generation models developed in this study. *First*, the text generated by multiple RNN models suffer from repetitive tags and words. The example of the email body below demonstrates an incoherent and absurd piece of text generated by the RNN trained on legitimate emails and 50% of phishing emails with a temperature of 0.5.

Hi 48 PDX Cantrell <LINK> <NET> <NET> ECT ECT <NET> <NET> ECT ECT <NET> <NET> ECT ECT <NET> <NET> ECT ECT <NET> F <NET> ECT ECT <NET> G Slaughter 06 07 03 57 DEVELOPMENT 06 09 2000 07 01 <NET> <NET> ECT ENRON 09 06 03 10 23 PM To <NET> <NET> ECT ECT cc <NET> <NET> ECT ECT Subject Wow Do not underestimate the employment group contains Socal study about recession impact <NET> will note else to you for a revised Good credit period I just want to bring the afternoon <NET> I spoke to <NET> Let me know if

This kind of repetitive text generation was observed a number of times. However, we have not yet investigated the reasons for these repetitions. This could be an inherent problem of the LSTM model, or it could be because of the relatively small training dataset we have used. A third issue could be the temperature setting. More experiments are needed to determine the actual causes.

The *second aspect* of error analysis is to look at the misclassification by the statistical detection algorithms. Here we look at a small sample of emails that were marked as legitimate despite being fake in nature. We try to investigate the factors in the example sample that can explain the misclassification errors by the algorithms.

Example (A):

Hi GHT location <EID> Inc Dear <NET> Password Location <NET> of <NET> program We have been riding to meet In a of your personal program or other browser buyer buyer The email does not commit to a secure F or security before You may read a inconvenience during Thank you <NET>

Example (B):

Sir we account access will do so may not the emails about the <NET> This <NET> is included at 3 days while when to <NET> because the link below to update your account until the deadline we will received this information that we will know that your <NET> account information needs

Example (C):

Sir This is an verificati= <LINK> messaging center, have to inform you that we are conducting more software, Regarding Your Password : <LINK> & June 20, 2009 Webmail Please Click Here to Confirm

Examples (A), (B) and (C) are emails generated from a model trained on legitimate and 50% of phishing data (Type (D) in Section 4.1.) using a temperature of 0.7. There can be quite a few reasons for the misclassification - almost all the above emails despite being 'fake' in nature have considerable overlap with words common to the legitimate text.

Moreover, Example (A) has lesser magnitude of indication of malicious intent. And the amount of malicious intent in Example (B), although notable to the human eye, is enough to fool a simple text-based email classification algorithm. Example (C) has multiple link tags implying possible malicious intent or presence of poisonous links. However, the position of these links play an important role in deceiving the classifier. A majority of phishing emails have links at the end of the text body or after some action words like *click*, *look*, *here*, *confirm* etc. In this case, the links have been placed at arbitrary locations inside the text sequence - thereby making it harder to detect. These misclassification or errors on part of the classifier can be eliminated by human intervention or by designing a more sensitive and sophisticated detection algorithm.

6. Conclusions and Future Work

While the RNN model generated text which had 'some' malicious intent in them - the examples shown above are just a few steps from being coherent and congruous. We designed an RNN based text generation system for generating targeted attack emails which is a challenging task in itself and a novel approach to the best of our knowledge. The examples generated however suffer from random strings and grammatical errors. We identify a few areas of improvement for the proposed system - reduction of repetitive content as well as inclusion of more legitimate and phishing examples for analysis and model training. We would also like to experiment with addition of topics and tags like 'bank account', 'paypal', 'password renewal', etc. which may help generate more specific emails. It would be interesting to see how a generative RNN handles topic based email generation problem.

7. Bibliographical References

Baki, S., Verma, R., Mukherjee, A., and Gnawali, O. (2017). Scaling and effectiveness of email masquerade attacks: Exploiting natural language generation. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 469–482. ACM.

Basnet, R. B., Mukkamala, S., and Sung, A. H. (2008). Detection of phishing attacks: A machine learning approach. *Soft Computing Applications in Industry*, 226:373–383.

Chandrasekaran, M., Narayanan, K., and Upadhyaya, S. (2006). Phishing email detection based on structural properties. In *NYS Cyber Security Conference*, volume 3.

Chen, Y.-N. and Rudnicky, A. I. (2014a). Two-stage stochastic email synthesizer. In *INLG*, pages 99–102.

Chen, Y.-N. and Rudnicky, A. I. (2014b). Two-stage stochastic natural language generation for email synthesis by modeling sender style and topic structure. In *INLG*, pages 152–156.

Drake, C. E., Oliver, J. J., and Koontz, E. J. (2004). Anatomy of a phishing email. In *CEAS*.

Enron Corpus. (2015). Enron Email Dataset. `EnronEmailDataset`. Online; accessed 10 December 2017.

Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85.

Ghazvininejad, M., Shi, X., Choi, Y., and Knight, K. (2016). Generating topical poetry.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Henderson, M., Thomson, B., and Young, S. (2014). Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299.

IT@Cornell. (2018). Phish Bowl. `https://it.cornell.edu/phish-bowl`. Online; accessed 10 February 2018.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Maldonado, S. and L'Huillier, G. (2013). Svm-based feature selection and classification for email filtering. In *Pattern Recognition-Applications and Methods*, pages 135–148. Springer.

Reiter, E. and Dale, R. (2000). *Building natural language generation systems*. Cambridge university press.

Sidhaye, P. and Cheung, J. C. K. (2015). Indicative tweet generation: An extractive summarization problem?

Thakur, T. and Verma, R. (2014). Catching classical and hijack-based phishing attacks. In Atul Prakash et al., editors, *Information Systems Security*, pages 318–337, Cham. Springer International Publishing.

The New York Times. (2011). Sarah Palin emails: The Alaska archive. `http://documents.latimes.com/sarah-palin-emails/`. Online; accessed 10 December 2017.

Turner, R., Sripada, S., and Reiter, E. (2010). Generating approximate geographic descriptions. In *Empirical methods in natural language generation*, pages 121–140. Springer.

Verma, R. M. and Aassal, A. E. (2017). Comprehensive method for detecting phishing emailsusing correlation-based analysis and user participation. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, CODASPY 2017, Scottsdale, AZ, USA, March 22-24, 2017*, pages 155–157.

Verma, R. M. and Das, A. (2017). What's in a URL: fast feature extraction and malicious URL detection. In *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics, IWSPA@CODASPY 2017, Scottsdale, Arizona, USA, March 24, 2017*, pages 55–63.

Verma, R. M. and Dyer, K. (2015). On the character of phishing urls: Accurate and robust statistical learning classifiers. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, CODASPY 2015, San Antonio, TX, USA, March 2-4, 2015*, pages 111–122.

Verma, R. M. and Hossain, N. (2013). Semantic feature selection for text with application to phishing email detection. In *Information Security and Cryptology - ICISC*

2013 - 16th International Conference, Seoul, Korea, November 27-29, 2013, Revised Selected Papers, pages 455–468.

Verma, R. M. and Rai, N. (2015). Phish-idetector: Message-id based automatic phishing detection. In *SECRYPT 2015 - Proceedings of the 12th International Conference on Security and Cryptography, Colmar, Alsace, France, 20-22 July, 2015.*, pages 427–434.

Verma, R., Shashidhar, N., and Hossain, N. (2012). Detecting phishing emails the natural language way. *Computer Security–ESORICS 2012*, pages 824–841.

WikiLeaks. (2016). Hillary Clinton Email Archive. `https://wikileaks.org/clinton-emails/`. Online; accessed 10 December 2017.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Xie, S., Rastogi, R., and Chang, M. (2017). Deep poetry: Word-level and character-level language models for shakespearean sonnet generation.

Yao, Y., Viswanath, B., Cryan, J., Zheng, H., and Zhao, B. Y. (2017). Automated crowdturfing attacks and defenses in online review systems. *arXiv preprint arXiv:1708.08151*.

Yu, W. D., Nargundkar, S., and Tiruthani, N. (2009). Phishcatch - a phishing detection tool. *2009 33rd Annual IEEE International Computer Software and Applications Conference*, 2:451–456.