Classifying Security Advice using NLP

Sahil Farishta
sahilf@umich.edu

Christine Konicki
ckonicki@umich.edu

Alexandra Veliche
aveliche@umich.edu

Abstract

In an ever-changing environment, it is becoming increasingly more important for users to properly implement good security practices. As a result, it is crucial for users to have security advice that they can use and is both correct and actionable. Previous research has shown that there is a lot of security advice on the internet that is correct and important to implement. There is, however, a difference between what security professionals perceive to be difficult and what users perceive to be difficult. As users turn to more informal sources to get their security advice, including online platforms like Reddit and StackExchange, it is important to examine the way advice on these platforms will be perceived by users. In this paper, we examine the quality of advice on these websites by training a classifier that uses natural language processing with labelled training examples from previous studies. Our model reveals common characteristics of posts that are found to be most actionable by users. Security professionals can use these insights to shape their answers and advice in a way that will be more conducive to users.

1 Introduction

In this section, we describe the motivation behind our project and review some related work on security advice recommendations and investigations involving the quality of online requests.

1.1 Motivation

End users are not as inclined to pore through professional documentation, but rather pursue more accessible sources, such as online articles, peer anecdotes, professional experience (if they work in the industry), online forums, and even past negative experiences involving costly security incidents. As the next section will show, there is a disconnect between the advice supplied by professionals and its use by everyday users. We want to investigate the quality of advice found on less formal online platforms, specifically Reddit and StackExchange. In particular, we want to investigate if the more personable and anonymous nature of these platforms is conducive to advice that is better structured to bridge the gap between experts and casual users. Ideally, our analysis of the quality of advice on these online platforms will give security professionals an insight into improving their security advice for end users, through positive and negative examples alike.

1.2 Background and Related Work

1.2.1 Redmiles' Study of Security Advice

Redmiles et al. [13] investigated the quality, feasibility, and comprehensibility of the advice given to end users to defend their systems from attackers. The authors gave the first exhaustive taxonomy of advice directed at end users by analyzing over 1200 documents from the web and identifying 374 unique pieces of advice. They also gave advice quality metrics: perceived actionability, perceived efficacy, and comprehensibility. Then they conducted a study consisting of approximately 1500 users and 41 professional security experts to evaluate the quality of all 374 advice imperatives using the given metrics as a rubric. The goal was to identify areas that needed improvement so that more users would incorporate the advice into their behavior on the web.

The results suggested that users understand the advice given to them and can take the necessary action, but prioritization is an issue. Giving users over 140 directives and describing them all as among the top 5 most important actions to take makes it extremely unlikely that they would actually adopt the advice. [13] ultimately concludes that the advice recommended needs to be of the highest impact, better organized, and much more minimal for users to actually follow it.

1.2.2 Other Works on Security Advice

It is important to note that [13] consists of one of many possible approaches to the evaluation of security advice, applied to one of many possible advice sources. Internet sources are unsurprisingly among the most common sources for advice on security [12], but it is still worth studying the quality of advice from other sources. [12] collected a dataset consisting of personal stories, news articles, and web pages giving informal security advice, and then used a probabilistic "bag of words" topic model to identify the security topics being discussed. They found that there is a disconnect between what educational sources encourage end users to direct their attention to and what end users tend to seek advice on when considering security (e.g. passwords, encryptions). They also found that the language used by professionals in security discourse was very different from the language used by everyday users. This suggests that (1) informal sources may be more useful to users if utilized properly, and (2) language manifests another noteworthy disconnect between professional advice and its relevance to users.

This disconnect is confirmed by [6], which conducted a study comparing self-reported security practices of both everyday users and professionals who had at least 5 years' experience in the field. Unsurprisingly, there was a large discrepancy between these two groups, and a similar study conducted by [2] replicated these results. The original starting point for the taxonomy of advice given in [13] can be found in [14], another work that sought to investigate why users often do not follow expert advice. Similar to other works, [14] found that the lack of a consensus amongst the community of security professionals was a likely cause for this discrepancy.

1.2.3 Success in Online Interaction

Often, the goal of studying interaction in online communities and social networks is to understand what drives users to consume the content that they do and respond to certain types of content. [10] argues that, at least in the case of scientific journalism articles, the quality of the writing is of considerable importance for user engagement. Spelling, grammar, and the presence of a consistent narrative are all relevant factors [10]. The persuasiveness of an argument can determine the success or failure of users engaged in an internet debate [16], which suggests that online users do have some standards for the content's adequacy and influence, even on an informal and anonymous platform such as Reddit. When submitting requests, who you are and how you ask matters [1]. On social media, the evaluation of users by other users can influence how a post or comment is perceived [4, 9]. In the case of Reddit, this may pertain to how active someone is in a particular community, how many upvotes a post gets, and quantities like karma, reputation with the moderators, and awards that can indicate the relative status of that user in the community.

Researchers have mined Reddit content and analyzed it to evaluate its success within a community or to evaluate the general themes of that content for other purposes. [1] used a webcrawler to extract the entire post and comment history of r/Random_Acts_Of_Pizza: a subreddit where users ask for free pizza from the community, a member (ideally) obliges, and the contribution is recorded. They analyzed what makes an online request for pizza successful or not by fitting the mined data to a simple regression model defined by different factors: text length, narrative, user reputation, community-age, and time. Requests that were successful tended to be recent, create a sense of trust in the audience, and construct a good narrative (i.e. "I'm having a bad night and would like a pizza to make it better" would be received more favorably than "I want a pizza to throw in someone's face") [1]. [11] has applied natural language processing (NLP) to Reddit comments left in dermatology subreddits from 2005 to 2017 in order to identify trends in patient engagement regarding afflictions such as eczema and acne. Topics and themes were identified using latent Dirichlet allocation, with the goal of using the results to improve dermatologic research and engagement with the general public. Although the Dirichlet model was unsupervised and therefore had no ground truth labels, it was assumed that there was no incentive for users to lie on an anonymous forum. This is one reason for our choice to investigate forums like Reddit and StackExchange.

1.2.4 Natural Language Processing

We looked at various NLP models to use in our study. Previous work has shown that "basic" NLP models, such as Continuous Bag of Words (CBOW), Skip-Gram, and GloVe, perform better than more advanced models [3]. This inspired our choice of doc2Vec which is a CBOW model that outputs a single vector per document, allowing for a condensed size-independent output for each document [7]. Additionally, doc2vec does not need labels for the dataset, which increases the potential training data and applications for this model. For example, doc2vec could work exclusively on webscraped data that is unlabelled to provide a model that is trained on documents most similar to the documents it will be evaluated on. The work done in [3] also found that the training time needed for a CBOW model was the shortest across all the methods. This combination of fast training time, lack of necessary labels, and high performance lead us to use a doc2vec model.

2 Methodology

In this section, we give our experimental setup. We begin by describing the webcrawling processes used to acquire the necessary data from Reddit and StackExchange. Then we introduce the NLP model for analyzing the datasets to find advice and the machine learning model used to classify the advice quality according to the taxonomy given by [13].

2.1 Webscraping

2.1.1 Reddit Webscraping

We decided to collect data from the subreddit r/AskNetsec, a forum for asking questions about information security from the perspective of a company or professional enterprise. Unlike r/Cybersecurity101, which is intended for beginner topics from a home, family, or personal perspective, r/AskNetsec is intended primarily for professional security advice and was therefore most relevant to our project. In addition, the subreddit rules require commenters to know what they are talking about when answering a question; comments that demonstrate a naive understanding or give wrong information are removed, and repeat violators are subject to removal from the community. To gather data from the subreddit, we used the Python Reddit API Wrapper (PRAW) to easily access the Reddit API. The API requires a unique user agent so that Reddit can identify the source of any and all network requests and make sure they follow the API's rules. Since PRAW has a built-in user agent, all we had to do was write a script that could intelligently scrape the subreddit for relevant posts and comments, and sort the comments from high to low upvotes. The final validation set consisted of approximately 430 posts.

To filter for posts that contained requests for professional security advice, the script searched for posts containing keywords such as "secure," "best practices," and "tips." However, r/AskNetsec also contained posts with themes that had nothing to do with security advice, but appeared frequently enough in the search results that they had to be filtered out. Keywords such as "career" and "military" were used to filter out posts asking about career advice, getting into the security field with a military background, or the average day in the life of a professional in the field. Another frequent topic was preparation for SANS GIAC certification exams, so these were filtered out as well. We also filtered out posts that asked for security advice but either had no comments or had comments that were sarcastic or asked the original poster for clarification without response but gave no advice. Once the filtration process was complete, we examined the remaining posts and removed any random outliers posts that were not filtered out by keyword. For example, one of the deleted posts involved a student who had breached his high school's firewall and wanted to know whether he would be suspended or arrested. We made sure to exclude user data (such as usernames and user karma) from our validation set for ethical reasons.

2.1.2 StackExchange Webscraping

We chose to focus on the StackExchange Information Security site because it was most relevant to cybersecurity advice, as reflected in its official description: "A question and answer site for information security professionals." To gather data from the Information Security StackExchange site, we wrote a Python webscraping script based on [15], which was originally intended for scraping StackExchange Hot Network Questions. The script relies on the Python BeautifulSoup library for HTML extraction and parsing and the Pandas library for data analysis. We first scraped the first 20 pages of the most recent posts on Information Security, then filtered out the "[closed]" and "[duplicate]" posts to ensure that each post was valid and unique. We then manually sifted through the set of returned posts and corrected some misinterpreted symbols. The information gathered from every post was limited to the question title, question description, first or accepted answer, and the post URL. At the end of the filtering process, we obtained a StackExchange validation set of approximately 530 posts.

For ethical reasons, we did not collect any user data. We also used periodic randomized pausing in the script to mimic the rate at which an average human user would access the information, and thereby avoid overwhelming the StackExchange server with too many requests. During our debugging stage, we did overwhelm the server a few times, and the scraper's IP address was temporarily blocked, but this impacted only one home network and was remedied soon after.

2.2 Natural Language Processing

We created an NLP model that would convert the text from the answers into a data vector. This transformed data vector was used as input to the machine learning model that would classify the answers. We used doc2vec [7] as our NLP model which outputs a single vector for each document. Our doc2vec model was trained on the documents collected in the Redmiles paper and was set to output a 300-dimensional vector. Once trained, the doc2vec model was then run on the documents collected in [13], and the output vectors were stored for use in training the machine learning model. We also saved the trained doc2vec model to transform the webscraped data later.

2.3 Machine Learning Model

We created a machine learning model that was designed to classify the various answers using four boolean metrics: whether the user would be confident implementing the advice, and whether the given advice would be time consuming, disruptive, and difficult to implement. The machine learning model used 306 pieces of information as input. The first 300 came in the form of the 300-dimensional representation of the answer text that is outputted by the NLP model. The next 4 inputs were the perceived sentiments of the answer. We used the VADER sentiment analysis tool to compute the negative, neutral, positive, and compound sentiment scores for each answer [5]. The next input was the number of grammar and spelling errors in the answer. Finally, we included the length of the post in words. Figure 1 shows an example of the training data along with the associated labels with the 300-dimensional vector output from the NLP model omitted due to size constraints.

During the training phase, we focused exclusively on the data provided in the Redmiles paper since the machine learning model relied on a pre-labelled dataset. We considered Support Vector Machines (SVCs) and Random Forest Classifiers (RFCs) for our models. For each model option, we performed a hyperparameter search to find the optimal parameters using cross-fold validation. We considered our model options for each label independently, aiming for four models by the end of this phase with one model per label. We compared the models using a weighted accuracy score which weighted each possible output equally to handle the unbiased data set. The Redmiles dataset primarily had advice that was classified as “non-confident”, “not time consuming”, “not disruptive”, and “not difficult”. This led to difficulties in training a robust model. We attempted to use oversampling techniques such as Sympathetic Minority Over-sampling Technique (SMOTE) [8] to combat this imbalance, but this led to trained models with lower balanced-accuracy so we continued without performing data balancing. After performing the hyperparameter search, we saved the models with the highest balanced accuracy for each label for further use.

3 Results

Using the saved models, we evaluated the performance both on the labelled dataset and on the dataset we curated from Reddit and StackExchange. We found during training that the best model for predicting each label was an SVC. The RFC was not able to produce a robust model due to higher sensitivity with regards to the imbalanced dataset.

3.1 Performance on Labelled Dataset

The results of the best models on the labelled dataset are seen in Table 1 along with the features each model prioritized.

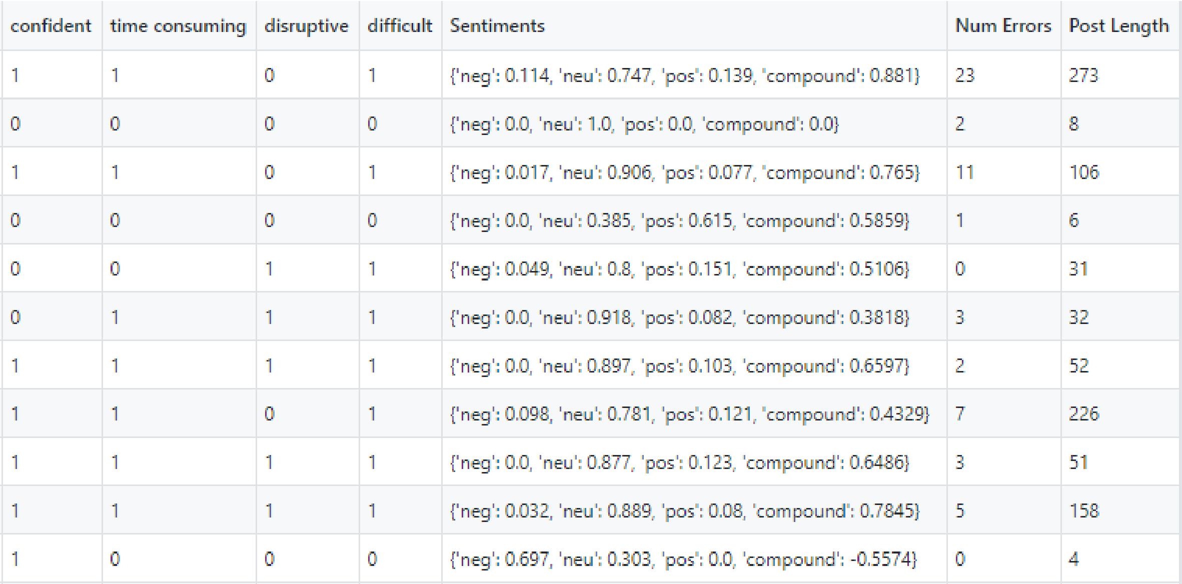
We see that the model for the “disruptive” label was not as robust as the rest of the models, with an accuracy of 50% compared to the other models which had accuracies around 70%. Additionally, we see that each model prioritized different features. The model for the ”confident” label prioritized the negative sentiment score and the post length in determining the label, while the model for the “disruptive” label primarily looked at the NLP word vector features. We emphasize here that all of th models valued the NLP word vector features; however, comparing various indices in the vector that each model prioritized is not meaningful as there is no information we can gain from this. We found that all of the models did not value the number of errors very much. After investigating this further, we realized that technical terms and abbreviations were being counted as errors by our library since they did not exist in the library’s dictionary, which explains why none of the models found this field to be particularly insightful.

3.2 Evaluation on Webscraped Dataset

After finding the best performing models, we ran the models on the data scraped from Reddit and StackExchange. We ran the scraped answers through the NLP model and computed the sentiment scores, number of errors, and post length, before passing those 306 features as input to the machine learning model. We show the primary results in Figure 2.

After obtaining the results, we compared the scores given by the model to each post with our assessment of the post. In this way, we checked that the model was correctly evaluating each post based on the specified criteria. Overall, the model seemed to be quite accurate in its evaluation.

We see that the answers from StackExchange were far more likely to be labelled as “confident”, “not difficult”, and “time consuming” than those on Reddit. However, we see that the model for the “disruptive” label performed very poorly on the StackExchange data, predicting only about 1% of the StackExchange answers to be disruptive to implement. In terms of the metadata, we see that the posts on Reddit were on average shorter, with fewer errors, than the ones on StackExchange. The answers on Reddit had 77 words and 3 errors on average, while the answers on StackExchange has 205 words and 35 errors on average. The added errors in the StackExchange dataset may come from a higher usage of technical terms and abbreviations that were considered errors.



confident	time consuming	disruptive	difficult	Sentiments	Num Errors	Post Length
1	1	0	1	{'neg': 0.114, 'neu': 0.747, 'pos': 0.139, 'compound': 0.881}	23	273
0	0	0	0	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	2	8
1	1	0	1	{'neg': 0.017, 'neu': 0.906, 'pos': 0.077, 'compound': 0.765}	11	106
0	0	0	0	{'neg': 0.0, 'neu': 0.385, 'pos': 0.615, 'compound': 0.5859}	1	6
0	0	1	1	{'neg': 0.049, 'neu': 0.8, 'pos': 0.151, 'compound': 0.5106}	0	31
0	1	1	1	{'neg': 0.0, 'neu': 0.918, 'pos': 0.082, 'compound': 0.3818}	3	32
1	1	1	1	{'neg': 0.0, 'neu': 0.897, 'pos': 0.103, 'compound': 0.6597}	2	52
1	1	0	1	{'neg': 0.098, 'neu': 0.781, 'pos': 0.121, 'compound': 0.4329}	7	226
1	1	1	1	{'neg': 0.0, 'neu': 0.877, 'pos': 0.123, 'compound': 0.6486}	3	51
1	1	1	1	{'neg': 0.032, 'neu': 0.889, 'pos': 0.08, 'compound': 0.7845}	5	158
1	0	0	0	{'neg': 0.697, 'neu': 0.303, 'pos': 0.0, 'compound': -0.5574}	0	4

Figure 1: Example of dataset

Model	Weighted Accuracy(%)	Features Valued
Confidence	69	Negative sentiment score, Post length
Time-Consumption	70	Positive sentiment score
Disruptiveness	50	Word vector features
Difficulty	70	Post length

Table 1: Labelled dataset performance

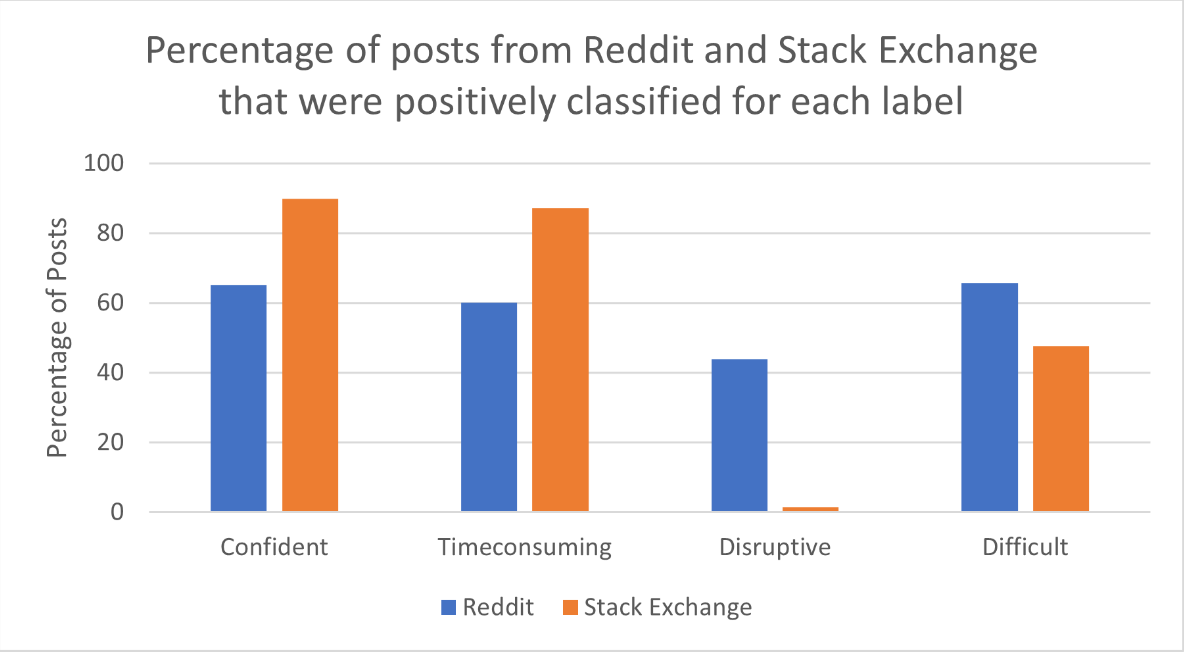
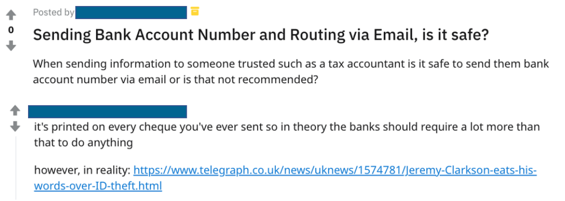


Figure 2: Classification results on Reddit and StackExchange datasets



0

Posted by [redacted]

Sending Bank Account Number and Routing via Email, is it safe?

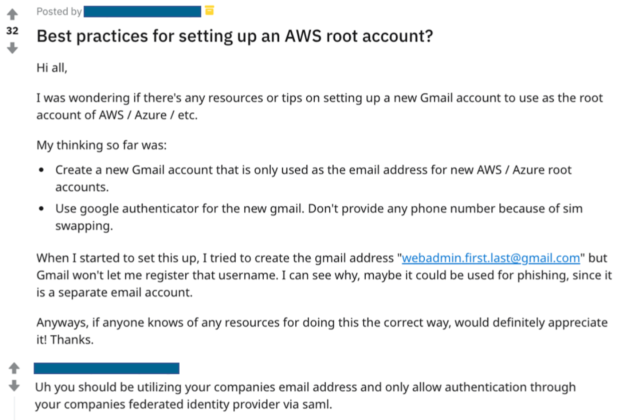
When sending information to someone trusted such as a tax accountant is it safe to send them bank account number via email or is that not recommended?

[redacted]

it's printed on every cheque you've ever sent so in theory the banks should require a lot more than that to do anything

however, in reality: https://www.telegraph.co.uk/news/uknews/1574781/Jeremy-Clarkson-eats-his-words-over-ID-theft.html

Figure 3: Reddit post that received a classification score of (1,0,0,0). The OP was asking whether or not it was safe to send their bank account and routing numbers over email. The comments were cogent and definitive.



32

Posted by [redacted]

Best practices for setting up an AWS root account?

Hi all,

I was wondering if there's any resources or tips on setting up a new Gmail account to use as the root account of AWS / Azure / etc.

My thinking so far was:

- Create a new Gmail account that is only used as the email address for new AWS / Azure root accounts.
- Use google authenticator for the new gmail. Don't provide any phone number because of sim swapping.

When I started to set this up, I tried to create the gmail address "webadmin.first.last@gmail.com" but Gmail won't let me register that username. I can see why, maybe it could be used for phishing, since it is a separate email account.

Anyways, if anyone knows of any resources for doing this the correct way, would definitely appreciate it! Thanks.

[redacted]

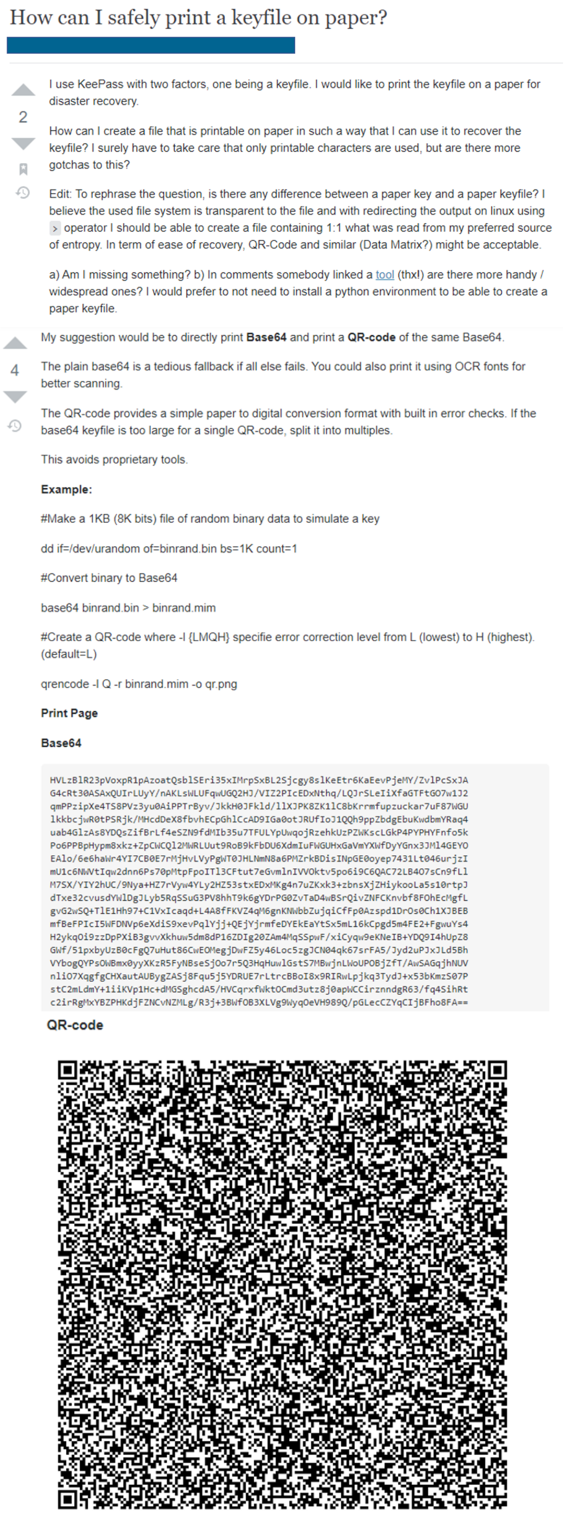
Uh you should be utilizing your companies email address and only allow authentication through your companies federated identity provider via saml.

Figure 4: Reddit post that received a classification score of (0,0,0,0). The OP was asking what the best security practices were for setting up an AWS root account. The language in the comments suggested some unsurety, but the advice given was still brief and clear.

3.2.1 Reddit Dataset

The results for the Reddit posts generally matched our own assessment of the posts based on the criteria. For example, Figure 3 depicts a post that received a score of (1,0,0,0), which is translated to "confident", "not time-consuming", "not disruptive", and "not difficult". This is reasonable because the answer is concise, straightforward and contains a reference which would help a user feel confident about implementing it, while not appearing to be particularly time-consuming, disruptive, or difficult.

The post in Figure 4, on the other hand, received a score of (0,0,0,0), which is translated to "not confident", "not time-consuming", "not disruptive", and "not difficult". The lack of direction reflects the non-confidence score as a user would not be much better equipped to resolve the problem after reading this answer. However, the brevity and lack of detail reflect that the advice does not appear to be time-consuming, disruptive, or difficult to implement.



How can I safely print a keyfile on paper?

[redacted]

2

I use KeePass with two factors, one being a keyfile. I would like to print the keyfile on a paper for disaster recovery.

How can I create a file that is printable on paper in such a way that I can use it to recover the keyfile? I surely have to take care that only printable characters are used, but are there more gotchas to this?

Edit: To rephrase the question, is there any difference between a paper key and a paper keyfile? I believe the used file system is transparent to the file and with redirecting the output on linux using `>` operator I should be able to create a file containing 1:1 what was read from my preferred source of entropy. In term of ease of recovery, QR-Code and similar (Data Matrix?) might be acceptable.

a) Am I missing something? b) In comments somebody linked a tool (thx!) are there more handy / widespread ones? I would prefer to not need to install a python environment to be able to create a paper keyfile.

4

My suggestion would be to directly print **Base64** and print a **QR-code** of the same Base64.

The plain base64 is a tedious fallback if all else fails. You could also print it using OCR fonts for better scanning.

The QR-code provides a simple paper to digital conversion format with built in error checks. If the base64 keyfile is too large for a single QR-code, split it into multiples.

This avoids proprietary tools.

Example:

#Make a 1KB (8K bits) file of random binary data to simulate a key

dd if=/dev/urandom of=binrand.bin bs=1K count=1

#Convert binary to Base64

base64 binrand.bin > binrand.mim

#Create a QR-code where -l {LMQH} specifie error correction level from L (lowest) to H (highest). (default=L)

qrencode -l Q -r binrand.mim -o qr.png

Print Page

Base64

```
HVLzBlR23pVoxpR1pAzoatQsblSEri35xIMrpSxBL2Sjcgy8slKeEtr6KaEevPjeMY/ZvlPcSxJA
G4cRt30ASAxQUIrLUyY/nAKLsWLUFqwUGQ2HJ/VIZ2PIcEDxNthq/LQJrSLeIiXfaGTFtGO7w1J2
qmPPzipXe4TS8PVz3yu0AiPPTrByv/JkkH0JFkld/llXJPK8ZK1lC8bKrrmfupzuckar7uF87WGU
lkkbcjwR0tPSRjk/MHcdDeX8fbvhECpGhlCcAD9IGa0otJRUfIoJ1QQh9ppZbdgEbuKwdbmYRaq4
uab4GlzAs8YDQsZifBrLf4eSZN9fdMIb35u7TFULYpUwqojRzehkUzPZWKscLGkP4PYPHYFnfo5k
Po6PPBpHypm8xkz+ZpCWCQl2MWRLUut9RoB9kFbDU6XdmIuFWGUHxGaVmYXWfDyYGnx3JMl4GEYO
EAlo/6e6haWr4YI7CB0E7rMjHvLVyPgWT0JHLNmN8a6PMZrkBDisINpGE0oyep7431Lt046urjzI
mU1c6NWVtIqw2dnn6Ps70pMtpFpoITl3CFtut7eGvmlnIVVOktv5po6i9C6QAC72LB4O7sCn9fLl
M7SX/YIY2hUC/9Nya+HZ7rVyw4YLy2HZ53stxEDxMKg4n7uZKxk3+zbnsXjZHiykooLa5s10rtpJ
dTxe32cvusdYWlDgJLyb5RqSSuG3PV8hhT9k6gYDrPG0ZvTaD4wBSrQivZNFCKnvbf8FOhEcMgfL
gvG2wSQ+TlE1Hh97+C1VxIcaqd+L4A8fFKVZ4qM6gnKNWbbZujqiCfFp0Azspd1DrOs0Ch1XJBEB
mfBeFPIcI5WFDNVp6eXdiS9xevPqlYjj+QEjYjrmfeDYEkEaYtSx5mL16kCpgd5m4FE2+FgwuYs4
H2ykqOi9zzDpPXiB3gvvXkhuw5dm8dP16ZDIg20ZAm4MqSSpwF/xiCyqw9eKNeIB+YDQ9I4hUpZ8
GWf/51pxbyUzB0cFgQ7uHut86CwEOMegjDwFZ5y46Loc5zgJCN04qk67srFA5/Jyd2uPJxJLd5Bh
VYbogQYPsOWBmx0yyXKzR5FyNBseSjOo7r5Q3HqHuwlGstS7MBwjnLWoUPOBjZfT/AwSAGqjhNUV
nliO7XqgfgCHXautAUBygZASj8Fqu5j5YDRUE7rLtrcBBoI8x9RIRwLpjkq3TydJ+x53bKmzS07P
stC2mLdmY+1iiKVp1Hc+dMGSghcdA5/HVCqrxfWktOCmd3utz8j0apWCCirznndgR63/fq4SihRt
c2irRgMxYBZPHKdjFZNCvNZMLg/R3j+3BWfOB3XLVg9WyqOeVH989Q/pGLecCZYqCIjBFho8FA==
```

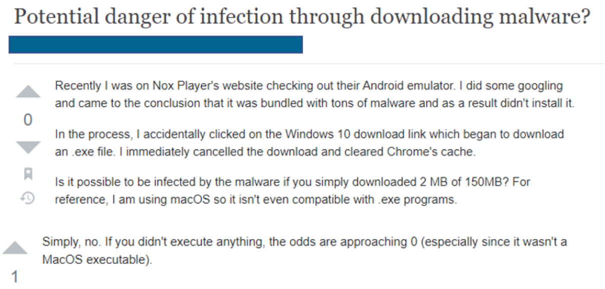
QR-code

Figure 5: StackExchange post that received a classification score of (1,1,0,0). The OP was asking for a safe way to print a keyfile on paper. The accepted response was explicit and straightforward.

3.2.2 StackExchange Dataset

As with the Reddit data, the results for the StackExchange posts generally matched our assessment of the posts based on the criteria. For example, Figure 5 displays post with a score of (1,1,0,0), which translates to "confident", "time-consuming", "not disruptive", and "not difficult". The response is very detailed and contains several explicit examples for what the original poster can do, which reflects the confidence score. While the length of the response is unusually long, the "disruptive" and "difficult" labels were not triggered. This is likely because the response largely consists of figures and examples that are helpful in understanding the advice and how to implement it. This also shows that even though the "difficult" model looks primarily at post length, that is not the only feature it considers as it can label long posts as not being difficult to implement.

In contrast, the post depicted in Figure 6 received a score of (0,0,0,0), which is translated to "not confident", "not time-consuming", "not disruptive", and "not difficult". This reflects the single-sentence brevity of the response and lack of justification. While nothing suggested stands out as time-consuming, disruptive, or difficult, there are no details here that would help a user investigate the issue any further, other than simply trusting the answer.



Potential danger of infection through downloading malware?

[redacted]

Recently I was on Nox Player's website checking out their Android emulator. I did some googling and came to the conclusion that it was bundled with tons of malware and as a result didn't install it.

In the process, I accidentally clicked on the Windows 10 download link which began to download an .exe file. I immediately cancelled the download and cleared Chrome's cache.

Is it possible to be infected by the malware if you simply downloaded 2 MB of 150MB? For reference, I am using macOS so it isn't even compatible with .exe programs.

Simply, no. If you didn't execute anything, the odds are approaching 0 (especially since it wasn't a MacOS executable).

Figure 6: StackExchange post that received a classification score of (0,0,0,0). The OP was asking if there is danger of infection just by downloading malware. The first response is brief with no justification.

4 Conclusion

We developed a machine learning model that can accurately classify advice from both Reddit and StackExchange. The model leverages a doc2vec model to transform the text responses into meaningful word vectors using NLP and specific metrics such as sentiment scores, post length, and number of errors. It classifies each piece of advice based on four criteria: whether the user would be confident implementing this advice and whether the solution is time-consuming, disruptive, or difficult.

Our model found that advice from StackExchange was generally more helpful, but had longer descriptions and was more time-consuming. While post content was an important factor in evaluating the advice, the conveyed sentiment and post length were also influential. This leads us to make the following recommendation: When offering security advice, information security professionals should consider the sentiment they convey and the length of their response, in addition to the essence of their advice.

4.1 Limitations

The model we developed has several limitations. First of all, it was trained using a limited and unbalanced dataset from [13], which made the model less robust than we would have liked. In particular, the model that measured disruptiveness performed quite poorly on the StackExchange data. This class imbalance is due to the lack of posts with labels "not confident", "time-consuming", "disruptive", or "difficult".

Another issue is that many technical terms and abbreviations were not part of the corpus that the grammar and spell check library leveraged, so the model marked these as grammatical and spelling errors. We believe this to be the main reason why the number of errors had little influence in the evaluation of the dataset, as answers that are poorly written and hard to follow due to grammar and spelling errors could have a similar number of errors flagged as a well written post with lots of technical terms and abbreviations.

For ethical reasons, we did not consider any information about the original poster. While user data is often anonymized on both Reddit and StackExchange, this information could be useful in determining how technical or long a piece of advice can be while still being labelled as "confident", "not time-consuming", "not disruptive", and "not difficult". This approach would primarily use information about the poster to figure out what level of advice they would be comfortable implementing. Some related issues are that users are not the same across both platforms, and the sub-forums themselves are intended for slightly different purposes, namely that the StackExchange Information Security site is restricted to professionals with sufficient reputation, while the AskNetSec subreddit is more permissive and relies on the moderators to purge users who leave bad, unprofessional comments. These natural differences make the evaluated data difficult to compare accurately. Finally, our StackExchange validation dataset was slightly larger than the Reddit dataset, which may have contributed to variations in the percentages of posts assigned to each label.

4.2 Future Work

Directions for future work include the following: To improve accuracy, it would be useful to focus on single models that output all four of the labels. We could also study the effects of the NLP model itself on trained classifiers and use pre-trained models, such as GloVe, to create word embeddings. It would also be worth training the NLP model on data from Reddit and StackExchange, as our model was trained exclusively on the dataset from [13]. Since unsupervised learning does not require data with known labels, this would be relatively straightforward to do. However, this would not work when experimenting with supervised algorithms such as the ones discussed in [3]. It would also be interesting to experiment with different hyperparameters for the NLP model, such as vector dimensionality. This work could be extended to different media on the internet other than just Reddit and StackExchange. Finally, a foreseeable application of this work is integrating the model into an online tool that would enable security professionals to check how their advice would be perceived by a typical user and modify their response accordingly.

References

[1] Althoff, T., Salehi, N., and Nguyen, T. Random acts of pizza : Success factors of online requests.

[2] Busse, K., Schafer, J., and Smith, M. Replication: No one can hack my mind revisiting a study on expert and non-expert security practices and advice. In *Symposium on Usable Privacy and Security (SOUPS)* (2019).

[3] Ghannay, S., Favre, B., Esteve, Y., and Camelin, N. Word embeddings evaluation and combination. In *10th edition of the Language Resources and Evaluation Conference (LREC)*.

[4] Guha, R. V., Kumar, R., Raghavan, P., and Tomkins, A. S. Propagation of trust and distrust. In *13th international conference on World Wide Web* (2004), ACM, pp. 403–412.

[5] Hutto, C.J. & Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*.

[6] Ion, I., Reeder, R., and Consolvo, S. ...no one can hack my mind": Comparing expert and non-expert security practices. In *Symposium on Usable Privacy and Security (SOUPS)*.

[7] Le, Q., and Mikolov, T. Distributed representations of sentences and documents. In *31st International Conference on International Conference on Machine Learning (ICML)* (2014), pp. 1188–1196.

[8] Lemaître, G., Nogueira, F., and Aridas, C. K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research 18*, 17 (2017), 1–5.

[9] Leskovec, J., Huttenlocher, D., and Kleinberg, J. Signed networks in social media. In *SIGCHI Conference on Human Factors in Computing Systems* (2010), ACM, pp. 1361–1370.

[10] Louis, A., and Nenkova, A. What makes writing great? first experiments on article quality prediction in the science journalism domain. In *Transactions of the Association for Computational Linguistics* (2013), pp. 341–352.

[11] Okon, E., Rachakonda, V., Hong, H. J., Callison-Burch, C., and Lipoff, J. B. Natural language processing of reddit data to evaluate dermatology patient experiences and therapeutics. In *Joural of the American Academy of Dermatology* (2019).

[12] Rader, E., and Wash, R. Identifying patterns in informal sources of security information. In *Journal of Cybersecurity Advance Access*, Oxford University Press, pp. 1–24.

[13] Redmiles, E. M., Warford, N., Jayanti, A., Koneru, A., Kross, S., Morales, M., Stevens, R., and Mazurek, M. L. A comprehensive quality evaluation of security and privacy advice on the web. In *Usenix Security*.

[14] Reeder, R. W., Ion, I., and Consolvo, S. 152 simple steps to stay safe online: security advice for non-tech-savvy users. In *IEEE Security and Privacy* (2017).

[15] Shaver, B. Stackexchange classification, Oct. 2017.

[16] Tan, C., Niculae, V., Danescu-Niculescu-Mizil, C., and Lee, L. Winning arguments: Interaction dynamics and persuasion strategies in good- faith online discussions. In *International Conference on World Wide Web (WWW)* (2016).