Comparing Word Embeddings and Neighborhoods of Spam and Non-Spam Emails

Katherine Ann Sick
Dep. of Computer Science and Linguistics
University of Illinois at Urbana-Champaign
kasick2@illinois.edu

Abstract

This paper follows the development of a semi-supervised experiment that uses Word2Vec models on our different classes in our spam and non-spam corpus. This allows us to compare word embeddings across different registers, and analyze the neighborhoods of certain words in each. In our findings, we see the formation of different neighborhoods based on which category of messages we are looking at, and we also see multiple patterns emerge in this data. In further observing this, we can learn valuable insights on how to classify certain messages into spam or not, which can be helpful in creating more accurate algorithms to identify them.

1 Introduction

Many social media sites or apps (including emails) utilize spam recognition tools in order to sort through messages that may contain unsolicited or unwanted information or text. However, these tools are not always perfect and may sometimes incorrectly identify which messages belong in which category. In these cases, we may find that some spam messages are identified as non-spam, and will be shown to the user even if it could potentially be harmful. On the other hand, if non-spam messages are identified as spam, its possible a user could miss important messages.

Although manual identification would be much more accurate, the volume of messages to sort through makes this an impossible task. There is also the case that some may not be able to identify malicious or ill-intended text on there own and could fall victim to its content. This is why this type of tool is necessary to the functionality of these websites and programs despite its possible short-comings.

In studying the differences in the word embeddings between select examples of spam vs non-spam, we can find useful trends that will benefit us in creating better identifying algorithms. The question that we observe in this paper specifically is what kind of words/word embeddings are common and utilized in the different classes of messages. Additionally, once we find these words, are there any underlying patterns that can help us understand how to classify a message as one or the other?

For the purposes of this experiment, we specifically are using labeled/unlabeled emails from the Enron spam dataset, and the TREC 2007 public corpus combined. However, even though we are only looking at email formatted messages, this study still applies to possible spam messages received on other platforms.

If we successfully observe these trends, we can use this information to efficiently and effectively identify spam before it ever gets to a user. Furthermore, in training a model on this data, we could easily automate the process where manual human annotation is not needed.

2 The Dataset and Corpus

In this paper, we analyze a combination of two corpora containing email data that is manually labeled as spam, or not spam.

The first of these corpora is the Enron-Spam dataset which contains thousands of entries with the raw original message text, and a label based on whether it is classified as spam or not. This data has already been stripped of any messages that contain viruses or messages sent from the owner of the inbox to themselves as specified in its documentation.

The second corpus here is the TREC 2007 Public Corpus. Similar to our first corpus, this dataset also contains thousands of entries with the raw original message text, and a label based on whether it is classified as spam or not.

In total, these corpora combined make up 83446 records of email data for us to use in this experiment. Conveniently, this data has already been combined in a kaggle dataset, and made to assimi-

late the labels to be in the same format. We can also see the distribution of spam (label of "1") vs. non-spam (label of "0") entries as observed in figure one below.

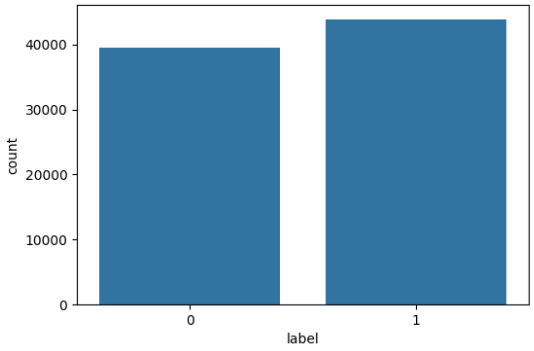

Figure 1: Label Distribution between Spam and Non-Spam Entries

This data is already fully labeled, but for the purposes of our experiment, we end up only using the labels as a way to split up our data for training, whereas the rest of the experiment is conducted as though the entries are unlabeled.

3 Processing the Data

In order to train our models, we first need to prepare our data. The two datasets that we were working with have already been combined into one, and the labels they used were integrated to follow a "1" label for spam, and a "0" label for non-spam. From here though, we still need to pre-process the data considering that it contains many unnecessary tokens and errors that could cause problems later.

The first thing we did was to remove duplicate rows which may cause some words to be overrepresented in our analysis. Some of the corpus contained the same data multiple times, so we deleted any duplicate rows that we found so that there were only one copy of each data entry.

We also have to clean the actual text in each entry as well to make sure that we aren't counting the same words with different capitalization or other minor changes differently. It's also necessary for us to remove random or unorthodox spacing and extra characters. To ensure we don't run into issues with this, we used regular expressions to remove punctuation, extra spaces, new line characters, and any other unnecessary additions to the words.

From there we removed any URLs from the text as well since these do not provide any necessary data for our analysis. URLs contents could provide insight into the contents of certain messages, but for our purposes we are only focusing on the actual words of the message, and not any additional content.

Finally, we made sure to save our processed data to a new file that only contains our cleaned data and its label ("0" or "1") as the two important columns. With this, our data is fully processed and able to be used in training our Word2Vec models.

4 Training and Classifying Word Embeddings

After preparing the dataset, we can now start training our models to observe the embedding spaces of each of these classes. Specifically, we used the Gensim Skip-Gram implementation of Word2Vec for this experiment.

Now given the full dataset, before we can train our models we have to split them into the two classes. We use the labeled data for this task, and we split up our entries into two separate dataset - one for spam, and one for non-spam.

For each of these classes, we trained an separate Word2Vec model (using the Skip-Gram Gensim application) to observe its embedding space. The spam model was trained only on the labeled spam subset, and the non-spam model was trained only with the labeled non-spam subset. This resulted in two distinct embedding spaces, each characterizing the unique language distribution and semantic structure of its respective class.

Given these two embedding spaces, we can now analyze the word neighborhoods of each class - that is, the sets of words that appear closest to a given word in the vector space. These neighborhoods help us compare how certain key terms are used differently across the two registers. For example, the word "offer" may cluster with persuasive terms in spam but appear with professional terms in non-spam. In comparing these "neighborhoods" we can gain valuable insight into the differences between spam vs. non-spam language.

For the sake of easier visualization, we also created a few graphs showing the neighborhoods of the top words in the corpus for each different register. For each of these top words, we see ten of its closest neighbors.

5 Analyzing the Results and Model

With our models trained, we can now analyze the resulting two embedding spaces of our registers.

In taking the top observed words in the dataset, and looking at their neighborhoods in vector space, we can make inferences about the language usage between the different classes.

One of the words that is heavily observed in this dataset is the word "free". This is also a word that one would likely have many specific associations with it when in association with spam messages. We can see the specific neighborhood of this word for the spam register as shown below in Figure 2.

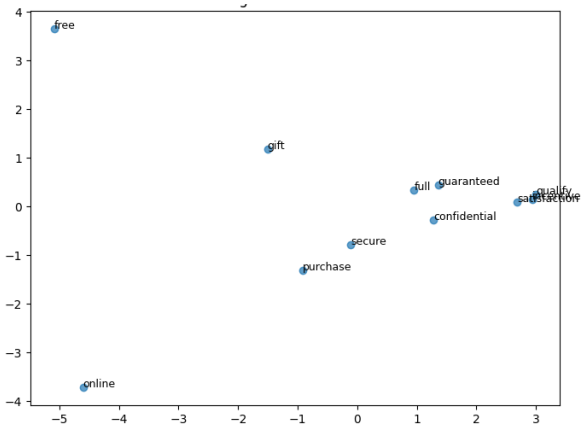

Figure 2: Observed neighborhood of "free" in the spam register

In comparison, we can also see the neighborhood distribution of the word "free" for the non-spam register as shown below in Figure 3.

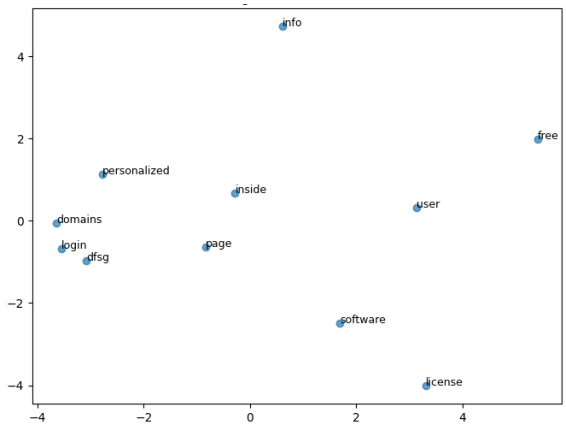

Figure 3: Observed neighborhood of "free" in the non-spam register

The words we see in these figures give us a look into the semantic distributions of the two embedding spaces. Of course this is only one example out of all the possible tokens contained in our corpus, but we can already see how the words related to spam or not can cluster or be distributed.

On the figure showing the "spam" distribution, we see that the nearest neighbors are words like "guaranteed", "satisfaction", and "qualify". These words are often associated with the type of persuasive language we see used in sales in order to market a product or service. This makes sense to be associated with spam messages as their purpose will often be in getting you to click on a link, sign up for a service, or buy a certain product.

On the other hand, in the figure showing the "non-spam" distribution, we see words such as "license", "software", and "domains", which could be more so associated with professional language in businesses. This makes sense as well to be associated with non-spam messages, which are likely to be legitimate people discussing more relevant or professional topics. This holds especially true given our corpus which contains company emails as its main data source.

While there is some variation across other words, these general patterns hold true for each register. We see more attention-grabbing and marketing based language in relation to spam, and more professional language in relation to non-spam.

Our approach here allowed us to find useful insights into the embedding spaces of these types of messages. These kinds of patterns can also help us better identify and correct algorithms to capture these different classes of messages.

6 Limitations

While we were able to gather valuable data from this experiment, there are certain ways that we could improve our accuracy and analysis further.

As mentioned previously, this corpus is based on solely company emails, which leaves out other possible corpora that contains spam messages. Many sites and apps contain possible spam messages as well including social media like Instagram, Twitter, Reddit, and more. With the corpus we chose here, these other datasets are not represented, and could possibly result in different outcomes. In our findings, much of the language that we found related to "non-spam" was very professional, which would likely not be the case if we included social media data.

Exploring this same type of experiment with social media sites could prove to be more difficult considering that the language might be harder to differentiate between. However, it is still important to observe this type of data to identify spam for

other types of services and websites.

Another way that this experiment could be improved is to use email data that isn't necessarily for a company or business. Unfortunately this is a bit hard to do considering we don't have access to personal email accounts, but if there were a good amount of data to observe this, we could greatly benefit from it.

7 Future Exploration

In the future, there are multiple ways we could take this experiment to gain further insights and information on this topic.

The goal of this project was to compare the embedding spaces of spam and non-spam messages, however, in training models to identify the word embedding spaces of our two classes, we were able to discover patterns in the types of language that identify each register. Since we have already observed said patterns, we could take this project a step further in making a model to classify new data that we didn't train it on.

Furthermore, as touched on in the limitations section, we could also apply the same methods to different types of corpora to observe different varieties of spam messages. This includes social media sites, apps, and other services that allow you to send and receive text. Again, any other new data that we could use to answer our question could be beneficial.

8 Conclusion

This study looks at how Word2Vec embeddings can be used to uncover linguistic patterns in spam and non-spam email messages. By training separate models on each class, we were able to observe meaningful differences in the semantic neighborhoods of commonly used words in this corpus. These differences reflect the underlying tone and language of each register—where spam messages tended to cluster around persuasive and promotional language, while non-spam messages leaned more toward professional and informational vocabulary.

The insights gained from these embeddings highlight how distributed representations can serve not only for classification tasks, but also for deeper linguistic analysis of spam and non-spam messages. While our findings are based on company email data, the approach lays the groundwork for further exploration into more diverse sources of spam, including those found on social media and messaging platforms. Future work may focus on expanding the corpus, improving model generalization, and applying these patterns to train more robust spam identification systems.

References

Thomas R. Lynam Gordon V. Cormack. 2007. Trec 2007 public corpus. https://plg.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/foo07. Accessed: 2025-04-07.

Paliouras G. Metsis V., Androutsopoulos I. 2006. Enron-spam dataset. https://www2.aueb.gr/users/ion/data/enron-spam/. Accessed: 2025-04-07.

Puru Singhvi. 2024. Spam email classification dataset. https://www.kaggle.com/datasets/purusinghvi/email-spam-classification-dataset?resource=download. Accessed: 2025-04-07.