1. Introduction

Spam is unsolicited e-mail on the Internet. (E-mail that is wanted is mostly referred to as ham). From the sender's point-of-view, spam is a form of bulk mail, often sent to a list obtained from a spam bot or to a list obtained by companies that specialize in creating e-mail distribution lists. To the receiver, it usually seems like junk e-mail.

Many email spam messages are commercial by nature. They may also contain disguised links that appear to be familiar websites but in fact lead to phishing websites or sites that are hosting malware. Spam email may also include malware as scripts or other executable file attachments. With the advent of bots and improved bot technology the use of the word "bulk" has become deprecated and the need for an “automatic” means to avoid being the end target has dramatically increased.

In a nutshell, spam has evolved from an annoyance to a criminal enterprise, it has, as a headline recently noted, taken a journey from hobby to profit-driven attack. Spam started out, just as email did, as a thought experiment (Question: Can I do this? Answer: yes). Email systems were set up initially between two computers in the same room, then colleagues on the same floor, then the same campus, and ultimately on the same network. In essence, societal norms and peer pressure were the crowd-sourcing required to keep the network clean. Everyone knew everyone else, and transgressions for violating community norms were rapid and severe.

When Gary Thuerek sent the first spam message touting a new DEC computer system to ARPANET users in 1978, the backlash was instantaneous, and it took years before another spam incident occurred. But, occur it did and occur it will for Amharic in the future. This is something that we should address before Amharic emails become even more popular and the problem gets out of hand.

2. Objective

The use of a spam filter programs in our mail servers keeps us from getting this unwanted bunch of emails. It saves us a lot of time and makes work less stressful by minimizing unread emails in our inboxes. Many spam filters have been made and modified over and over throughout the years. A lot of experiments has been done to get to where we are in the realm of detecting spams now. This experiments and improvements have mainly focused on the English language.

There hasn’t been that much of work on other languages specially our very own Amharic. We have tested sending Amharic and Spanish spam emails to each other using Zimbra8,

Google and Yahoo and there was no detection at all. As the world is adopting to our written language over the web, the use of Amharic has been growing steadily throughout the years. It's very obvious to assume that people will start emailing in their own language more frequently than ever and more easily. Thus it's a matter of time before we start getting Amharic spam mails telling us "እንኳን ደስ አልዎት። የ ፩ ሚልዮን ብር አሸናፊ ሆነዋል። ገንዘብዎን ለመቀበል እባክዎ እዚህ ጠቅ ያድርጉ።"

These days, one can't open their email without seeing countless spam messages in their inbox. For the email-recipient, spam is easily recognized. However, the receiver of spam loses countless hours manually deleting the intrusive messages from their inbox. Spam filter software can help mitigate this overwhelming chore, reducing the amount of junk mail delivered to a user's inbox.

The content of spam email can range from the incomprehensible to the downright obscene. Spam is dangerous to both the computer and its users. Junk mail can contain viruses, key loggers, phishing attacks and more. These types of malware can comprise a user's sensitive private data by capturing bank account information, usernames and passwords. Spam blocker applications can assist a user in preventing these types of PC contaminations. Certainly, there are advantages to using a spam blocker. Spam filter software can also assist parents in blocking email that contains pornography and other questionable content.

Clearly, a war is waging inside a user's inbox. The battle to stop spam is an ongoing ever-changing fight. When it comes to Amharic the battle has not "started" yet but we believe it will someday. Implementing spam filter software is a good first protective step. No spam filter software is 100% effective. But despite its limitations, it can help a user create a solid wall of defense that only lets wanted emails into their inbox.

3. Methodology

Spam filtering is a classic example of a binary classification task familiar to anybody who has ever used email services. The task is to distinguish between two types of emails, *"spam"* and "non-spam" often called *"ham"*. We have introduced our own algorithm for this classification. A machine learning classifier can detect that an email is spam if it is characterized by certain features. The textual content of the email – words like "Viagra", "bank", "lottery" or "password" is crucial in spam detection and offers some of the strongest cues. Our works involved two major parts. First, we needed to train the classifier to find such kind of spam trigger words. And then we'll look for those trigger words in an incoming email to make the decision of pushing it to the user's spam folder. To do this, our spam detection algorithm involves the following 5 steps.

3.1. Loading Data

We have used a supervised statistical learning algorithm for collecting Amharic spam trigger words. A collection of spam – ham (not spam) email pairs will be inputs of this step. In a real world implementation of our system, a mail server will collect emails marked as spam by users as inputs to our system, along with a random ham email pair assigned to them. For our experiment, we've used google translated (English to Amharic) emails and saved them as text files to be read by our program. We've selected the most common English spam emails from a popular publicly available spam archive (http://untroubled.org/spam). We've translated 100 emails for our experiment and saved them as utf-8 text files in our program directory to be used as a training corpus. The main activities in this step:

- Loading spam emails
- Loading ham emails

3.2. Preprocessing (Morphological analyzer)

To be able to use the words in these emails as features for our classifier, we needed to preprocess the data and normalize it (so that different forms of the same word are treated as the same word). We combined the 50 spam and 50 ham emails separately, removed common punctuation marks and splitted the words by spaces. Then we made a separate list for both and morphologically analyzed (stemmed) each word in both texts.

The tool that we used for stemming was HornMorpho, Morphological analysis and generation of Amharic, Oromo, and Tigrinya (https://github.com/fgaim/HornMorpho). We chose the tool because it let us create our own list of prefixes and postfixes to use as rules for implementation of our word analyzer function.

Finally, we saved our stemmed (word by word) plus concatenated spam and ham emails into two separate texts which are ready to be used by the classifier.

3.3. Extracting Features

We needed to extract three parameters for our spam/ham classifier.

- Unique words. We made a list of all words that are found in at least one of the two texts.
- frequency count of every unique word in the analyzed spam text and

- frequency count of the every unique word in the analyzed ham text.

3.4. Training the Classifier

Once we have a list of unique words along with their respective counts in the collected spam and ham texts, we needed to create a simple parameter/weight that can measure the probability of finding a word in a spam email. We decided to go with spam to ham frequency count ratio of the unique words for our implementation. We haven't removed stop words from the texts during preprocessing and using a ratio will make up for that. That is also why we include a pair ham email for every spam email input to our system. We used a frequency ratio of 15 or above for classifying a unique words as a spam trigger word. This value accounts of the size of our manually selected and somewhat biased training corpus and was used in our implementation after weighing out the errors caused due to over-stemming.

3.5. Checking Incoming Emails

After making a list of spam trigger words that were outputs of our classifier, we check if an incoming email contains any of them and label it as spam or not. A spam email can therefore be rejected or forwarded straight to a spam folder of a user. The more spam-marked emails the mail server gets with time (users marking emails as spam), the more efficient and inclusive the program will be. For our experiment we've included a few email texts in our program directory. Our program will take their text filenames as input and check the emails for spam words.

3.6. Tools Used

The major tools used are Python 3.4.0 , Notepad, Google Translator, Microsoft word. Steps used to implement code on python 3.4.0.

1. Download python 3.4.0 from https://www.python.org/download/releases/3.4.0/
2. Install the HornMorpho package to python.
   a. Download the files from https://github.com/fgaim/HornMorpho
   b. Extract the downloaded zip file
   c. Open and run setup.py

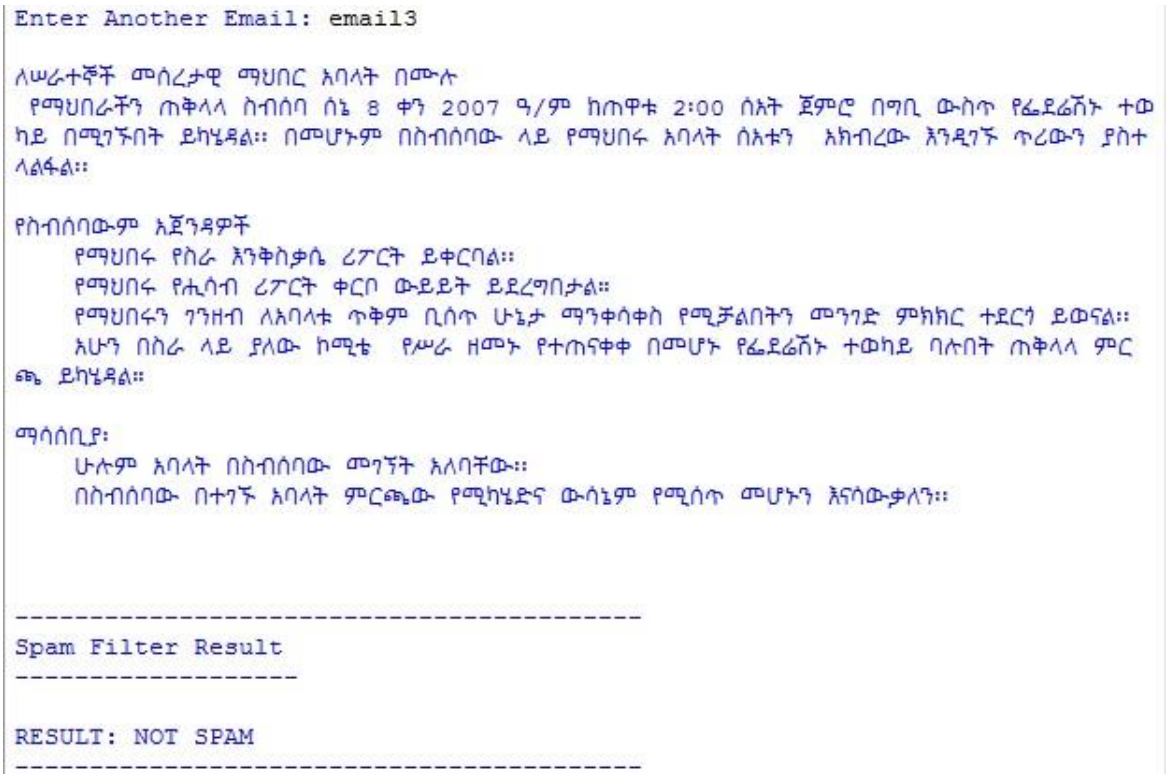
4. Results and Evaluation

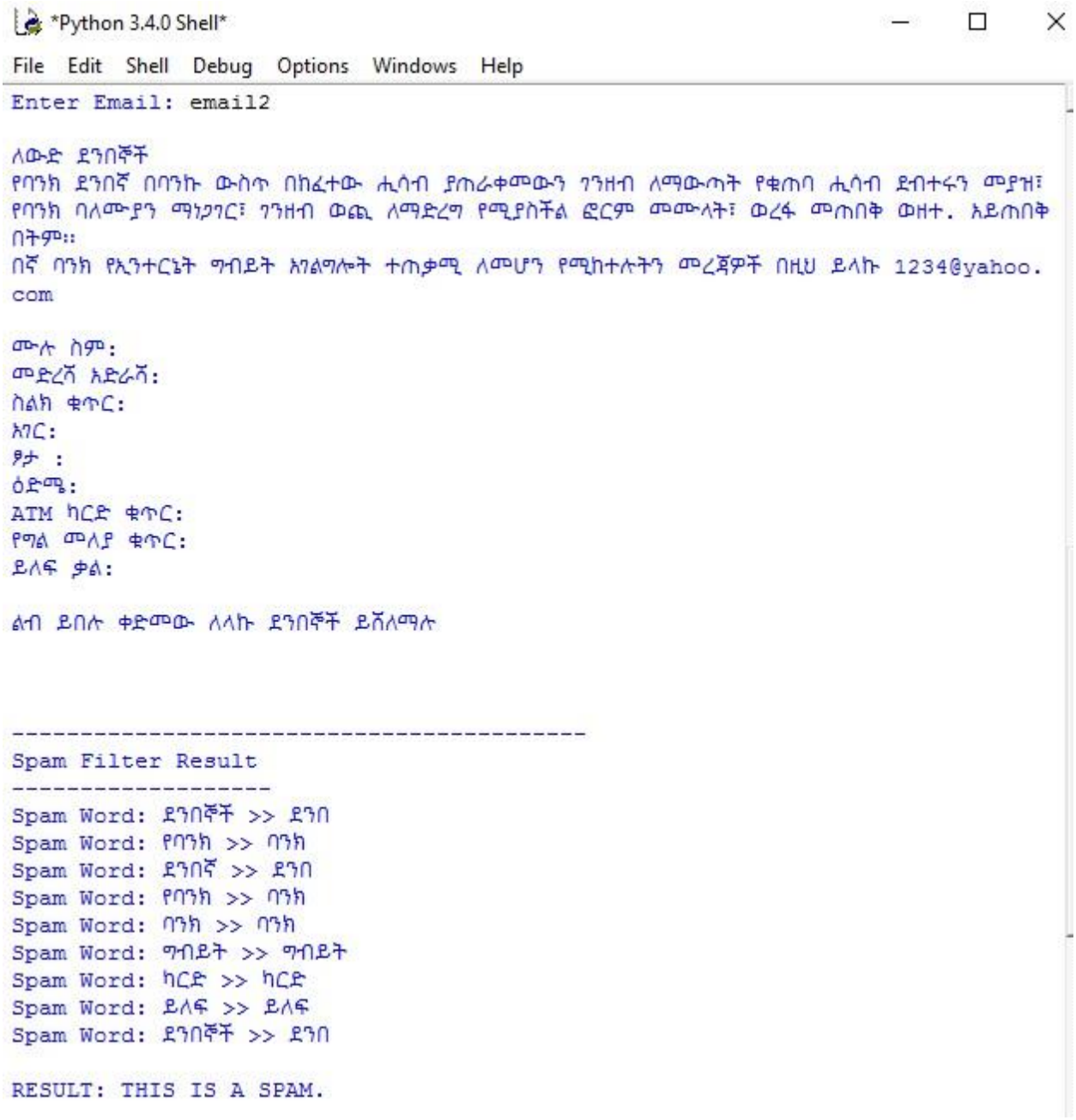
The results of our classifier output was satisfying enough. It marked 27 words as spam trigger words. They are listed below along with their frequency ratio and the English words we believe they were translated from.

Freq. Ratio	Spam Trigger Words Amharic	English
59	ደንበ	Subscriber
53	ባንክ	Bank
46	ተሳካ	Successful
39	ሚሊዮን	Million
39	ያሸልማል	rewarded
30	ደብዳቤ	Letter
28	ላኪ	sender
27	ስመ	Name
26	ሜል	Mail
26	ይመስላል	Feel like
23.5	መስመር	Line
21.5	አድራሻ	Address
21	ወሲብ	Sex
21	ያልጠበቀ	Unexpected?

Freq. Ratio	Spam Trigger Words Amharic	Amharic
20	ሸፍጥ	
19	ይለፍ	Pass(from password)
19	ጠቅ	Click
18.25	መልዕክት	Message
18	ካርድ	Card(from ATM card)
18	-ሜል	-mail (from e-mail)
18	ራስጌ	Header
17	ግብይት	Transaction
17	ርዕሰ	Title
16	መላክ	Sending
16	ራሱ	
16	የ	
15.5	ማረጋገጥ	Confirm

The following images show output (results) of checking incoming emails using the above spam trigger results.





We've tested the result with random emails and the outcome was satisfying. Considering the small sized corpus we were able to develop for training, we hope it'll deliver even better results when implemented in a mail system and gets a chance to have a bigger list of spam-marked emails.

A few words have been wrongly stemmed and we will need a better morphological analyzer for future versions of the application. Most of the words in the list can also be used

in a different context in ham emails and that will result in getting important emails rejected or moved to a spam folder.

5. Conclusion and Feature Works

It's definitely a work in progress and can wrongly mark many emails emails as spam but we believe it's a good start. The method we use now uses a unigram word by word detection method which for a starter could be somehow acceptable just for now. The spam trigger words could be used in a different context to what got them in to many spam emails so using word for word detection could have result in labeling of many ham emails as spams. This is a very huge misfortune that needs to be tackled in our future works. We will plan for implementing n-gram detection to include searching and matching of phrases and sentences to come up with a better algorithm in the future.

Our binary classifier also uses a simple ratio. We'll do an experimental research and implement a more suited probability formula like "Naïve Bayes algorithm of binary classification" for our future versions. We also think it would be more efficient to use a combination of conditions as a trigger instead of just one word/phrase matching.

A larger Amharic spam and ham email archive is needed for an efficient machine training. Amharic emails are not that popular yet and we don't have a public spam archive in our country to train and test with a bigger corpus so that will be another challenge we'll need to tackle.

6. Resources

https://www.python.org/download/releases/3.4.0/
https://wiki.python.org/moin/BeginnersGuide
https://github.com/fgaim/HornMorpho
http://stackoverflow.com/
http://untroubled.org/spam
https://en.wikipedia.org/wiki/Amharic
http://www.eruxelf.com.et/rdamamorph.htm
http://untroubled.org/spam/