Case Discussion

Lexical Analysis- Spam Detection

Divya Kundra and Ashish Sureka, *An Experience Report on Teaching Compiler Design Concepts using Case-Based and Project-Based Learning Approaches*, International Conference on Technology for Education (T4E 2016)

As the use of email is indispensable nowadays, an American company wants to come up with a new email service called 'mails.com'. The developers want to make it better than the present email services like Gmail, Yahoo etc. Now a days it is seen that email is perfect way to send advertisements to millions of people at no cost. This fact is exploited by organizations. Email boxes of millions of people thus get cluttered with "spam" or "junk mail". Spam is the use of electronic-messaging systems to send unsolicited bulk messages, especially advertising, indiscriminately. Uninvited and unwanted emails called 'spams' waste a lot of network bandwidth. The problem of email spam is growing tremendously in the present email services. Thus the developers of mails.com wants to make their new email service fully robust to the spams. To achieve this, they plan to use a spam filter which automatically detects and removes spam as it is received. Spam Filters work by analyzing email's content in some way to decide whether it is ham (i.e. legitimate email) or spam. Tokenization plays an important role in identifying spam mails. The fundamental goal of tokenization is to separate and identify specific features of the text based upon some predefined rules. Tokens are identified as the sequence of characters that would obey the predefined pattern or rule set by the filter. Pattern matching extends the idea of keyword searches to use regular expressions. Instead of searching for a particular word or phrase, pattern matching searches an email for a pattern which matches the specified regular expression. This is a much more powerful technique. Regular expressions are less easy to get around than keywords, but are able to identify words that are obscured by misspelling or other tricks. Advantage of regular expression over term matching is that regular expression allow for matching more complex patterns of multiple terms. The concern of developers of mails.com is that while pattern matching is much more effective at identifying spam than keyword or key-phrase matching, no single message characteristic can be relied upon to consistently distinguish spam from legitimate email, and they must be updated regularly as spammers adapt to them and employ different techniques. From the already available mail services, it is observed that the risk of false positives is very high. Thus for mails.com the developers wish to carefully craft the regular expressions. Building up the correct regular expression will in turn lead to tokenization process doing correct feature extraction in detecting spams from hams. The results returned by tokenization process is passed further to analysis phase. A simple change in tokenization can tremendously affect the accuracy of the filter.

So for the spam detection mails.com wishes to make a filter which will detect the spam as soon as it arrives in the mailbox and will inform the user about it. The filter would consists of thousands of pre-defined 'spam-rules' against which the email content will be compared. Anything matching to 'spam-rules' would categorizes to be a spam component. The developers know that as spam filters is evolved to better classify spam, spammers will adapt their messages to

avoid detection. Thus to build the rules that will be most effective, the developers of mails.com begins to observe closely the kinds of spam attacks on filters that has happened in past. Developers notice that most usually spammers use obfuscation to fool the rules to avoid detection. Example as statistical spam filters began to learn that words like 'offer' mostly occur in spam, and starts to think 'offer' as 'spam-rule', spammers began to obfuscate them with spaces and punctuation, such as 'o.f.f.e.r'. The regular expression for 'offer' fails over 'o.f.f.e.r' thus the spam component gets ignored. Most spammers commonly use punctuation marks to disguise words that really trigger detection like instead of using "Low Mortgage Rates" in subject line, spammers might use "Low Mort!gage Rat-es. The developers of mails.com believe that it is impossible to capture all possible variations and misspellings as spammers will obfuscate trigger words with deliberate misspellings by adding invisible letters to the words and with new tricks every week. Thus the developers want to come up with a good plan to counter such type of situation. In some scenarios, spammers also use the technique of "composite" words it concatenates small number of words to form longer words. Examples of such words are "freepictures", "downloadvideo", "freemp3", etc. Thus even if the rules asks filter to look for "free" as spam, the word "freemp3" would not get captured as spam. While studying the spam attacks, developers made an observation that spammers love to SHOUT about their wares. They even make a combination of lower and upper case letters to pass the statistical filter. For example, SpAm and sPaM might not be comparative for a statistical filter thus these can be used by spammers to successfully pass the filter. Another interesting observation made by developers is that to try and avoid simple filtering system spammers will often transpose letters in words or even miss out the vowels. The human brain is amazing. It can read text even when it's seriously obfuscated. Let's look at an example:

Take a look at this paragraph. Can you read what it says? All the letters have been jumbled (mixed). Only the first and last letter of each word is in the right place:

"Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr theltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at therghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe."

Thus in such a case the entire contents of the message can pass through the filter as it is even after containing large spam content. Another category of attack observed by developers is that of the one in which spammers uses symbols to represent letters. For example, l33t is very popular on the Internet and most if not all people can read and understand it. A prominent form of obfuscation is the utilization of leetspeak. Leetspeak or leet (usually written as l33t or l337) is argot primarily used on internet. The leet speech uses various combinations of alphanumeric characters to replace proper letters. Typical replacements are '4' for 'A', '8' or '13' for 'B', '(' for 'C', ')' or '|)' for 'D', '3' for 'E' 'ph' for 'F' etc. Using this offer would be written as Off3r which would not be understood by the filter. It's just one more way to obfuscate text in an attempt to make it hard to classify the content. Thus all of these attacks suggest that spammers only have to change a few letters around or miss out a few vowels and the tokenization rules would recognize the tokens incorrectly. Developers also concluded the fact that as well as purposeful obfuscation, spam is often riddled with poor grammar and spelling. This is because

most spam originates from countries where English may not be their first language. Spammers want to target the widest audience possible and whilst it would be wrong to assume all spam is in English it is probably not wrong to assume a majority of it will be. Other type of attack as observed in the past consists of addition of some real random words in the message but not letting the user to see the words. For instance doing adding a random words before HTML, writing colored text on colored background etc. Spammers love to put in URL links into your mail which on click downloads some malware into the system. Spammers will often try and obfuscate URLs to prevent rules based detection. For example, they may encode the URL or add unnecessary parameters. Another observation made is that the position of keywords that we are looking for does affect the detection by filter. For instance “free Poker at Texas” can be a spam due to close positioning/proximity of word “Texas” and “Poker” but when used in 2 different sentences, the words might “Texas” and “Poker” might not constitute a spam. While doing the research, developers also came across ‘picospam attack’ which consists of appending random words to a short spam message, trying that those random words would be recognized as ‘good words’ by spam filter. Spammers will add more common words to the spam in order to make the overall count of good words greater. Accuracy of filters can thus be decreased in such a case.

Thus after studying the above case help the developers of mails.com in making a good filter to catch the spams. Taking into account the attacks that have been observed by developers, suggest appropriate ways to do tokenization and how rules can be built to achieve maximum accuracy of the filter.