A NOTE ON CONSTRUCTING DOMAIN SPECIFIC STOP WORD LIST

Kavita Ganesan
ganesan.kavita@gmail.com

While it is fairly easy to use a published set of stop words, in many cases, using a published set of stop words is insufficient. For example, in clinical texts, terms like "mcg" "dr." and "patient" occur almost in every document that you come across that these can be regarded as potential stop words. The same thing for tweets where terms like "#" "RT", "@username" can be regarded as stop words. The common language stop word list **DOES NOT** cover such domain specific terms, but it is actually fairly easy to construct your own domain specific stop word list. Here are a few ways of doing it assuming you have a large corpus of text from the domain of interest you can do one or more of the following:

1. MOST FREQUENT TERMS AS STOP WORDS

Sum the term frequencies of each unique word, **w** across all documents in your collection. Sort the terms in descending order of raw term frequency. You can take the top **N** terms to be your stop words. You can also eliminate common English words (using a publish stop list) prior to sorting so that you are sure that you target the domain specific stop words. The benefit of this approach is that it is so easy to implement, the downside however is if you have a particularly long document, the raw term frequency from just a few documents can dominate and cause the term to be at the top. One way to resolve this is to normalize the raw tf using a normalize such as the document length (i.e. number of words in a given document).

2. LEAST FREQUENT TERMS AS STOP WORDS

Just as terms that are extremely frequent could be distracting terms rather than discriminating terms, terms that are extremely **infrequent** may also not be useful for text mining and retrieval. For example the username **"@username"** that occurs only once in a collection of tweets, may not be very useful. Other terms like **"yoMateZ!"** which could be just made-up terms by people again may not be useful for text mining and retrieval. Note that certain terms like **"yaaaaayy!!"** can often be normalized to standard forms such as **"yay"**. However, despite all the normalization if terms still have a term frequency count of one you could remove it. This could significantly reduce your overall feature space.

3. TERMS THAT OCCUR FREQUENTLY ACROSS DOCUMENTS – LOW IDF TERMS AS STOP WORDS

Inverse document frequency (IDF) basically refers to how many documents in your collection contain a specific term. Let us say you have N documents. And term t_i occurred in M of the N documents. Therefore, IDF of t_i is thus computed as:

$$\mathbf{IDF(t_i)=Log\ N/M}$$

So the more documents t_i appears in, the lower the IDF score. This means terms that appear in each and every document will have an IDF score of 0. If you rank each t_i in your collection by its IDF score in descending order, you can treat the **bottom K terms** with the lowest IDF scores to be your stop words. Again, you can also eliminate

common English words (using a publish stop list) prior to sorting so that you are sure that you target the domain specific low IDF words.

WOULD STOP WORDS HELP MY TASK?

So how would you know if removing domain specific stop words would be helpful in your case? Easy, test it on a subset of your data. See if whatever measure of accuracy and performance **improves, stays constant or degrades**. If it degrades, needless to say, don't do it unless the degradation is negligible and you see gains in other forms such as decrease in size of model, ability to process things in memory, etc. If you see a significant drop in performance you may also want to try a minimal stop word removal approach.