COLUMBIA UNIVERSITY – CUNY

Homework 1 - NLP/ML/Web

Jessica Ouyang and Joe Ellis

October 7, 2013

1 LINGUISTIC ANALYSIS OF WORD REPRESENTATIONS

Document Classification is an interesting and widely studied task in the field of Natural Language Processing. It allows for unique challenges based on the type of classification that is attempted, and many open questions still exist. We will be discussing in this section different word representations, and how they effect the performance of given classification tasks. Document Classification is different than many computer vision tasks, because where as SIFT features are in many ways the one-feature-fits-all prototype for almost all computer vision recognition and classification tasks, different document classification tasks require very separate feature and word representations. For example, general document topic classification may be helped with the removal of stopwords, because they have very little to do with the topic of a given document. However, for author identification the stopwords can be useful grammatical choices that the author makes, and can therefore be very useful.

PRE-PROCESSING STEPS We will begin by discussing the topic of pre-processing steps, and how they can be useful for a given document classification tasks. One possible pre-processing step is known as word-stemming. This process takes words like “estimat”, “estimate”, and “estimate” stems them down to the same word-representation. This can be a useful process for dimensionality reduction of a given word representation for a document, and also helps to disambiguate similar words. However, there are also some disadvantages to using off the shelf stemming processes. Stemming is a very difficult problem, and do to the high variability in the English language all similar words are not stemmed down into the same representation. For example, the words “summarize” and “summary”, although simply they are same word in noun and verb form they are rarely stemmed down into the same word, this performance problem can be an issue in some instances. Another common pre-processing step in word representations is stopword removal. Many words such as “about”, “the”, “because”, etc. are widely used throughout the English language and give little to no topic based discrimination between documents. Therefore, these words are commonly removed before creating the word representation for many NLP classification tasks. However, for our task in particular which is chef recipe classification these stop-words may in face be highly useful. The choice between using different conjunctions, and other functional words are stylistic choices made by an author for rhetorical reasons known to them. Many authors also develop patterns within their writing style of using similar words and sentence structures to make their material accessible to a reader, and in this case including functional words in the representation is very important. Pre-processing of the documents before choosing a word representation scheme can have great effect on the outcome of the results.

BAG OF WORDS REPRESENTATION After pre-processing we then must choose a particular document representation for the task at hand. The simplest and one of the most transmittable word representations that can be used is the “bag-of-words” feature representation. This idea has been used widely in a variety of different fields, and is not limited to simply document classification, but is also widely used in Computer Vision as well. “Bag-of-Words” (BoW) simply searches through a document and denotes whether a word is present within the document that is within the word dictionary. This representation has some very nice features, which for one is that it is very intuitive and easy to understand. This is one reason that is has had such high adoption for representation in other fields as well. The BoW features are made up solely of binary values for each word, because they simply denote whether or not a word is present in a document. This allows them to be highly compact and efficiently stored in memory, if memory consumption is a prevalent issue within the system. They are also geometrically very convenient, because each of the dimensions within the vector have the same range. This allows for us to calculate useful distance measures between different points with little to no need for normalization beforehand. Many Machine Learning tasks can be highly dependent on the normalization scheme performed on the data before classification, and therefore using BoW can alleviate some of the dependencies on extra processes. However, one issue that arises with BoW is that this feature representation effectively throws away the count information for the words within each document. For instance, if the word “rice” appears 10 times in a very short recipe it would be safely presumed, that rice is a very important ingredient for this particular recipe. However, using BoW “rice” would have the same numerical value in our document representation as ingredients that only appear one time. By throwing this count data away the BoW representation may be losing valuable information that can be useful in classification.

TERM VECTOR REPRESENTATION Instead of simply denoting if a word was present in a document, we could then instead count the number of times that each word appears in a document and use this integer valued number as our word representation. This representation is known as the term-vector representation, and allows us to capture the frequency data that is lost within BoW representation. However, it does have some disadvantages when compared to BoW. Probably the most obvious disadvantage is that term vector values are integer values, and therefore each value takes up more space in memory than a dimension in the BoW representation. Another possible problem with the term vector representation is that the range of the values within the representation can be very large, and therefore two different representations can be very far away in Euclidean space making classification difficult. One solution is to normalize the term vector in some way, possibly divide each dimension by the sum of all the values in the term vector to normalize every value by the total length of the document. Other normalization schemes exist, and it was probably trial and error could be used to find effective normalization techniques for a given task.

TF-IDF REPRESENTATION One very useful normalization procedure is known as TF-IDF, and can be used on BoW or term vector representations. TF-IDF is a measure of how important and discriminant each word within a dictionary is. To perform TF-IDF normalization we count the number of documents within our collection, and then divide by the number of documents that the word that which we are trying to normalize appears in. We then finally take the logarithm of this value to obtain the IDF score. Then to gain the TF-IDF vector we multiply IDF by whatever the TF (word-representation) for that word may be. The TF could be BoW, Term Vector, or some other type of representation. This representation allows for us to prioritize the words that are discriminant across the document sets, and then downgrade those words that appear in every document. It is a very useful strategy, and this technique is also widely used in the field of Computer Vision. I have personally often used TF-IDF in CV tasks to great effect, and I believe that this representation is one of the most useful and intuitive document representations available to us.