CIS399: The Science of Data Ethics **Lecture 12: Feb 26, 2019**
Instructor: Michael Kearns & Ani Nenkova **Scribes: Sierra Mills & Allison Domm**

1 Book Club Information

The plan is to read books that give a more nuanced and detailed view of data handling and privacy to complement topics from class. We will read the books in groups and present in class. The main point to be highlighted in the books is the impact of predictive technology on the topic the book covers. Discussion boards on piazza are to be used to talk with your group about the book throughout the reading. The TA's and Professors will be checking up on these boards to make sure that groups are doing work throughout the semester. The discussion board shouldn't replace in person meetings. For in person meetings, someone should take notes on what is discussed and post it to the discussion board so the instructors know what groups were talking about.

2 Word Representations

The news often makes it seem as though word representations are trivial. In reality, most application that deals with language (speech recognition, translation, etc) make use of word representations. If you're doing any predictions, you need to be able to do large scale automatic analysis of language. Language can be used as features for machine learning, but there needs to be a way to represent words and their meanings in a way that computers can interpret.

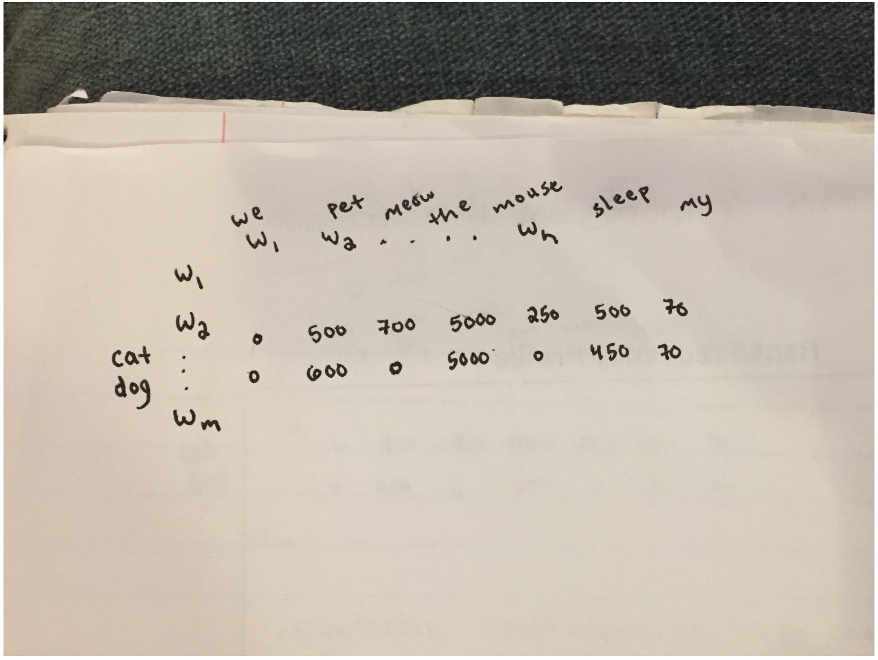
2.1 One-Hot Representation

In one-hot representation, each word is a feature that is on or off depending if text contains the word or not. The vocabulary V is all of the words that appear in the training data. Each document in the training set can be represented as a vector of fixed length where $V[i] = 1$ if word at dimension i appears in the document and $V[i] = 0$ if it doesn't. This way, we have a feature representation of words that appear in the document. There are some concerns that arise from using one-hot representation. First of all, it can be very inefficient. There can be many many words and as the number of total words increases, the number of unique words increases as well. This is inefficient since a word that only appears once is likely not useful. We will not know whether 70 percent of the features are useful or not, because they won't have appeared many times in the training set. One way to deal with this concern is to discard words that appear fewer than 5 times. This, however, exposes another problem with one-hot representation. The test set is always going to include words that don't appear in the training set. Further more, a word that appears less than 5 times in the training set may appear a lot in the test set. The term for words that we don't see during training but can be exposed to later is ”out of vocabulary words” (OOV). Since these words have never been seen before, the model will not know how to deal with them.
In the graphs shown above, the lower diagram can be represented by the equation $V > c\sqrt{n}$ where $c = 10$ and n is the number of distinct words existing in the training data.

2.2 Word Embeddings

Words are first represented in a space that is defined by the same words (they are both the object and the feature). The counts of words are more robust because we can see overlapping dimensions. By looking for co-occurrences of words in documents, we can find similarities between words and figure out what words might mean. Words that are nearby each other in documents often have a syntactic relationship. We want the representation we use to come from some pre-learned relationship between words. We can determine that words are similar to one another because of the words they are both associated with (for example, cat and dog are both associated with the word pet). The similarity of x and y is equal to $(x_1y_1 + x_2y_2 + ... + x_ny_n)/(\sum i = 1nx_i^2 * \sum i = 1ny_i^2$. We don't want to have to do this for all x and y however, because that would take way too long and cause some of the same problems as one-hot representation. We need to run dimensionality reductions to reduce the times we need to preform this operation.

In order to visualize the process of finding syntactic relationships between words, we can draw a table similar to the one shown below. In this table we have m rows and n columns, where each row and column is associated with a different word, w_i. We calculate the number of times that words w_i and w_j appear within five words of each other in the training set, and we place this value at the index $[i, j]$ in the table where words w_i and w_j intercept. The higher the value at index $[i, j]$, the morrer syntactically related words w_i and w_j are.



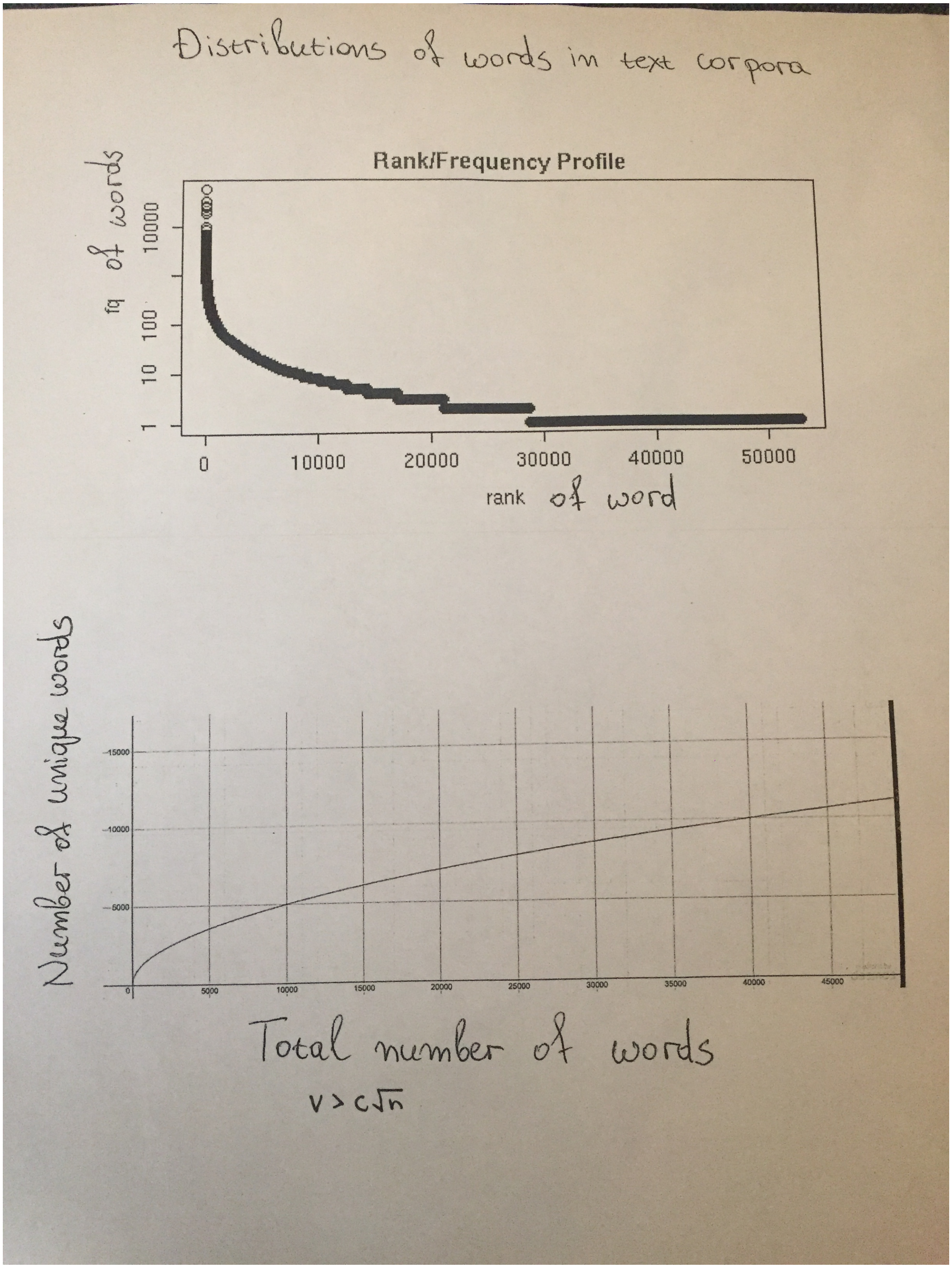
		we	pet	meow	the	mouse	sleep	my
		w_1	w_2	. .	. .	w_n		
	w_1							
	w_2							
cat	:	0	500	700	5000	250	500	70
dog	:	0	600	0	5000	0	450	70
	w_m							

If a word occurs within 5 words of the other, put the count in the spot. If not, put 0.

2.3 Implicit Association Test (as discussed in the lecture 12 reading from Science Magazine)

The implicit association test asks players to map words in categories to either "pleasant" or "unpleasant." The test looks at the average time it takes someone to sort objects into categories. It first asks to associate things that are usually associated with the word pleasant, like types of flowers, to the word pleasant, and words that are typically thought of as unpleasant, like bugs, to the word unpleasant. It later asks to do the opposite, and map things like bugs to pleasant and flowers to unpleasant. In this case, we consider the category and the object to be ”misfit,” which implies that it will take longer for the subject to connect the two. Computers played the same game, and the results were very similar, meaning that computers show just as much bias towards certain words as humans do. The implicit association test exemplifies a potential problem with word representations: if we let algorithms learn human words, they also learn human biases. For example, word vectors often associate females with jobs like ”secretary” over ”CEO.” This association is particularly prominent in language translation programs. The paper discusses a specific example where someone uses Google translate to translate an English sentence into Turkish and then back to English. When the sentence was originally put into the translator, it was referring to a male nurse and a female doctor. However, once the sentence was translated to Turkish and back into English, it then was referring to a male doctor and male nurse. This implies that the word embedding used in Google translation associate the word ”doctor” with the male gender and the word ”nurse” with the female gender. Google has since started fixing their translations in order to account for gender biases such as this one, but situations like these raise concern of whether or not we are properly addressing biases in word embedding.

Distributions of words in text corpora



$V > c\sqrt{n}$