Vector Space Models for the Digital Humanities

Ben Schmidt, October 25, 2015

Word Embeddings for the digital humanities

Recent advances in vector-space representations of vocabularies have created an extremely interesting set of opportunities for digital humanists. These models, known collectively as word embedding models, may hold nearly as many possibilities for digital humanitists modeling texts as do topic models. Yet although they're gaining some headway, they remain far less used than other methods (such as modeling a text as a network of words based on co-occurrence) that have considerably less flexibility. "As useful as topic modeling" is a large claim, given that topic models are used so widely. DHers use topic models because it seems at least possible that each individual topic can offer a useful operationalization of some basic and real element of humanities vocabulary: *topics* (Blei), *themes* (Jockers), or *discourses* (Underwood/Rhody).[1] The word embedding models offer something slightly more abstract, but equally compelling: a *spatial analogy to relationships between words.* WEMs (to make up for this post a blanket abbreviation for the two major methods)[2] take an entire corpus, and try to encode the various relations between word into a spatial analogue.

A topic model aims to reduce words down some *core meaning* so you can see what each individual document in a library is really about. Effectively, this is about getting rid of words so we can understand documents more clearly. WEMs do nearly the opposite: they try to ignore information about individual documents so that you can better understand the *relationships between words.*

The great interest (http://www.wise.io/tech/five-takeaways-on-the-state-of-natural-language-processing) that WEMs–particularly the initial word embedding model, word2vec, have generated in the machine (http://multithreaded.stitchfix.com/blog/2015/03/11/word-is-worth-a-thousand-vectors/) learning (http://rare-technologies.com/deep-learning-with-word2vec-and-gensim/) world (https://gigaom.com/2013/08/16/were-on-the-cusp-of-deep-learning-for-the-masses-you-can-thank-google-later/) stems from their remarkable performance, compared to previous models, at tasks of simile and analogy. Criticisms of them come because the method does not scale up to examining large-scale syntax very well.[3] For digital humanists, they merit attention because they allow a much richer exploration of the *vocabularies* or *discursive spaces* implied by massive collections of texts than most other reductions out there.

Over a few posts, I'm going to explore WEMs through two models I've trained. I should explain what those are up front. One, `teaching_vectors`, is of 14 million reviews of teachers from RateMyProfessors.com. (I've already produced one visualization of the data here. (http://benschmidt.org/profGender)) The other, `chronam_vectors`, is larger: about 6 million newspaper pages from the NEH/Library of Congress *Chronicling America* project.[4] I'll do another post later that gets a little more into detail about one particularly interesting application, of the gender binary on teacher evaluations.

I also want to make it easier for other digital humanists to explore WEMs, so I've put an R package for exploring WEMs on GitHub. (https://github.com/bmschmidt/wordVectors) You can install it in R by typing `install_github("bmschmidt/wordVectors")`.[5] This package provides a useful syntax with working with WEM outputs, and bundles the original word2vec code so that you can train your own. There is a fairly explicit tutorial at the end for training your own model on a set of cookbooks: at least one person with no

previous experience with R was able to train a big model on 10,000 books using it, so give it a shot.

WEMs for methodological diversity

There's a broader agenda here. I think digital humanists could use a passing acquaintance for more basic methods from machine learning. As I say in my piece for the Debates in the Digital Humanities 2016, one of the most interesting features of the computational side of digital humanities is that mathematical transformations recast the world in interesting ways. We don't need to understand the mechanics of a transformation, but we do need to understand the change it effects. In that piece I use the analogy of sorting. J. W. Ellison (http://www.historyofinformation.com/expanded.php?id=3491) didn't need to know what sorting algorithm IBM was using to create a concordance of the bible: but he did need to understand what "sortedness" is to want to get a bible in the first place.

A useful transformation offers up texts in new lights.[6] But we don't have many generally useful transformations in DH. Sortedness is one. Topic modeling can count as another; so can, if you stretch the definition, the term-document matrix. Beyond that, though, we don't have many *general-purpose* transformations that can be tossed at a variety of texts. The specific implementations of word embedding out there are imperfect. But the basic goals of the transformations are interesting and useful ones to think through.

What's a word embedding model?

So what's the transformation offered by WEMs? Topic models ask "what if all texts could be reduced to a single number of basic vocabularies?" The Fourier transformations asks "what if all temporal phenomena were regular and infinitely repeatable?" In the domain of social phenomena, these questions are almost entirely wrong. The question that word embedding models ask is: what if we could model all relationship between words as spatial ones? Or put another way: how can we reduce words into a field where they are purely defined by their relations? Such a space allows us to do two things. The first, much like topic models, is to think in terms of *similarity*: what words are like other words? How can we learn from those relations? How do unexpected closenesses extend our understanding of a field?

The new word embedding models aim to create a space like this by placing all words in a linear ordering. The exact operations of the new vector-space models aren't always easy to figure out, but it's easy to understand their central goals.

1. Word embedding models try to reflect similarities in usage between words to distances in space.

This is not a particularly new goal. Digital humanists have been doing for a while with network diagrams, with a variety of scatterplots, and so forth. The difference is that the new methods are much more clearly defined, and therefore more useful for exploratory data analysis.

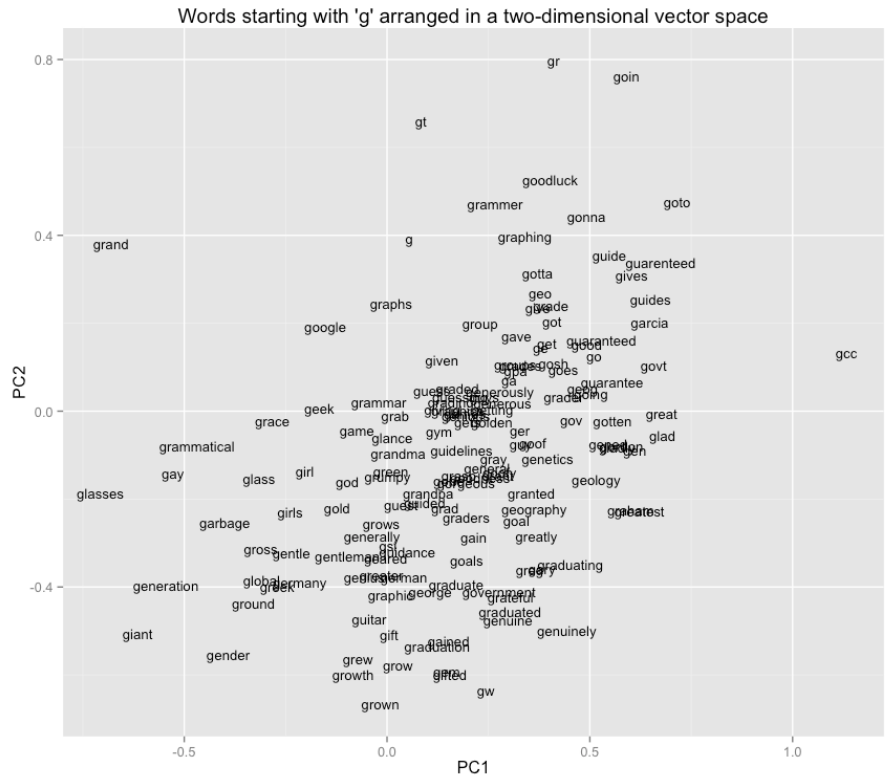
2. Word embedding models try to reflect similar relationships between words with similar paths in space.

We train a model that 'learns' scores for each word in the text for some arbitrary number of characteristics. The number of characteristics that we choose are called the dimensions: each word occupies a distinct point in that broader "space."

Optimally positioning words in space.

Here is a simple example: a plot of words starting with 'g' in a two-dimensional space. On the dimension `PC1`, the word 'grandma' scores -1.1 and 'grandpa' scores -.95; they are similarly close in dimension `PC2`.[7]

A non-optimal positioning of words in space.



If you look at this plot, you'll see that there are a lot of pairings where words with similar meanings are nearby. "Girl" is near "girls" and not all that far from "guy" and "guys"; "gonna" and "gotta", which clearly have something in common, are next to each other; "green" and "gold," the two colors, are relatively near to each other.

On the other hand, there's a lot of junk. Why is "golden" far from "gold"? Maybe you can explain why "grumpy" is between "grandpa" and "grandma," but why "gym?" If you've ever tried to read a network diagram, you've encountered a lot of strange juxtapositions like this, because two dimensions is simply not enough to capture broad relationships. You'd want the closest word to "grandma" to be "grandpa", not "gym." But there's no single system of ranking where that happens automatically.

This isn't entirely the fault of PCA. There more than two or three types of relationships in the world. "Mother" is like "father" except it's female; like "grandmother" except it's a generation removed; like "Mom" except that it's more formal. Each of those express a different type of relationship. The goal of a perfect WEM transformation (something that doesn't exist) is a vector space that can encode all of those relationships, simultaneously.

An R package for creating and exploring WEMs

The biggest is that they don't have the instant gratification of *one single display* that summarizes their contents. I have printed out the top words in each topic of a model and brought it into a classroom: a WEM can't fit onto a piece of paper like that.

At best it lets us reduce the vocabulary down into a two dimensional space like an improved word cloud.[8]

Above is an example of such a plot based on one reduction of the dimensional space. It shows a number of clusters of similar words, though little overall structure. Terms that appear together (like "United" and "States", "May" and "July", or "yesterday" and "today") cluster together on the chart. These plots have advantages over wordclouds, where position is completely meaningless: but they aren't much more than a list of words.