Glove Word Embeddings - Text to Numbers

16:198:530,536

In order to compute on text (for instance, use text as the input to neural networks) we need a way to represent text in a numeric form. An easy way to do this would be to assign every word a number (cat = 1, dog = 2), etc. However, much like general categorical data, this is problematic as it essentially asserts relationships between the elements that may not exist (for instance in this case that cat < dog). A general solution to this problem (applicable to all categorical data) is the idea of *one hot encoding*. If you have N possible values, each value is encoded as a vector of N elements, where each element is zero except for one, which is 1. So you might encode cat as $(1, 0)$ and dog as $(0, 1)$, so the values are literally orthogonal to one another.

This is a good solution, but in some cases becomes infeasible because of the number of possible values is very large - there are tens of thousands of words in the english language, for instance. It would be infeasible to represent each word with an $80,000$-dimensional vector, so that a sentence or block of text becomes a sequence of $80,000$-dimensional vectors.

What we would like is a more compact embedding - but not so compact that we fall back to the original cat = 1, dog = 2 case which is unhelpfully condensed. One way to do this is to try to ensure that our embedding/encoding captures some notion of the important structure of the words and the relationships between them.

But what kind of structure?

Let $X_{i,k}$ be the number of times that word i has occurred 'in the context of' word k - context here meaning some notion of spatial proximity, or other relationship, that you may decide as relevant for your data. We can then say that $X_k = \sum_i X_{i,k}$, the number of total occurrences in the context of k, so that

$$P_{i,k} = \frac{X_{i,k}}{X_k}, \tag{1}$$

estimates the probability of word i occurring in the context of word k.

The 'GloVe' embeddings start with the idea that the important 'structure' that we want our embeddings to capture are the values $P_{i,k}/P_{j,k}$. These ratios express how word i and word j relate to one another through the *probe word* k. If this ratio is ≈ 1, then i and j occur with almost equal frequencies in the context of k, and k provides no useful information about how to distinguish them. If however $P_{i,k}/P_{j,k}$ is particularly large or particularly small, then something about k provides a strong distinguishing effect between i and j.

Our goal then is to find a set of word-vector embeddings that would allow us to recover this ratio information. That is, given $\underline{w}_i, \underline{w}_j, \underline{w}_k$, and some decoder function F, we would like

$$F(\underline{w}_i, \underline{w}_j, \underline{w}_k) = \frac{P_{i,k}}{P_{j,k}}. \tag{2}$$

There are a lot of possible embeddings and decoders F that might satisfy this - we could for instance take F to be a neural network with three vector inputs, and try to solve for a model that predicts the correct output values.

The original GloVe paper proposes limiting ourselves though to certain linear relationships between the vectors, and constrain the above so that we want to find vectors and F such that

$$F((\underline{w}_i - \underline{w}_j).\underline{w}_k) = \frac{P_{i,k}}{P_{j,k}}. \tag{3}$$

Potentially lots of solutions to the above as well, but notice that we would also like

$$F((\underline{w}_j - \underline{w}_i).\underline{w}_k) = \frac{P_{j,k}}{P_{i,k}}. \tag{4}$$

In other words, $F(-x) = 1/F(x)$. One potential solution to this is an exponential function, $F(x) = e^x$. Applying this to the above:

$$F((\underline{w}_i - \underline{w}_j).\underline{w}_k) = \frac{e^{\underline{w}_i.\underline{w}_k}}{e^{\underline{w}_j.\underline{w}_k}} = \frac{P_{i,k}}{P_{j,k}}. \tag{5}$$

. Unpacking this, we've essentially solved for / eliminated F at this point, and would like to solve, for any i, k pair,

$$\underline{w}_i.\underline{w}_k = \ln P_{i,k} \tag{6}$$

At this point, we've essentially reduced the original problem to a straight regression problem, trying to find values of the vectors such that the dot products all encode these log probabilities.

However, observe here the lack of symmetry - we have that $\underline{w}_i.\underline{w}_k = \underline{w}_k.\underline{w}_i$, but $P_{i,k}$ doesn't necessarily equal $P_{k,i}$ (even though $X_{i,k}$ will equal $X_{k,i}$). Unpacking this probability term,

$$\underline{w}_i.\underline{w}_k = \ln X_{i,k} - \ln X_k, \tag{7}$$

we can 'restore symmetry' by absorbing the X_k term into 'bias' terms on the other side, giving us our final embedding problem:

Find a set of vectors $\{\underline{w}\}$ and bias values $\{b\}$ such that for any pair i, k we have

$$\underline{w}_i.\underline{w}_k + b_i + b_k = \ln X_{i,k}. \tag{8}$$

At this point, it is tempting to throw the following optimization problem at your nearest solver: find $\{\underline{w}\}, \{b\}$ to minimize

$$L = \sum_{i,j} \left(\underline{w}_i.\underline{w}_j + b_i + b_j - \ln X_{i,j}\right)^2. \tag{9}$$

But there are a few points worth making further. For instance, notice the numerical singularity if i and j never occur in the same context, giving $X_{i,j} = 0$ and sending the loss function to $-\infty$. This could be corrected by changing the log term to something like $\ln(1 + X_{i,j})$, but we can also note that certain pairs of words should contribute to the overall loss with different weights. If a pair of words never occurs together, we really don't need to worry too much about their embedded relationship. Similarly though, we do not want the loss to be dominated by words that occur together very frequently. Experimentally, the GloVe authors settled on including a weighting function

$$L = \sum_{i,j} w(X_{i,j}) \left(\underline{w}_i.\underline{w}_j + b_i + b_j - \ln X_{i,j}\right)^2, \tag{10}$$

where

$$w(x) = \begin{cases} (x/x_{\max})^\alpha & \text{if } 0 \le x \le x_{\max} \\ 1 & \text{else.} \end{cases} \tag{11}$$

Numerical experiments suggested a good value of α to be about $\alpha = 3/4$.

There are a lot of elements of this formulation that can be potentially tweaked to lead to different final embeddings. Note, for instance, that the dimension of the underlying vector space has not been specified. Additionally the weighting function could differ depending on the specific context of the embedding problem. One thing to think about additionally is what it means for i and j to occur 'in the same context'. It is convenient computationally to just throw down a window of a certain size, and if words occurs within that window say they occur within the same context. This is potentially problematic though as words are often related even if separated in a sentence by a large number of words; however, these longer distance correlations would emerge from analysis of the text once the words of the text have been embedded. You would not expect to capture all nuance and variation possible in a single vector representation. But one potentially more dynamic way of defining context might be to weight the contribution to the total $X_{i,j}$ - if an instance of i and j are separated by d many words, add a contribution of $1/d$ to the total. There is some flexibility to be had here.

Word Analogy Tasks

A classic task for 'word comprehension' is being able to complete analogies. For instance, 'day is to night as black is to X'. Being able to complete these kinds of tasks requires some understanding of how various words relate to one another. Because of the linear structures inherent in the formulation, GloVe embeddings provide a good solution to this task. Given a task 'A is to B as C is to X', we can look at how word A and word B relate to one another by the difference $\underline{w}_A - \underline{w}_B$, and try to find a word that has this same difference with respect to C, i.e., find a D such that $\underline{w}_A - \underline{w}_B = \underline{w}_C - \underline{w}_D$, or in other words find a word D such that its embedding is as close as possible to

$$\underline{w}_C - (\underline{w}_A - \underline{w}_B). \tag{12}$$

And this is surprisingly effective, uncovering relationships like man-woman, king-queen, relationships between countries and their capitols, etc.