Introduction to Word Embeddings

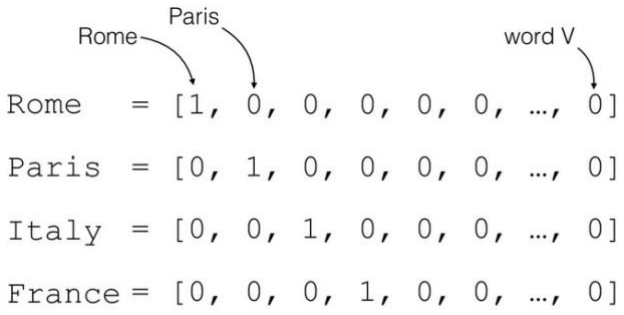
What is a word embedding?

- A very basic definition of a word embedding is a real number, vector representation of a word.
- Typically, these days, words with similar meaning will have vector representations that are close together in the embedding space (though this hasn't always been the case).
- When constructing a word embedding space, typically the goal is to capture some sort of relationship in that space, be it meaning, morphology, context, or some other kind of relationship.
- By encoding word embeddings in a densely populated space, we can represent words numerically in a way that captures them in vectors that have tens or hundreds of dimensions instead of millions (like one-hot encoded vectors).

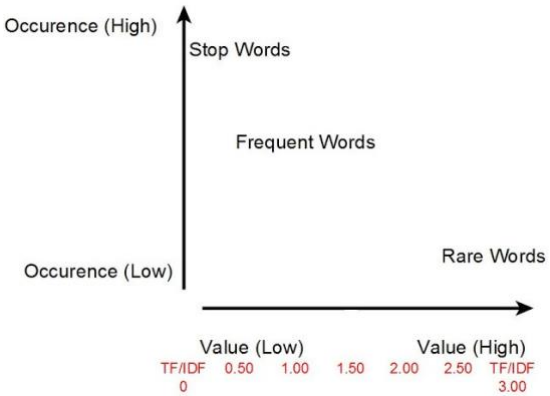
Why do we use word embeddings?

- Words aren't things that computers naturally understand. By encoding them in a numeric form, we can apply mathematical rules and do matrix operations to them.
- Different ways we can numerically represent words
    1. **One-Hot Encoding (Count Vectorizing)**
        - One of the most basic ways we can numerically represent words is through the one-hot encoding method (also sometimes called count vectorizing).
        - The idea is super simple. Create a vector that has as many dimensions as your corpora has unique words.
        - Each unique word has a unique dimension and will be represented by a 1 in that dimension with 0s everywhere else.

          The result of this? Really huge and sparse vectors that capture absolutely no relational information. It could be useful if you have no other option. But we do have other options, if we need that semantic relationship information.



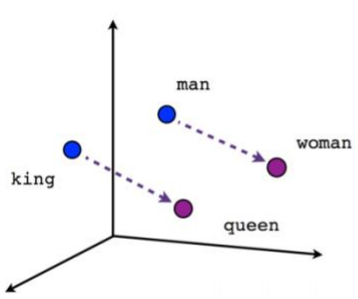
    2. **TF-IDF Transform**
        - TF-IDF are related to one-hot encoded vectors.
        - However, instead of just featuring a count, they feature numerical representations where words aren't just there or not there. Instead, words are represented by their term frequency multiplied by their inverse document frequency.
        - If a word appears very little or appears frequently, but only in one or two places, then these are probably more important words and should be weighted as such.
        - Again, this suffers from the downside of very high dimensional representations that don't capture semantic relatedness.



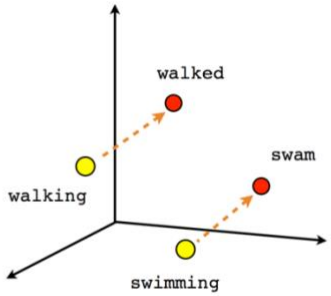
3. **Co-Occurrence Matrix**
   - A co-occurrence matrix is exactly what it sounds like: a giant matrix that is as long and as wide as the vocabulary size.
   - If words occur together, they are marked with a positive entry. Otherwise, they have a 0. It boils down to a numeric representation that simple asks the question of "Do words occur together? If yes, then count this."
   - And what can we already see becoming a big problem? Super large representation! If we thought that one-hot encoding was high dimensional, then co-occurrence is high dimensional squared. That's a lot of data to store in memory.

$$X = \begin{array}{c} \\ I \\ like \\ enjoy \\ deep \\ learning \\ NLP \\ flying \\ . \end{array} \begin{array}{c} \begin{array}{cccccccc} I & like & enjoy & deep & learning & NLP & flying & . \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

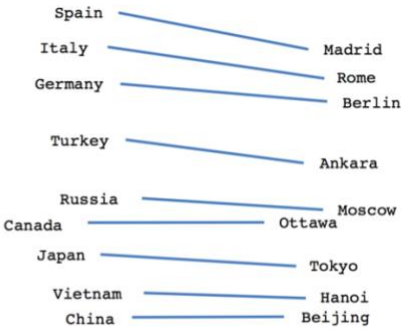
4. **word2vec**
   - Word2Vec is a better successor to the neural probabilistic model.
   - We still use a statistical computation method to learn from a text corpus, however, its method of training is more efficient than just simple embedding training.
   - It is more or less the standard method for training embeddings the days.
   - It is also the first method that demonstrated classic vector arithmetic to create analogies:



Male-Female



Verb tense



Country-Capital