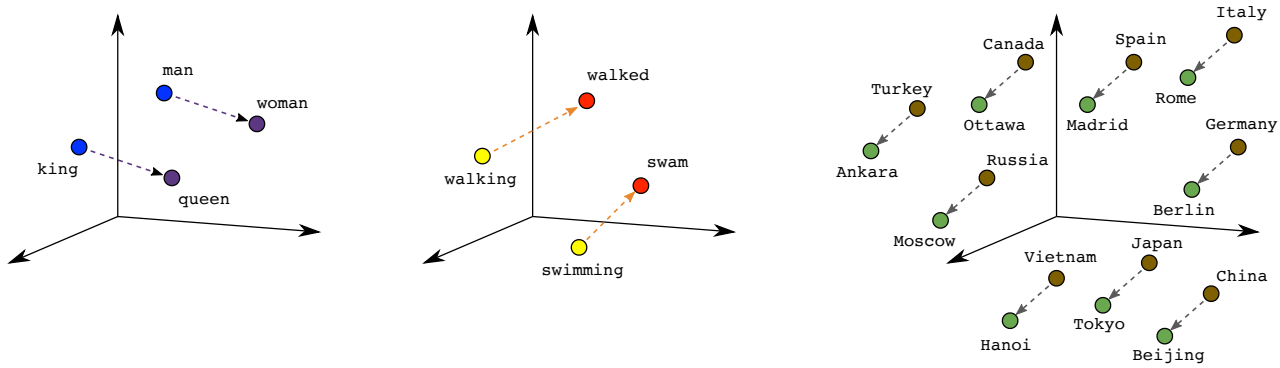
(https://google.com/racialequity?authuser=0)

Embeddings: Translating to a Lower-Dimensional Space

ıted Time: 5 minutes

You can solve the core problems of sparse input data by mapping your high-dimensional data into a lower-dimensional space.

As you can see from the paper exercises, even a small multi-dimensional space provides the freedom to group semantically similar items together and keep dissimilar items far apart. Position (distance and direction) in the vector space can encode semantics in a good embedding. For example, the following visualizations of real embeddings show geometrical relationships that capture semantic relations like the relation between a country and its capital:



Male-Female Verb Tense Country-Capital

Figure 4. Embeddings can produce remarkable analogies.

This sort of meaningful space gives your machine learning system opportunities to detect patterns that may help with the learning task.

Shrinking the network

While we want enough dimensions to encode rich semantic relations, we also want an embedding space that is small enough to allow us to train our system more quickly. A useful embedding may be on the order of hundreds of dimensions. This is likely several orders of magnitude smaller than the size of your vocabulary for a natural language task.

Embeddings as lookup tables

An embedding is a matrix in which each column is the vector that corresponds to an item in your vocabulary. To get the dense vector for a single vocabulary item, you retrieve the column corresponding to that item.

But how would you translate a sparse bag of words vector? To get the dense vector for a sparse vector representing multiple vocabulary items (all the words in a sentence or paragraph, for example), you could retrieve the embedding for each individual item and then add them together.

If the sparse vector contains counts of the vocabulary items, you could multiply each embedding by the count of its corresponding item before adding it to the sum.

These operations may look familiar.

Embedding lookup as matrix multiplication

The lookup, multiplication, and addition procedure we've just described is equivalent to matrix multiplication. Given a 1 X N sparse representation S and an N X M embedding table E, the matrix multiplication S X E gives you the 1 X M dense vector.

But how do you get E in the first place? We'll take a look at how to obtain embeddings in the next section.

ɪrms

ıbeddings (https://developers.google.com/machine-learning/glossary?authuser=0#embeddings)