**Similarity**

How a similar a clothing item is to another will depend on the following 4 attributes of the clothing item

**What gender is the clothing meant for**? Male, female or Unisex.

**What is type and subtype of clothing?** Is it a top wear or bottom wear or accessories or footwear? If its top wear; whether it is shirt, t-shirt or jacket.

**Design aesthetics of the clothing?** Whether it is regular fit or straight fit jeans. How many pockets does it have?

**Fabric of the clothing?** Is the shirt of Linen, cotton or other fabric?

**Size of the Clothing and what age group does the clothing serve?** Whether it’s an XXL shirt or XL shirt

In the given the file, Column Keywords contains information about the clothing type, it’s gender, size, color whereas the description about the clothing design and aesthetics and occasion on which it can be worn. The key word column can be thought as of labels (multi label in our case). The Key word column on its own does not tell anything about the clothing as such except that its meant for specific gender and is of specific color. Just having boot cut jeans does not tell anything that it is bottom-wear or top-wear to the algorithm. Also using Key words column as label column and description as feature to build a classification model to predict labels does not makes sense as we want similar items and not predict exact labels. Although we can predict a clothing item as “Formal Slim Shirt” we cannot find similar shirts (or for that matter Casual shirts, T-shirts and blazers). Thus, it makes sense to concatenate the two columns Keywords and description so that algorithm can learn that “jeans” word is associated with certain words which could be common with other similar items like trousers and cargo pants.

One important thing we need to decide how important each of the above attributes to determine similarity. For e.g. whether Men’s Jeans and Men’s Shirt are more similar, or Men’s Jeans and Women’s Jeans are more similar. Whether dark blue Jeans and dark blue trousers are more similar or dark blue Jeans and grey jeans are more similar. For the given data, we can build separate vector representation for Gender (Can extract from ‘Name’ or ‘KEYWORD’, clothing type and design, size (‘Label’ column) and color (‘Format’ column indicates size. Although the measure of scale is different for different class of items, we can give ordinal representation (label encoding) to each size and then normalize the ordinal numbers so that for each type of clothing item we would the same range. For e.g. S, M, L, XL, XXL could be coded as 1,2,3,4 and 5 respectively and then we can normalize 1 to 5 to 0 to 1 range) and then calculate similarity for each of these representations. Similarity w.r.t color, similarity w.r.t design and clothing type etc. and then assign different weights to each similarity and calculate overall similarity. This approach although is not followed in this exercise.

**Measuring Similarity:**

Once it was decided that item would be characterized by concatenation of keywords (removing sku from keyword column), the next step was to decide how to calculate similarity between texts. Representing text in TF-IDF or other count-based methods, can help to identify a label for the text effectively for e.g. if email is spam or ham. However, this relationship is not semantically or syntactically related, it's just about the level of common occurrence in the textual units to be learnt from. For example, the "similarity" of two words "neural" and "network" (computed by cosine distance) formed from inverted index of a deep learning corpus might be very large (almost 1), meaning they are very similar, but in fact, they are commonly occurred in this kind of corpus, not similar in meaning.

On the other hand, more advanced vector representations of term, such as word2vec or other distributed or distributional representation, are based on the distributional hypothesis (Harris, 1954): you can determine the meaning of a word by looking at its company (its context). If two words occur in a same "position" in two sentences, they are very much related either in semantics or syntactics.

For example, in a big text corpus, there are two sentences: "BMW is a German car manufacturer" and "BMW is a German automobile manufacturer". We, and the computer, can infer "car" and "automobile" are synonyms. Other two sentences: "We will go there this Thursday" and "I will go there this Sunday", Thursday and Sunday have the same syntactic role and they are somehow semantically related (days of the week). Hence, it was decided to use word vector kind of representation.

It was decided to use doc2vec instead of word2vec vector representation as it makes more sense to learn about vector representation of entire description rather than each word if we have to measure similarity between two different descriptions (corresponding to respective skus).

**Testing**

This problem is not a supervised machine learning but unsupervised one. However, to choose whether to chose distributed bag of models or distributed memory model, I decided to select randomly 10 items and calculated their most similar items and manually labelled whether the results were matching or not. Distributed memory gave better performance.