What is Latent Semantic Analysis (LSA)?

LSA and its applications.

Vimarsh Karbhari Follow
Feb 11 · 3 min read ★

Latent Semantic Analysis, or LSA, is one of the basic foundation techniques in topic modeling. It is also used in text summarization, text classification and dimension reduction. It is similar to the cosine similarity. For LSA, we generate a matrix by using the words present in the paragraphs of the document in the corpus. The rows of the matrix will represent the unique words present in each paragraph, and columns represent each paragraph.

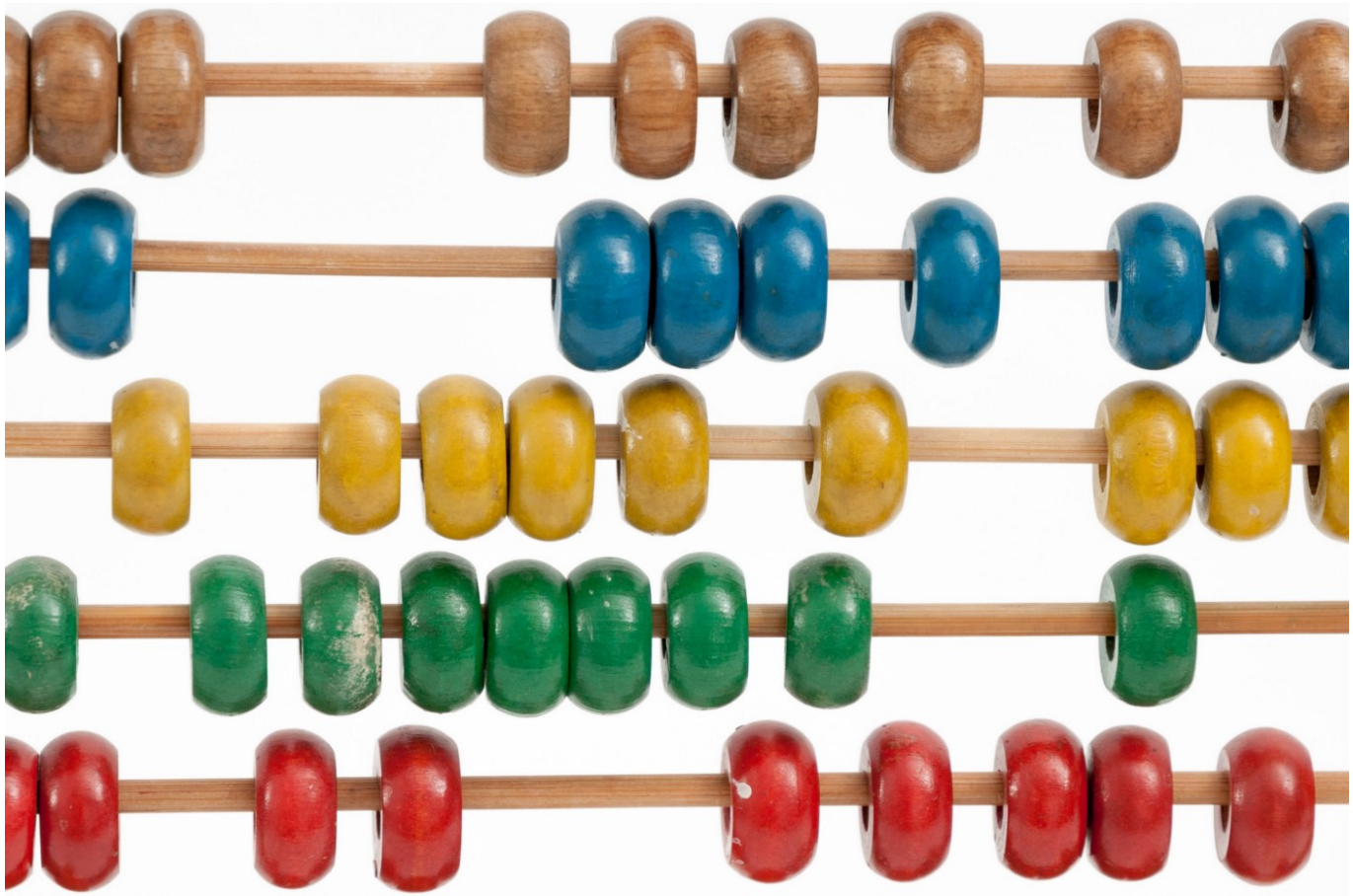

Photo by Crissy Jarvis on Unsplash

The basic assumption for the LSA algorithm is that words that are closer in their meaning will occur in a similar excerpt of the text. Let us consider an example to understand this in detail.

```
#Define Documents
Document_A: Alpine snow winter boots.
Document_B: Snow winter jacket.
Document_C: Alpine winter gloves.
```

	Documents		
Words	Document_A	Document_B	Document_C
alpine	1	0	1
snow	1	1	0
winter	1	1	1
boots	1	0	0
jacket	0	1	0
gloves	0	0	1

As you can see from the preceding example, if we say that the word pair (snow, winter) occurs more frequently, it means that it carries higher semantic meaning than the (alpine, winter) word pair. This is the underlying context of the algorithm.

Consequently, LSA models might typically replace raw counts in the document-term matrix with a tf-idf score.

Usually, once this first level matrix is generated, we do a reduction. LSA learns latent topics by performing a matrix decomposition on the document-term matrix using singular value decomposition.

Singular Value Decomposition

Taking the previous example further, we generate the matrix that is given in the and then try to reduce the number of rows of the matrix by using the single value decomposition (SVD) method. SVD is basically a factorization of the matrix. Here, we are reducing the number of rows (which means the number of words) while preserving the similarity structure among columns (which means paragraphs).

Lets try to implement these using Python and Scikit-Learn.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD
from sklearn.pipeline import Pipelinedocuments = ["Document_A.txt",
"Document_B.txt", "Document_C.txt"]


#Raw documents to tf-idf matrix (or normal count could be done too)


vectorizer = TfidfVectorizer(stop_words='english',
                             use_idf=True,
                             smooth_idf=True)


#SVD for dimensionality reduction


svd_model = TruncatedSVD(n_components=100,         // num dimensions
                         algorithm='randomized',
                         n_iter=10)
```

```
#Pipe tf-idf and SVD, apply on our input documents

svd_transformer = Pipeline([('tfidf', vectorizer),
                            ('svd', svd_model)])svd_matrix =
svd_transformer.fit_transform(documents)
```

SVD matrix can later be used for comparing documents, comparing words, or even to compare queries on a document.

Applications:

1. LSA could be leveraged to extract text summaries from text documents or even product descriptions (like the example above). This could be summarizing product descriptions, unstructured medical reports or even resume summarization.
2. Topic models are built around the idea that the semantics of our document are actually being governed by some hidden, or topics that shape the meaning of our document and corpus. LSA along with SVD can help with topic modelling on a text corpus.
3. LSA and SVD are used as a precursor to find similarities between different words, different documents or comparison on queries on

documents which are done by applying cosine similarity. This could be leveraged in SEO and recommendation systems.

Important links for reference:

1. ML Book: ML Solutions
2. TF-IDF: TF-IDF feature modelling
3. Cosine Similarity: Cosine Similarity Matrix

. . .

Thanks for reading! 😊 If you enjoyed it, test how many times can you hit 👏 in 5 seconds. It's great cardio for your fingers AND will help other people see the story.