Lecture 2: Text Representation

SpaCy

spaCy is a free, open-source Python library that provides advanced capabilities to conduct natural language processing (NLP) on large volumes of text at high speed. It helps you build models and production applications that can underpin document analysis, chatbot capabilities, and all other forms of text analysis.

```
$ pip install spaCy
import spaCy
```

Word Contractions

https://github.com/kootenpv/contractions: This package is capable of resolving contractions.

Usage

```
import contractions
contractions.fix("you're happy now")
# "you are happy now"
contractions.fix("yall're happy now", slang=False) # default: true
# "yall are happy"
contractions.fix("yall're happy now")
# "you all are happy now"
```

Adding custom

```
import contractions
contractions.add('mychange', 'my change')
```

Methods of Text Representation

- One-Hot
- Sparse
- Bag of Words
- TF-IDF

One-Hot Encoding

In one hot encoding, every word (even symbols) that are part of the given text data is written in the form of vectors, constituting only 1 and 0. So one hot vector is a vector whose elements are only 1 and 0. Each word is written or encoded as one hot vector, with each one hot vector being unique.

	Restaurant Reviews
R1	Great restaurant and great service !
R2	They can do better to provide better service
R3	Only two thumbs up, worst service ever

Entire Corpus

Applying One-Hot:

Set of all the words in the corpus		R1: Great Restaurant and great service !	R2: They can do better to provide better service	R3: Only two thumbs up, worst service ever
great	.....	1	0	0
restaurant	.....	1	0	0
and	.....	1	0	0
service	.....	1	1	0
they	.....	0	1	0
can	.....	0	1	0
do	.....	0	1	0
better	.....	0	1	0
to	.....	0	1	0
provide	.....	0	1	0
only	.....	0	0	1
Two	.....	0	0	1
thumbs	.....	0	0	1
up	.....	0	0	1
worst	.....	0	0	1
ever	.....	0	0	1

Sparse vectors

- Optimizing One Hot Encoding using Sparse vectors.
- Only storing indices of each word, to save space.

Bag of Words

Step 1: Determine the Vocabulary

Document:

- the cat sat
- the cat sat in the hat
- the cat with the hat

We first define our vocabulary, which is the set of all words found in our document set. The The only words that are found in the 3 documents are: the , cat , sat , in , the , hat , and with.

Step 2: Count

To vectorize our documents, all we have to do is count how many times each word appears:

Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

TF - IDF

Tf-Idf stands for Term frequency-Inverse document frequency. It tends to capture :

- How frequently a word/term Wi appears in a document dj . This expression can be mathematically represented by Tf(Wi, dj)
- How frequently the same word/term appears across the entire corpus D. This expression can be mathematically represented by df(Wi, D).
- Idf measures how infrequently the word Wi occurs in the corpus D.

Advantages

Captures both the relevance and frequency of a word in a document.

Drawback

Each word is still captured in a standalone manner, thus the context in which it occurs is not captured.

Method of Comparing words

Cosine Similarity

The cosine similarity between words A and B is expressed as follows:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

T-SNE

t-SNE (t-distributed Stochastic Neighbor Embedding) is a technique aimed at reducing high-dimensional embeddings into a lower dimensional space. In practice, it is commonly used to visualize word vectors in the 2D space.