

TERM FREQUENCY (TF AND TF-IDF EMBEDDING)

NATURAL LANGUAGE PROCESSING

- KEERTHANA GOKA
- KUNAL MALHAN

Machine learning and Deep Learning need data to be in numerical form for analysis. Known as Embeddings.




Many crucial tasks can be performed on such embedding like:

Classification

Search engine ranking

Sentiment Analysis etc.



Commonly used embedding techniques:

One hot
encoding

Bag of Words

Category based
embedding

N-gram

TF / IDF



In this presentation, we will explore the concepts behind TF and TF-IDF embeddings, their importance in NLP, and how these methods form the foundation for various text-based applications.

Introduction

Importance in NLP

Importance of TF Embedding

- In its standard for TF embedding is equivalent to non-binary BoW.
- Usually Normalized TF (NTF) is used for embedding.

Importance of Normalized Term Frequency (NTF)

- Relative relevance of word as compared to other words in document, as appose to BoW (which in not relative in nature).

Importance of TF-IDF Embedding

- TF-IDF is designed to capture word importance by balancing two factors:
 - How frequently a word appears in a document (TF)
 - How rare it is across a collection of documents (IDF).

Objectives (What to expect by end of presentation?)

Text to Vectors

- Learn how text data is converted into numerical form, making it suitable for machine learning (ML) and deep learning (DL) models.
- Grasp the Concept of Term Frequency (TF) and TF-IDF
 - Understand how TF and TF-IDF work to highlight word importance and relevance within documents and across a corpus.
- Learn Practical Applications of TF and TF-IDF:
 - Explore how TF and TF-IDF are used in real-world applications such as search engines, recommendation systems, and NLP tasks.

Python code for TF / IDF

Limitations and Challenges on TF / IDF

Problem Definition

- In Bag of Words, relative importance of a word to the other words in the document/corpus not considered.
- Example: Doc 1, Doc 2 : Research papers on fruits:

	Total words	TF	NTF
		Apple	Apple
Doc 1	1000	100	100/1000
Doc 2	200	50	50/200

Pre-work (before embedding)

Tokenization

Case Consistency (lowercase)

Stemming / Lemmatization

Stop-word removal

One-Hot-Encoding (OHE), Bag-of-Word (BoW), TF-IDF, Word2Vec

Normalized -TF

- Normalization helps compute the importance of a word relative to other words in that document.

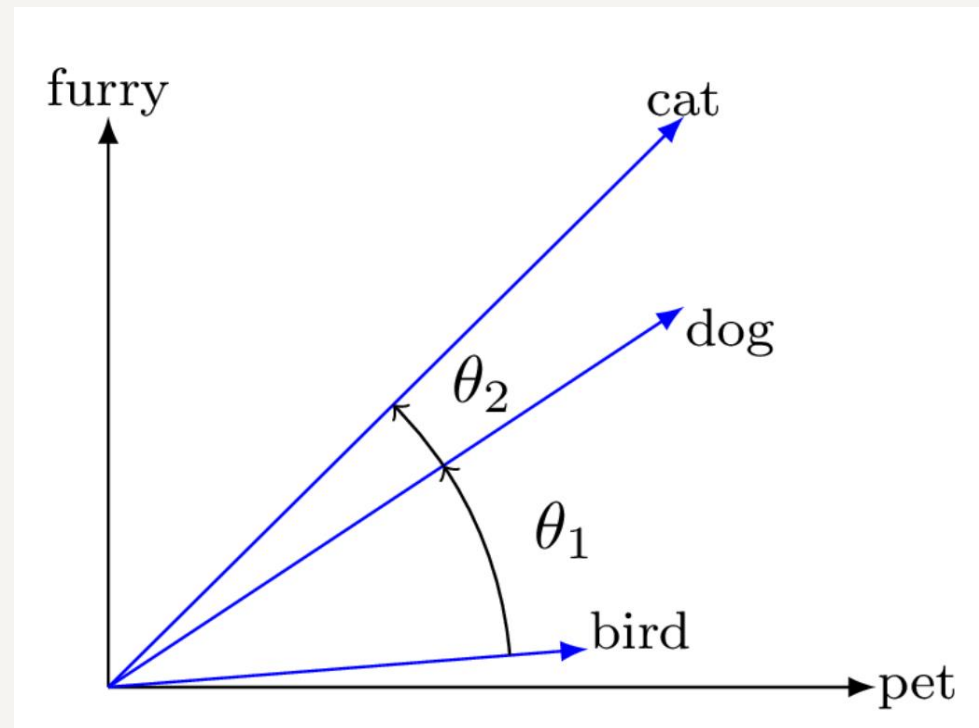
$$\text{TF}_{\text{normalized}} = \frac{\text{Number of occurrences of the term in a document}}{\text{Total number of terms in the document}}$$

- Strengths:
 - More precise and accurate results than previous techniques.
- Limitations:
 - Fails to consider the semantic aspects of words, such as synonyms, antonyms, analogies, etc
 - To cluster the documents together, the same group of words (exact match) needs to occur in them in a similar proportion to increase their similarity score.

Similarity Score

- NTF vectors are projected into the vector space, it is possible to compute similarity scores between query and document, or between documents, by looking at the cosine angle between the respective vectors.
- cosine similarity closer to 1, it implies that the documents are using similar words in proportion.
- The cosine similarity of 0 represents perpendicular vectors, sharing nothing in common.

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \|y\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n (x_i)^2} \sqrt{\sum_{i=1}^n (y_i)^2}}$$



IDF (Inverse Document Frequency)

- NTF does not capture the importance of a word relative to the rest of the documents in the corpus.
- IDF gives importance or weightage to such words which are unique to a certain set of documents, and that can therefore help distinguish and classify documents easily.
- IDF is the ratio of the total number of documents in the corpus to the number of documents the term appears in.

$$IDF_i = \log \frac{N}{df_i}$$

where 'N' is the total number of documents in the corpus, and df_i is the number of documents containing term t_i .

TF-IDF - Introduction

- NTF was used to increase the similarity between vectors if they shared similar words in similar proportions.
- In contrast, IDF reduces the similarity between vectors if they share similar words that are not unique to those documents and might frequently occur in the corpus.

TF-IDF - Definition

- Implicitly embeds and upscales the weights of the words into the document vector that are important in highlighting and classifying that document vector.

$$w_{i,j} = tf_{i,j}idf_i = tf_{i,j} \left(\log \frac{N}{df_i} \right)$$

Where,

$tf_{i,j}$ is the term frequency of term i in document j .

idf_i is the inverse document frequency of term i , and

df_i is the document frequency or the number of documents in which the term appears.

TF-IDF – Strengths and Limitations

- **Strengths:**
 - In terms of accuracy, TF-IDF vectors outperform the previously listed approaches
- **Limitations:**
 - TF-IDF matrices are high-dimensional and sparse.
 - Synonyms with different spellings produce TF-IDF vectors that are not close to each other in the vector space.

Step by Step Explanation: TF and NTF

Example:

Sentence 1: 'Apple is good for health'

Sentence 2: 'Eating an apple daily is good'

Sentence 3: 'Good habits keeps your health good'

TF Embedding Example							
	apple	good	health	eating	daily	habits	keeps
Apple good health	1	1	1	0	0	0	0
eating apple daily good	1	1	0	1	1	0	0
good habits keeps health good	0	2	1	0	0	1	1

Normalized-TF Embedding Example							
	apple	good	health	eating	daily	habits	keeps
Apple good health	1/3	1/3	1/3	0	0	0	0
eating apple daily good	1/4	1/4	0	1/4	1/4	0	0
good habits keeps health good	0	2/5	1/5	0	0	1/5	1/5

Step-by-Step Explanation IDF and TF-IDF

IDF Embedding Example							
	apple	good	health	eating	daily	habits	keeps
IDF	$\log(3/2)$	$\log(3/3)$	$\log(3/2)$	$\log(3/1)$	$\log(3/1)$	$\log(3/1)$	$\log(3/1)$
Values	1/6	0	1/6	1/2	1/2	1/2	1/2
TF-IDF Embedding Example							
	apple	good	health	eating	daily	habits	keeps
Apple good health	0.06	0.00	0.06	0.00	0.00	0.00	0.00
eating apple daily good	0.04	0.00	0.00	0.12	0.12	0.00	0.00
good habits keeps health good	0.00	0.00	0.04	0.00	0.00	0.10	0.10

Tools and Libraries



NLTK library

Base library - Natural Language Tool-Kit

- nltk

NLTK Word Tokenizer

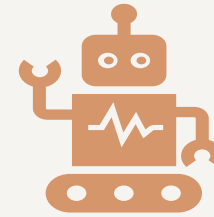
- nltk.word_tokenize

NLTK Stop words list

- nltk.corpus.stopwords

NLTK Snowball Stemmer

- nltk.stem.snowballStemmer



sklearn

TF-IDF Vectorizer Class

- sklearn.feature_extraction.text TfidfVectorizer

- SpaCy
- TensorFlow & PyTorch
- genesis

TfidfVectorizer

- `class sklearn.feature_extraction.text.TfidfVectorizer(*,`
 - `input='content',`
 - `encoding='utf-8',`
 - `decode_error='strict',`
 - `strip_accents=None,`
 - `lowercase=True,`
 - `preprocessor=None,`
 - `tokenizer=None,`
 - `analyzer='word',`
 - `stop_words=None,`
 - `token_pattern='(?u)\\b\\w\\w+\\b',`
 - `ngram_range=(1, 1),`
 - `max_df=1.0,`
 - `min_df=1,`
 - `max_features=None,`
 - `vocabulary=None,`
 - `binary=False,`
 - `dtype=<class'numpy.float64'>,`
 - **`norm='l2',`**
 - **`use_idf=True,`**
 - **`smooth_idf=True,`**
 - `sublinear_tf=False`

Code snippet

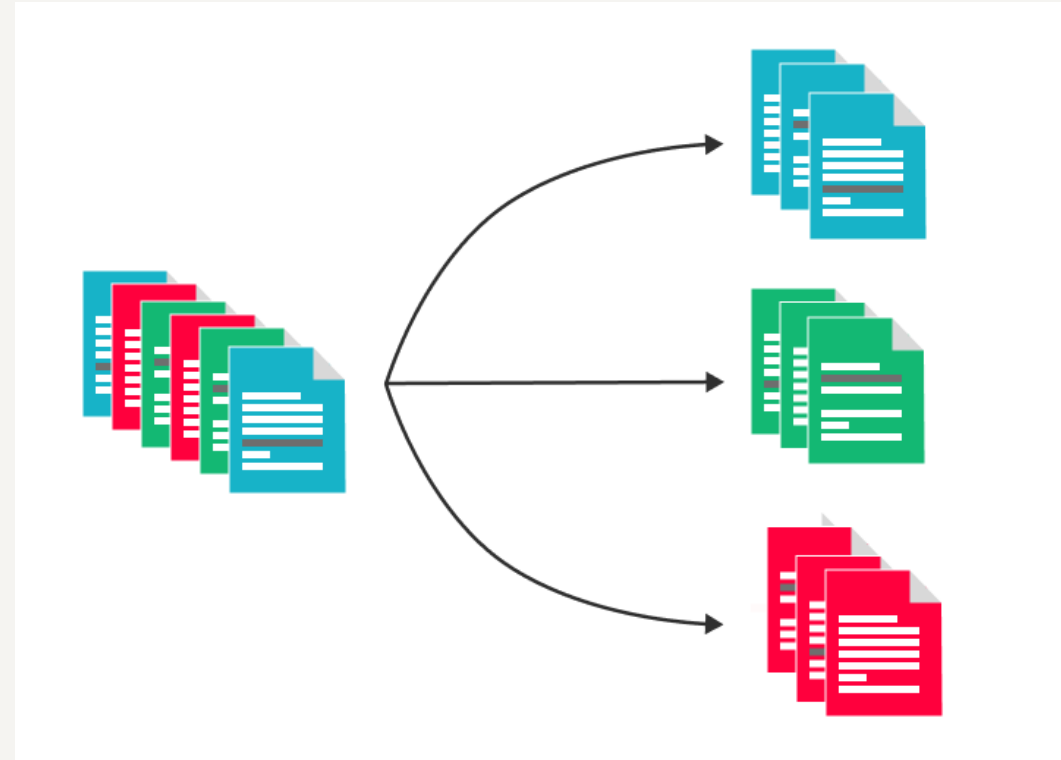
```
tfidf = TfidfVectorizer(norm=None, use_idf=True, smooth_idf=False)
embedding = tfidf.fit_transform(messages).toarray()
```

Code Walkthrough Examples

- Example 1 - Sample (Small) Data
 - Apple is good for health
 - Eating an apple daily is good
 - Good habits keeps your health good
- Example 2 – Ham Spam dataset

Examples

- Application of TF / IDF
 - Text classification
 - Information Retrieval



Challenges and Limitations

Sparsity

- Sparsity is a challenge with machine learning and deep learning algorithms, as it indicates that weights of model are not trained with enough examples.
- Dimensionality reduction techniques can be applied.

Sometimes two lemmatized TF-IDF vectors close to each other are not similar.

Synonyms maynot be close to each others.

Out of Vocabulary

- How to handle the new data point (value which is not in current vocabulary list)?

Conclusion

- TF-IDF has multiple advantages over OHE and BoW:
 - Capturing of word importance
 - **Capture two important information:**
 - Isolated usage of a term (TF)
 - Term's usage across a set of documents (IDF)
 - Differentiate between common and rare terms
 - Fixed size matrix (as of Vocabulary)
- TF / IDF is not the most efficient embedding technique. There exist embedding techniques that uses machine learning and deep learning technique for embedding and works very well.
 - Word2Vec

Thank you