NIT Rourkela

Lab-2, CS6379

NLP Lab, Spring'25

Topics: Word embeddings in NLP

Input: Take the input text of your choice of minimum 500 words.

Q1. Frequency-based embedding: Count Vectors

Extract the corpus C {d1, d2 ... dD} of the document D and the N unique tokens (words) from the corpus C. N unique form our dictionary and the size of the count vector matrix M by DX N. D (i) is the number of times each row of the matrix contains M tokens in the document.

Let us understand this with a simple example.¶

- **D1: He is a lazy boy. She is also lazy.**
- **D2: Neeraj is a lazy person.**

The dictionary created can be a word with a **unique tag in the corpus**: ***['He', 'She', 'lazy', 'boy', 'Neeraj', 'person']***

- Here, **D = 2, N = 6**, The count matrix M of size 2 X 6 will be represented as –

	He	She	lazy	boy	Neeraj	person
D1	1	1	2	1	0	0
D2	0	0	1	0	1	1

Q2. Frequency-based embedding: TF-IDF

1. **TF Score (Term Frequency)** : Considers documents as bag of words, agnostic to order of words. A document with 10 occurrences of the term is more relevant than a document with term frequency 1. But it is not 10 times more relevant, relevance is not proportional to frequency
2. **IDF Score (Inverse Document Frequency):** We also want to use the frequency of the term in the collection for weighting and ranking. Rare terms are more informative than

frequent terms. We want low positive weights for frequent terms and high weights for rare terms.

Mathematical Example¶

***Term Frequency*¶**

TF(***Term Frequency***) = (**Number of times term t appears in a document)/(Number of terms in the document)**

- **TF(This,Document1)** = 1/8
- **TF(This, Document2)**=1/5

It **denotes the contribution of the word to the document i.e words relevant to the document should be frequent.** eg: **A document about Messi should contain the word 'Messi' in large number.**

***Inverse Document Frequency*¶**

IDF(***Inverse Document Frequency***) = **log(N/n)**, where, **N is the number of documents and n is the number of documents a term t has appeared in.**

- where **N is the number of documents** and **n is the number of documents a term t has appeared in.**
- **IDF(This) = log(2/2) = 0.**
- So, how do we explain the reasoning behind IDF? Ideally, if a word has appeared in all the document, then probably that word is not relevant to a particular document. But if it has appeared in a subset of documents then probably the word is of some relevance to the documents it is present in.

Let us compute IDF for the word 'Messi'.

- **IDF(Messi)** = log(2/1) = 0.301.

Now, let us compare the TF-IDF for a common word 'This' and a word 'Messi' which seems to be of relevance to Document 1.

- TF-IDF(This,Document1) = (1/8) * (0) = 0
- TF-IDF(This, Document2) = (1/5) * (0) = 0
- TF-IDF(Messi, Document1) = (4/8)*0.301 = 0.15

Document 1

Term	Count
This	1
is	1
about	2
Messi	4

Document 2

Term	Count
This	1
is	2
about	1
Tf-idf	1

Tutorial: https://www.kaggle.com/code/ashishpatel26/word-embedding-with-beginner-to-advance/notebook