In [1]:
```
#!pip install fasttext
# https://fasttext.cc/
```

Applied ML Series by Sanjana Sahayaraj

In [2]:
```
import fasttext as ft
from sklearn.metrics.pairwise import cosine_similarity
```

In [3]:
```
from string import punctuation
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
stop_words = set(stopwords.words('english'))
```

In [4]:
```
corpus = [
    "I would like to learn machine learning",
    "Natural Language Processing is a sub field in machine learning",
    "NLP stands for Natural Language Processing",
    "Text embedding is an important step in NLP",
    "Text embedding produces numerical representation of texts",
    "TFIDF can produce dense numerical vector form of text"
]
boilerplate = []
```

In [5]:
```
cleanCorpus = []
for sentence in corpus:
    text_tokens = word_tokenize(sentence)
    cleanCorpus.append([w.lower() for w in text_tokens if (not w in stop_words) and (not w in punctuation) and
cleanCorpus
```

Out[5]:
```
[['i', 'would', 'like', 'learn', 'machine', 'learning'],
 ['natural', 'language', 'processing', 'sub', 'field', 'machine', 'learning'],
 ['nlp', 'stands', 'natural', 'language', 'processing'],
 ['text', 'embedding', 'important', 'step', 'nlp'],
 ['text', 'embedding', 'produces', 'numerical', 'representation', 'texts'],
 ['tfidf', 'produce', 'dense', 'numerical', 'vector', 'form', 'text']]
```

In [6]:
```
with open('input/datafile.txt','w') as f: #create an input directory
    for line in cleanCorpus:
        f.write(str(line))
```

In [7]:
```
embedding_model = ft.train_unsupervised(input='input/datafile.txt', model='skipgram', thread=4, dim=20, epoch=1
```

```
Read 0M words
Number of words:  27
Number of labels: 0
Progress: 100.0% words/sec/thread:     791 lr:  0.000000 avg.loss:  4.143502 ETA:   0h 0m 0s
```

In [8]:
```
vector1 = embedding_model.get_word_vector("language")
vector1
```

Out[8]:
```
array([ 0.00023307, -0.00705989,  0.00475206, -0.0015216 ,  0.00270557,
       -0.00106408,  0.00166221, -0.00054808,  0.00079976,  0.00313543,
       -0.00018508, -0.00144309,  0.00092846,  0.0011722 , -0.00523766,
       -0.00083455,  0.00219534,  0.00092186, -0.00625458,  0.00259048],
      dtype=float32)
```

In [9]:
```
vector2 = embedding_model.get_word_vector("natural")
vector2
```

Out[9]:
```
array([ 0.00378634,  0.00466334, -0.00014143, -0.00064331, -0.00039815,
       -0.00649031, -0.00257149, -0.00021205, -0.00292604,  0.00750898,
       -0.00535969, -0.00064032, -0.00509109, -0.00318423, -0.002876  ,
        0.00212803,  0.00281598, -0.00461093, -0.00608426,  0.00702247],
      dtype=float32)
```

In [10]:
```
cosine_similarity([vector1], [vector2]).item(0)
```

Out[10]:
```
0.22484886646270752
```

In [11]:
```
vector3 = embedding_model.get_word_vector("text")
cosine_similarity([vector2], [vector3]).item(0)
```

Out[11]:
```
0.09798339009284973
```

In [12]:
```
vector4 = embedding_model.get_word_vector("speech")
vector4
```

Out[12]:
```
array([ 0.00082471, -0.00050087, -0.00199241, -0.002722  , -0.00123138,
        0.00940048,  0.00419787,  0.01166777,  0.00342794, -0.00709778,
        0.00056245,  0.00293621, -0.00419826, -0.00301849,  0.00231609,
       -0.00390078,  0.00366392,  0.00281464, -0.00234629, -0.00212018],
      dtype=float32)
```

In [13]:
```
cosine_similarity([vector1], [vector4]).item(0)
```

Out[13]:
```
-0.12811897695064545
```

In [14]:
```
embedding_model.save_model("model/ft-w2v.bin")
```