Word Embeddings

Text Representation

Text Representation

1. Bag of Words

Text Representation

1. Bag of Words
2. TF-IDF

Text Representation

Text	service	was	bad	meal	good
service was bad meal was good					
meal was bad service was good					

Text Representation

Bag of Words

Text	service	was	bad	meal	good
service was bad meal was good	1	2	1	1	1
meal was bad service was good	1	2	1	1	1

Text Representation

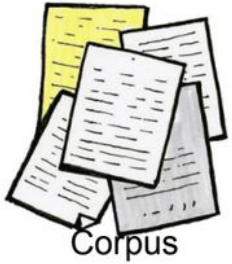
Bag of Words

Text	service	was	bad	meal	good
service was bad meal was good	1	2	1	1	1
meal was bad service was good	1	2	1	1	1

Tf - IDF

Text	service	was	bad	meal	good
service was bad meal was good	0	0	0	0	0
meal was bad service was good	0	0	0	0	0

Word Embeddings



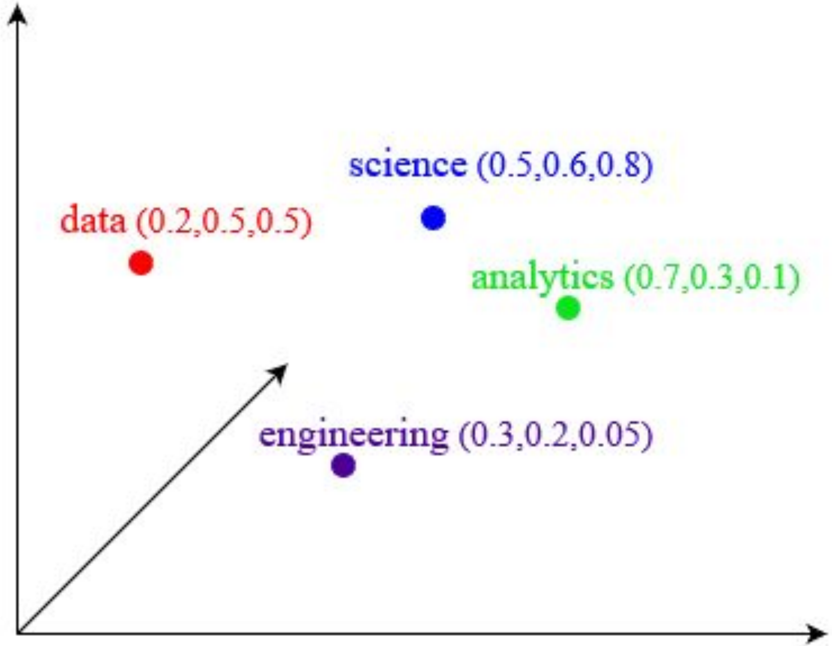
Word Embeddings

Corpus

```
array([[-0.01236233, -0.04655259,  0.00508882, ..., -0.00993368,
         0.01379246,  0.00122126],
       [-0.03087116, -0.02232517,  0.01138248, ..., -0.02389362,
         0.02484551, -0.0087585 ],
       [-0.03504547, -0.04104917,  0.00930308, ..., -0.03002032,
         0.01539359, -0.00338876],
       ...,
       [-0.03802555, -0.017358  ,  0.02445563, ..., -0.0131221 ,
         0.02305542, -0.00747857],
       [-0.02819404, -0.04432267,  0.01159158, ..., -0.02953893,
         0.01612862, -0.0099255 ],
       [-0.0326709 , -0.0484228 ,  0.01606839, ..., -0.03584684,
         0.00761068, -0.00948259]], dtype=float32)
```

Word Vectors

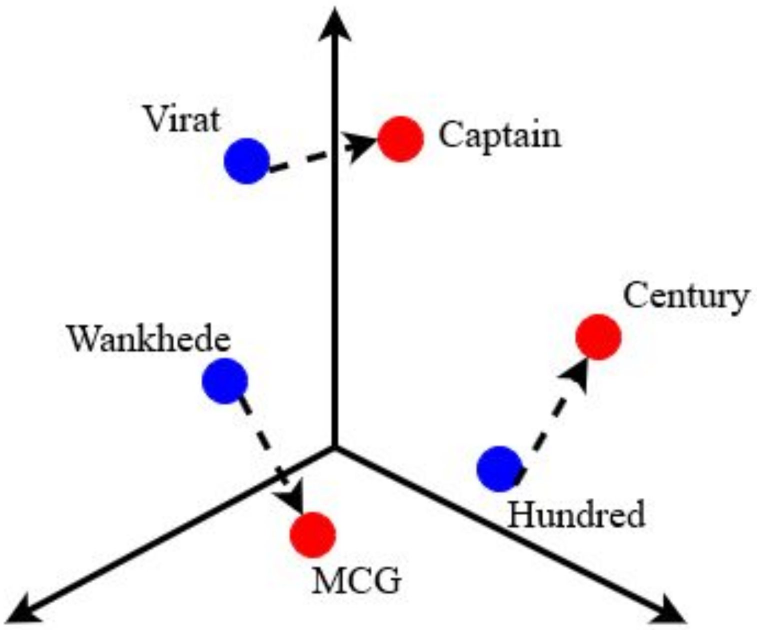
Word Embeddings



Word Embeddings

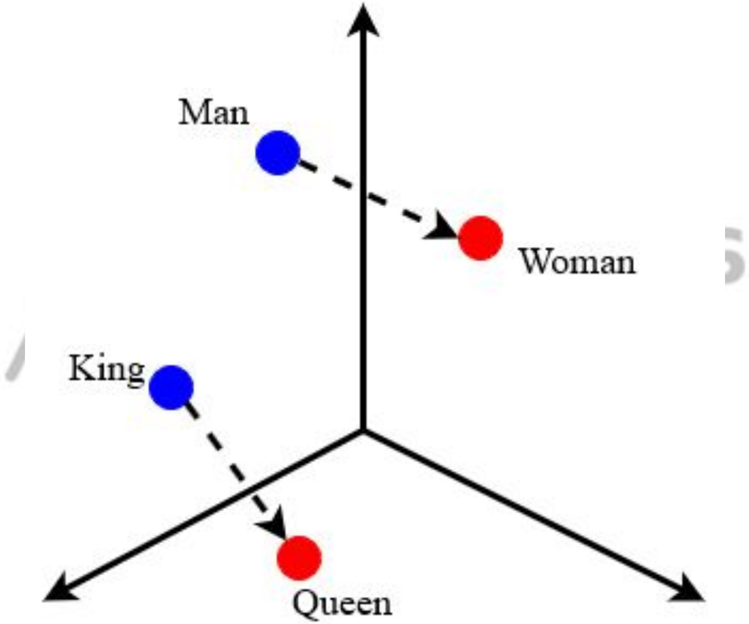
Word Vectors : Context / Meaning + Relationships

Word Embeddings



Analytics Vidhya

Word Embeddings



Word Embeddings

- I love eating fruits
- I love driving cars

Word Embeddings

Term	Vector representation
I	[-0.04813035, -0.08041322, 0.02042717, -0.04620057, 0.00856122, 0.02766979]
love	[-0.01097935, 0.0055207 , -0.02713158, 0.04876678, 0.01179293, 0.02840331]
eating	[-0.00256152, -0.04594067, -0.02137552, 0.05613157, -0.04852077, 0.05093377]
fruits	[-0.03204666, -0.06197819, 0.02622314, -0.01787718, -0.02552203, 0.07250848]
driving	[0.02127126, -0.00173423, -0.04276158, -0.06915958, 0.03542514, -0.03850113]
cars	[0.02993043, -0.03943288, 0.08275513, -0.06427795, 0.07180958, 0.01986287]

Word Embeddings

Text	Vector representation
I love eating fruits	[-0.02342947, -0.04570285, -0.0004642 , 0.01020515, -0.01342216, 0.04487884]
I love driving cars	[-0.02955485, -0.03744626, -0.0033522 , 0.00128311, 0.01017707, 0.02803655]

Word Embeddings

Text	Vector representation
I love eating fruits	[-0.02342947, -0.04570285, -0.0004642 , 0.01020515, -0.01342216, 0.04487884]
I love driving cars	[-0.02955485, -0.03744626, -0.0033522 , 0.00128311, 0.01017707, 0.02803655]

Text	Vector representation
I love eating fruits	[1, 1, 1, 1, 0, 0]
I love driving cars	[1, 1, 0, 0, 1, 1]

Analytics Vidhya

Word Embeddings

Word Embeddings

- Training Word embeddings from scratch

Word Embeddings

- Training Word embeddings from scratch
  - Sparsity of training data

Word Embeddings

- Training Word embeddings from scratch
  - Sparsity of training data
  - Computationally expensive to train

Word Embeddings

- Training Word embeddings from scratch
- Use Pre-trained Word embedding models
  - Word2Vec
  - Glove
  - Fastext

Word2Vec

- Continuous Bag of Words (CBOW) model
- Skip-gram model

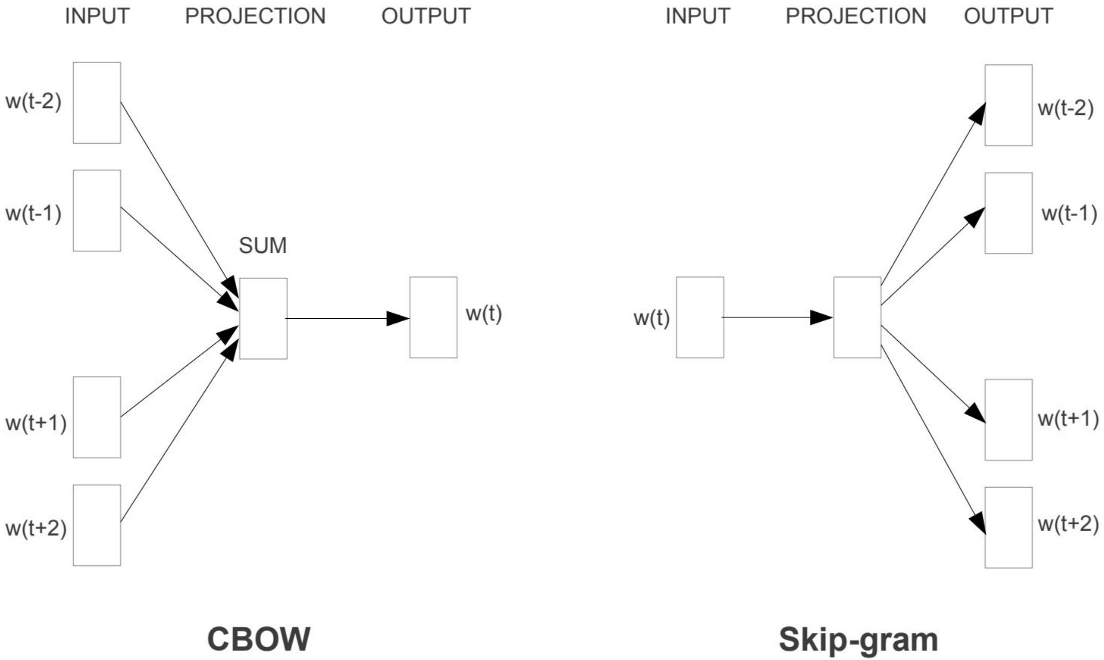
Word2Vec

- Continuous Bag of Words (CBOW) model

Word2Vec

- Skip-gram model

Word2Vec


INPUT
PROJECTION
OUTPUT
w(t-2)
w(t-1)
SUM
w(t)
w(t+1)
w(t+2)
CBOW
INPUT
PROJECTION
OUTPUT
w(t)
w(t-2)
w(t-1)
w(t+1)
w(t+2)
Skip-gram

Word2Vec

<WORD: ???> <Context: ate the food>

Word2Vec

<WORD: DOG> <Context: ???>

Word2Vec

- Context Window

Word2Vec

- Context Window - Number of words appearing to the left and right of a word

Word2Vec

A picture is worth a thousand words

Word2Vec

A picture is worth a thousand words

Continuous Bag of Words:

Input = [“picture”, “is”, “a”, “thousand”], Output = “worth”

Skip-gram:

Input = “worth”, Output = [“picture”, “is”, “a”, “thousand”]

GloVe

GloVe: Global Vectors for Word Representations

- An extensions to word2vec

GloVe

- Word2vec disadvantage of local context

  “Ice cream is in the fridge”

GloVe

- Takes advantage of global statistics and local context (word2vec)

GloVe

- Co-occurrence matrix

GloVe

- I play cricket
- I love cricket
- I love football

GloVe

	play	love	football	I	cricket
play	0.0	0.0	0.0	1.0	1.0
love	0.0	0.0	1.0	2.0	1.0
football	0.0	1.0	0.0	0.0	0.0
I	1.0	2.0	0.0	0.0	0.0
cricket	1.0	1.0	0.0	0.0	0.0

Analytics Vidhya

GloVe

P(cricket/play) = 1

P(cricket/love) = 0.5

GloVe

P(cricket/play) / P(cricket/love) = 2

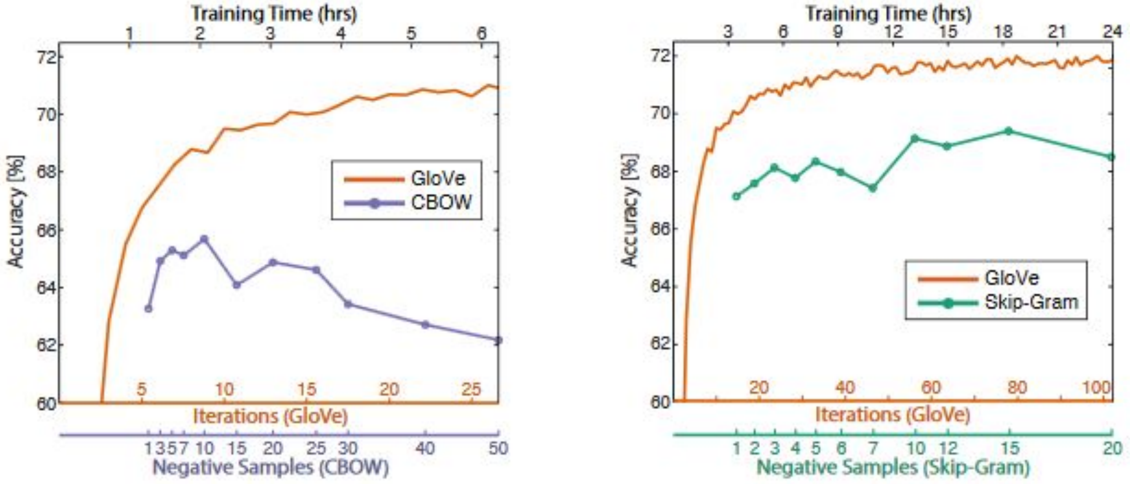
GloVe

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

GloVe

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

GloVe vs Word2Vec



(a) GloVe vs CBOW

(b) GloVe vs Skip-Gram

Figure 4: Overall accuracy on the word analogy task as a function of training time, which is governed by the number of iterations for GloVe and by the number of negative samples for CBOW (a) and skip-gram (b). In all cases, we train 300-dimensional vectors on the same 6B token corpus (Wikipedia 2014 + Gigaword 5) with the same 400,000 word vocabulary, and use a symmetric context window of size 10.

fastText

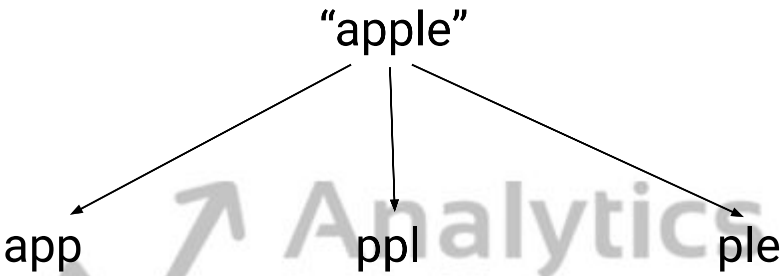
fastText

- Extension of word2vec

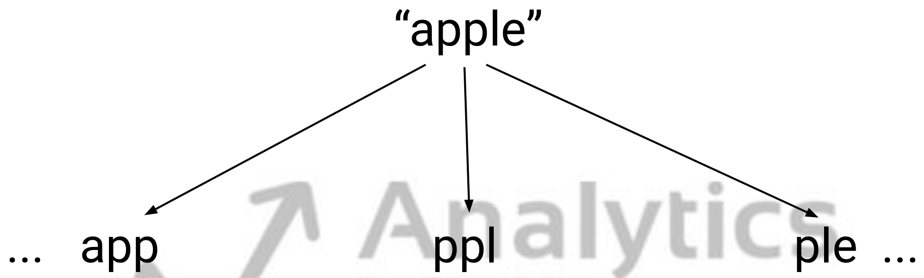
fastText

- Extension of word2vec
- Complete word and the n-gram character representations of the word

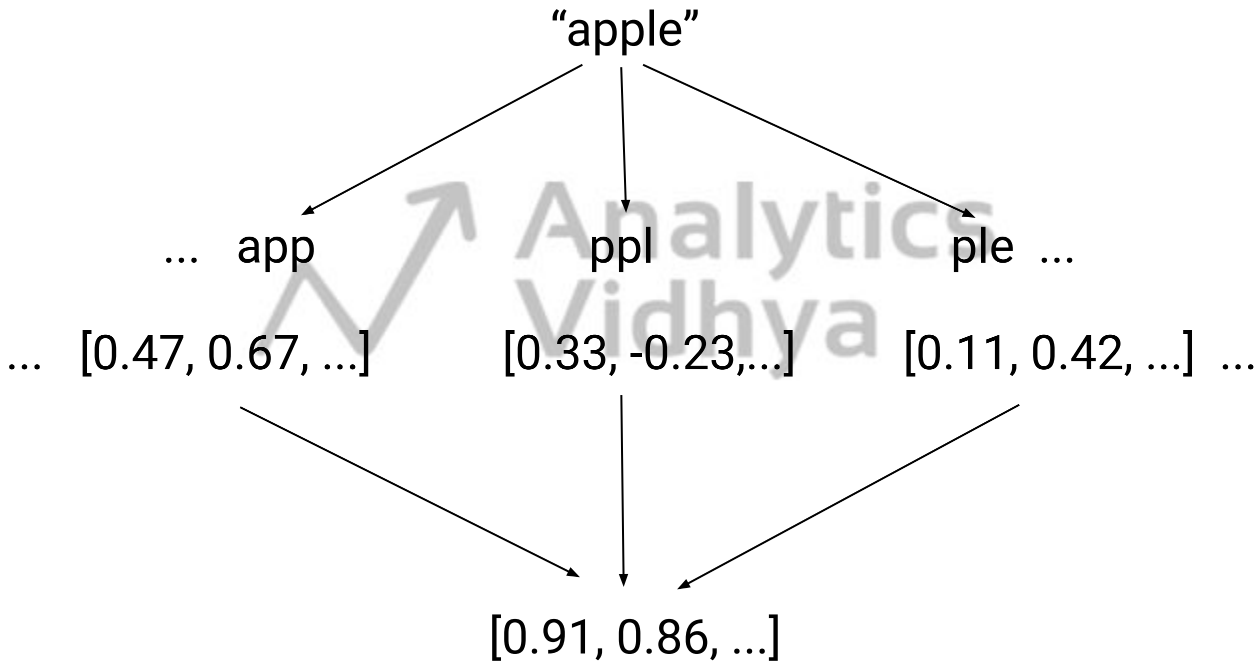
fastText



fastText



fastText



fastText

- Generate better word embeddings for rare words

fastText

- Generate better word embeddings for rare words
- Generate word embeddings for out of vocabulary words

fastText vs Other models

- Longer time to train

fastText vs Other models

- Longer time to train
- Larger memory requirement

fastText vs Other models

- Longer time to train
- Larger memory requirement
- Better performance than word2vec and GloVe

Which embedding model to use?

Which embedding model to use?

- Right answer - It depends!

Which embedding model to use?

- Right answer - It depends!
- Best approach - Experiment!

Thank You