Computing Lab

CS 69201
Department of Computer Science and Engineering
IIT Kharagpur

Locality Sensitive Hashing(LSH)

Similarity Search and Limitations

- Most relevant documents or items need to be retrieved for a query
- Often appears in the NLP domain, search engines or recommender systems
- Can be computationally expensive when dealing with large datasets
- We need a method that can quickly identify similar items without having to exhaustively compare each pair

Locality Sensitive Hashing

- Efficiently approximates similarity search by reducing the dimensionality of data
- Still preserves similarity between the data
- Core idea behind LSH is to hash data points in such a way that similar items are mapped to the same hash buckets with high probability
- In normal hashing techniques we try to reduce collisions, in LSH we wants to maximise the collision

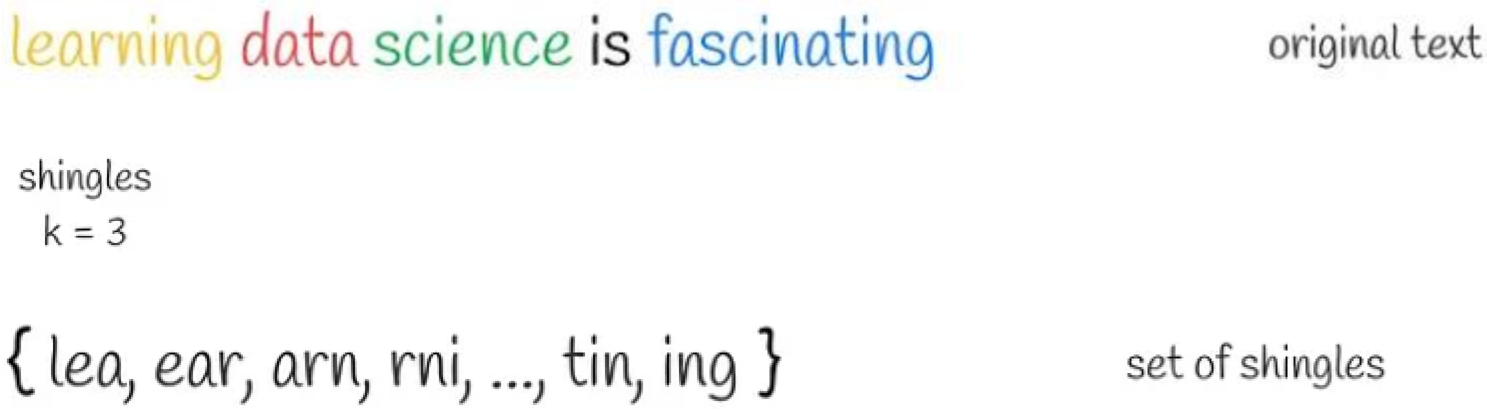
LSH Algorithm

Comprises three main steps

- Shingling
- MinHashing
- Band and Hash

Shingling

- Process of collecting *k*-grams on given texts
- *k*-gram is a group of *k* sequential tokens
- Collect all k-grams to create the shingling set


learning data science is fascinating
original text
shingles
k = 3
{ lea, ear, arn, rni, ..., tin, ing }
set of shingles

Vocabulary

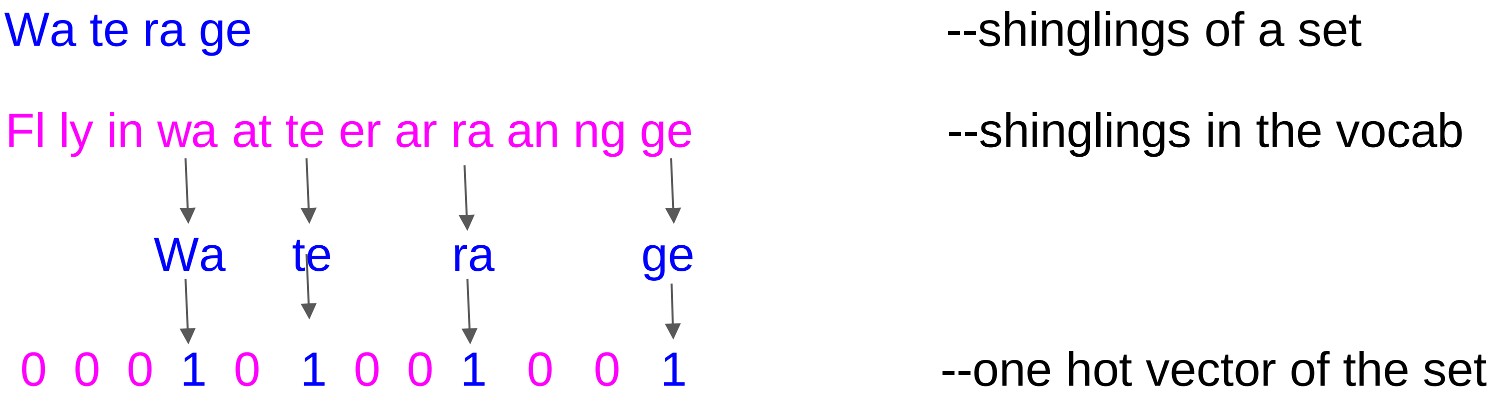
- Vocabulary is the collection of all set of shingles across our dataset
- We can create vocabulary by taking the union of all the set of shingles

Lion roars: {lio,ion,on_,n_r,_ro,roa,oar,ars}
Dog barks: {dog,og_,g_b,_ba,bar,ark,rks}

Vocabulary: {lio,ion,on_,n_r,_ro,roa,oar,ars,dog,og_,g_b,_ba,bar,ark,rks}

One-hot Vector

- Create an empty vector full of zeros and the same length as our vocab
- Then consider the shingles which are appearing in the set of the sentence
- For every shingle that appears, identify the position of that shingle in the vocab and set the respective position in our new zero-vector to 1.



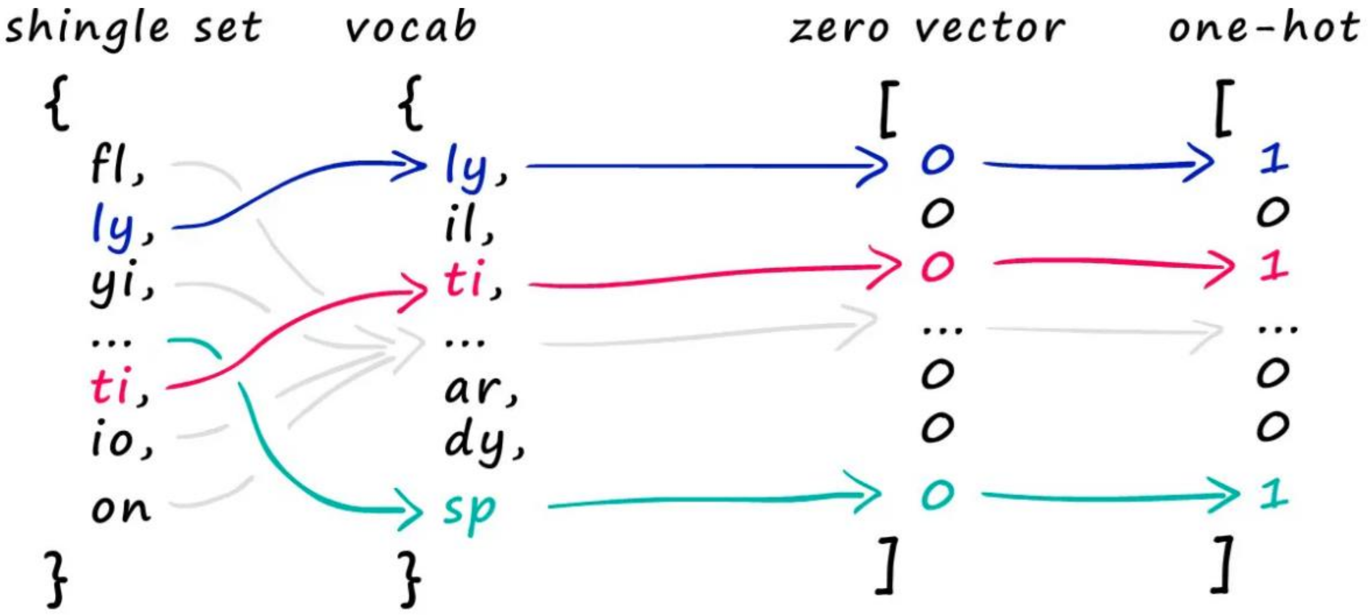
Wa te ra ge --shinglings of a set

Fl ly in wa at te er ar ra an ng ge --shinglings in the vocab

Wa te ra ge

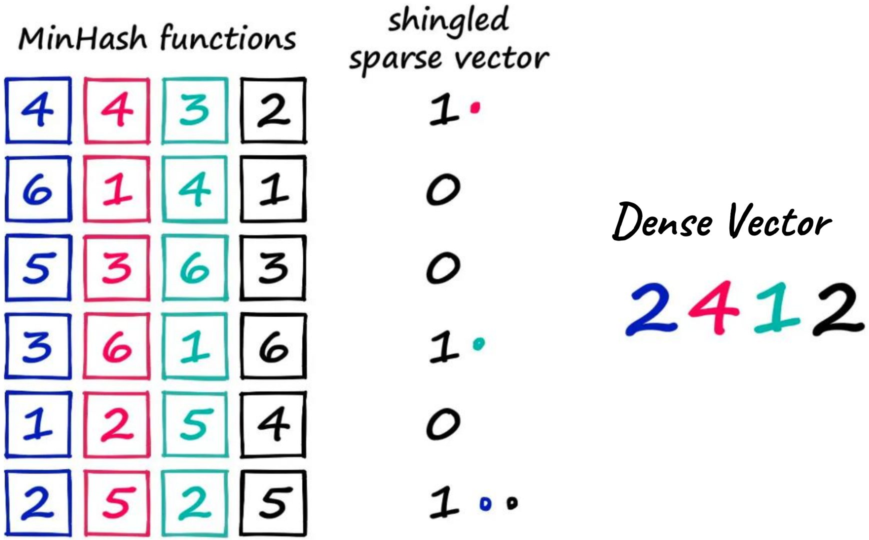
0 0 0 1 0 1 0 0 1 0 0 1 --one hot vector of the set

Shingling


shingle set
vocab
zero vector
one-hot
{
fl,
ly,
yi,
...
ti,
io,
on
}
{
ly,
il,
ti,
...
ar,
dy,
sp
}
[
0
0
0
...
0
0
0
]
[
1
0
1
...
0
0
1
]

Minhashing

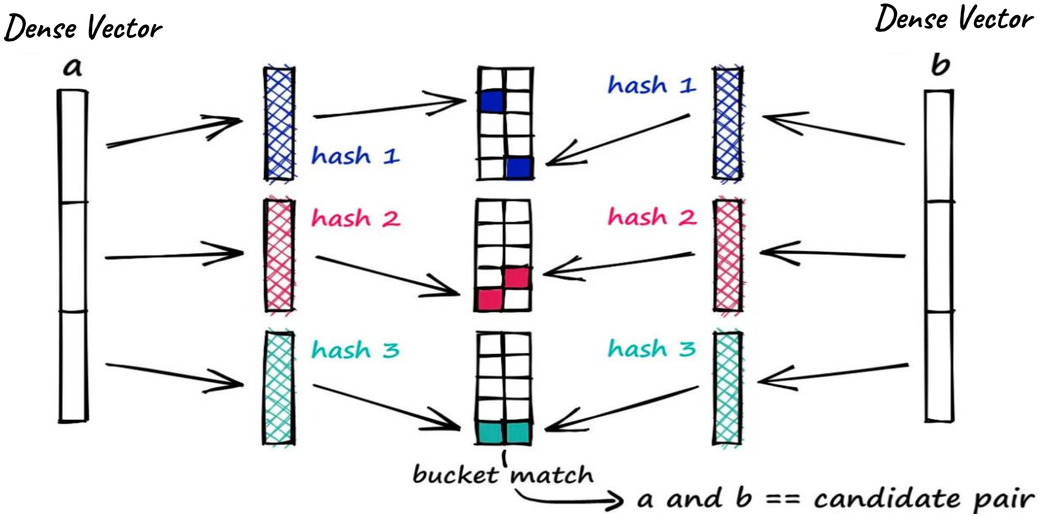
- Converting sparse to dense vector
- MinHash functions are simply a randomized order of numbers from *1* to len(vocab)



MinHash functions				shingled sparse vector
4	4	3	2	1
6	1	4	1	0
5	3	6	3	0
3	6	1	6	1
1	2	5	4	0
2	5	2	5	1

Hash and Band

- Splitting our vectors into sub-parts called *bands* b
- Rather than processing the full vector through our hash function, we pass each band of our vector through a hash function
- Given a collision between any two sub-vectors, we consider the respective full vectors as candidate pairs



Resources

- Locality sensitive hashing - https://www.pinecone.io/learn/series/faiss/locality-sensitive-hashing/

Thank You!