NLP Intuition

Deep Learning and NLP A-Z

NLP Intuition

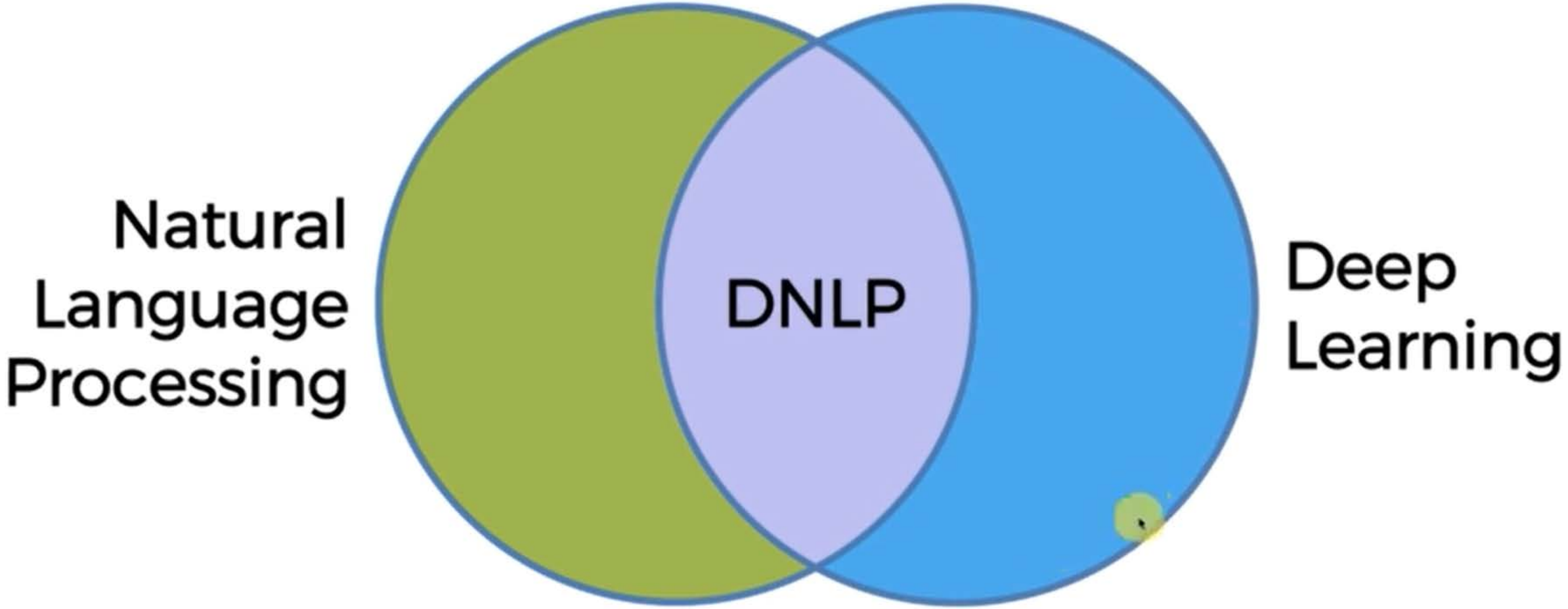
Here's what we will learn:

- Types of Natural Language Processing
- Classical vs Deep Learning Models
- Bag-Of-Words

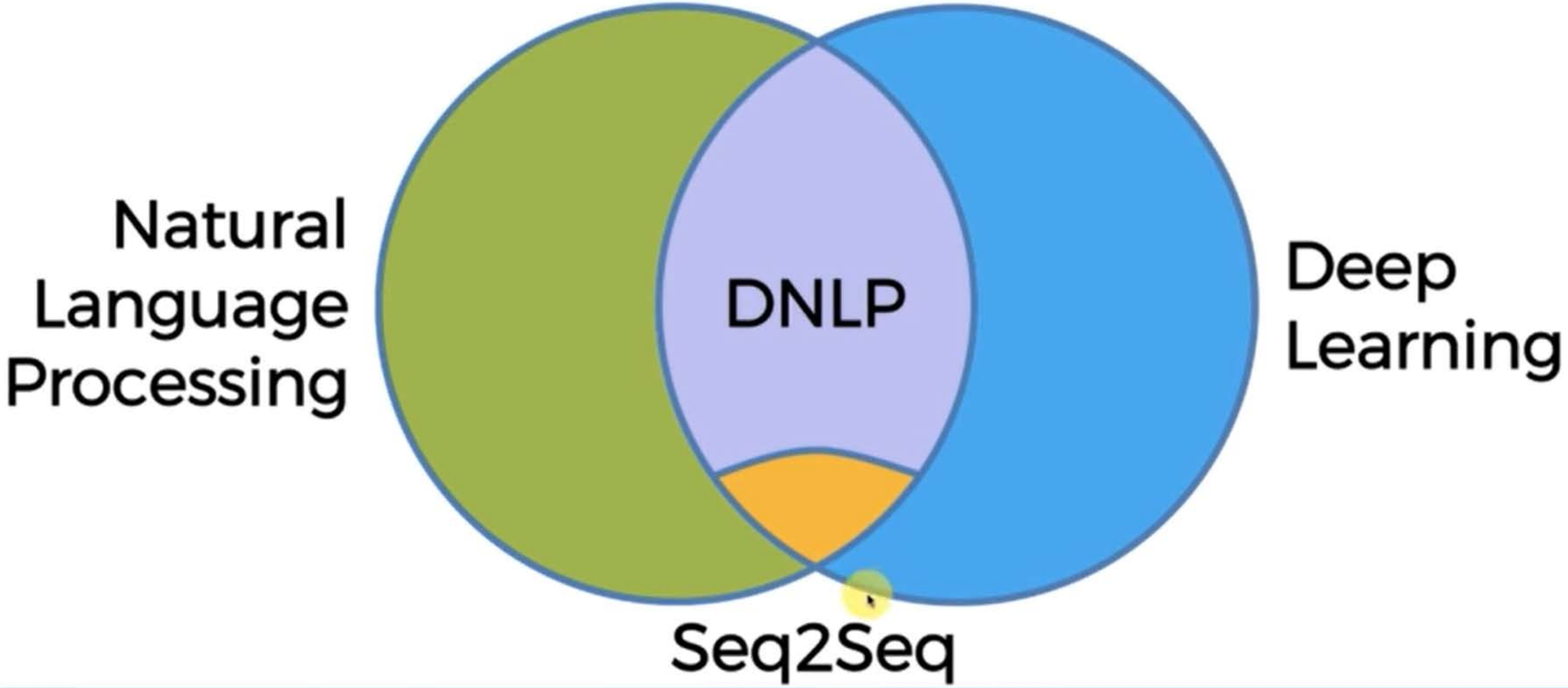
- *Note: we will not talk about Seq2Seq or Chatbots*

Types of NLP

Types of NLP



Types of NLP

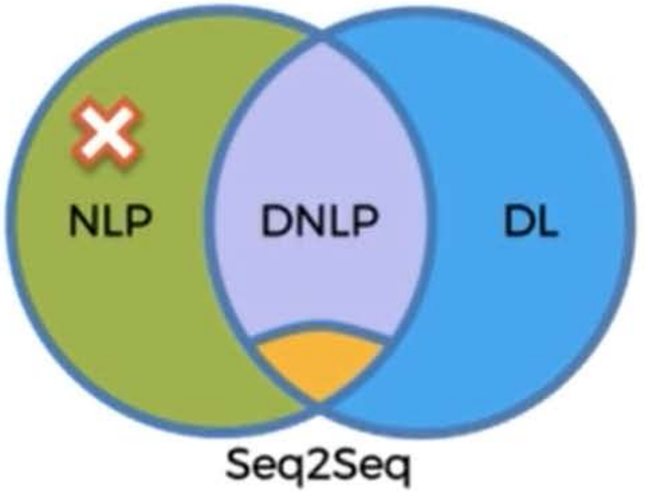


Classical vs Deep Learning Models

Classical vs Deep Learning Models

Some examples:

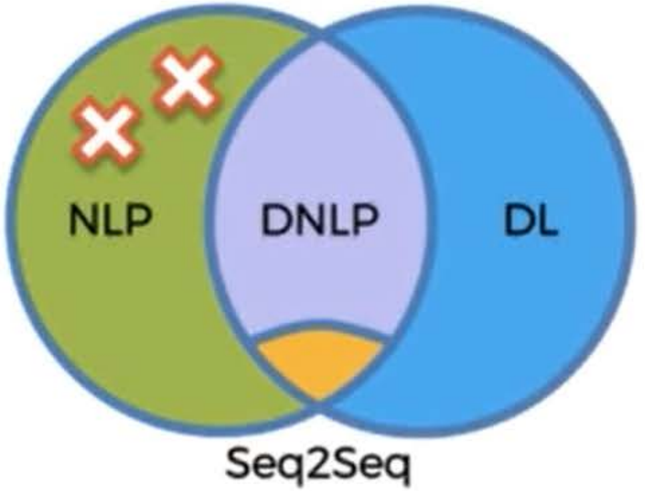
1. If / Else Rules (Chatbot)

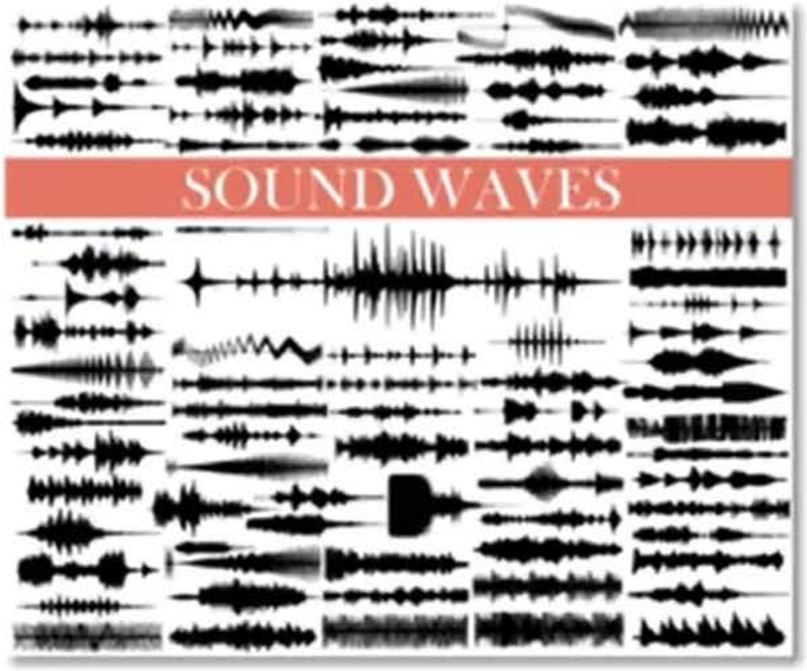


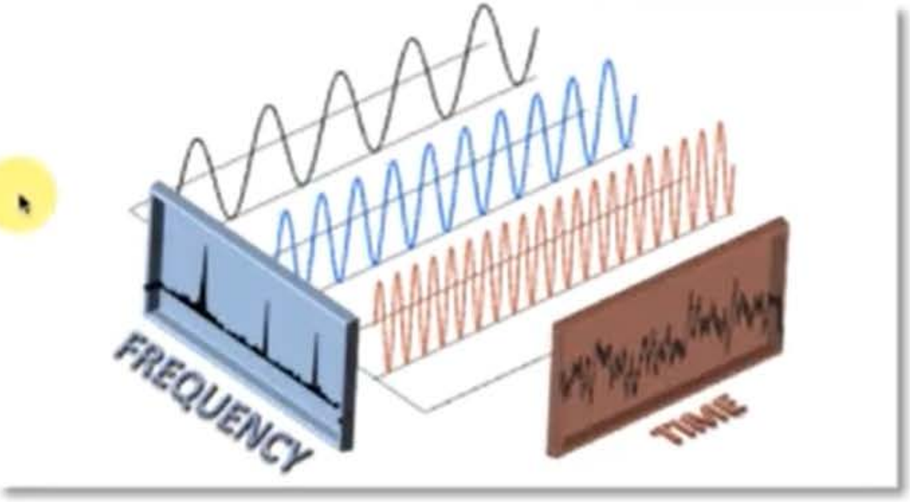
Classical vs Deep Learning Models

Some examples:

1. If / Else Rules (Chatbot)
2. Audio frequency components analysis (Speech Recognition)





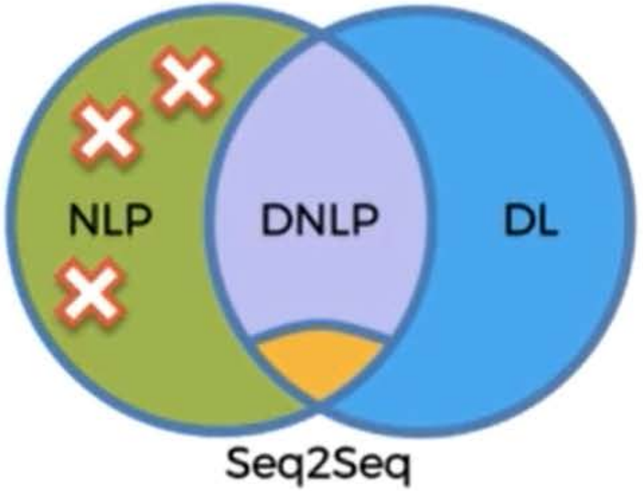


Classical vs Deep Learning Models

Some examples:

1. If / Else Rules (Chatbot)
2. Audio frequency components analysis (Speech Recognition)
3. Bag-of-words model (Classification)

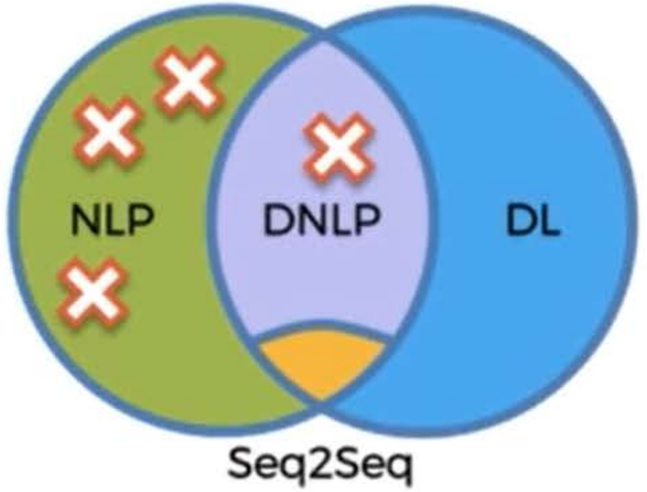
Comment	Pass/Fail
Great job!	1
Amazing work.	1
Well done.	1
Very well written.	1
Poor effort.	0
Could have done better.	0
Try harder next time.	0
...	...



Classical vs Deep Learning Models

Some examples:

1. If / Else Rules (Chatbot)
2. Audio frequency components analysis (Speech Recognition)
3. Bag-of-words model (Classification)
4. CNN for text Recognition (Classification)



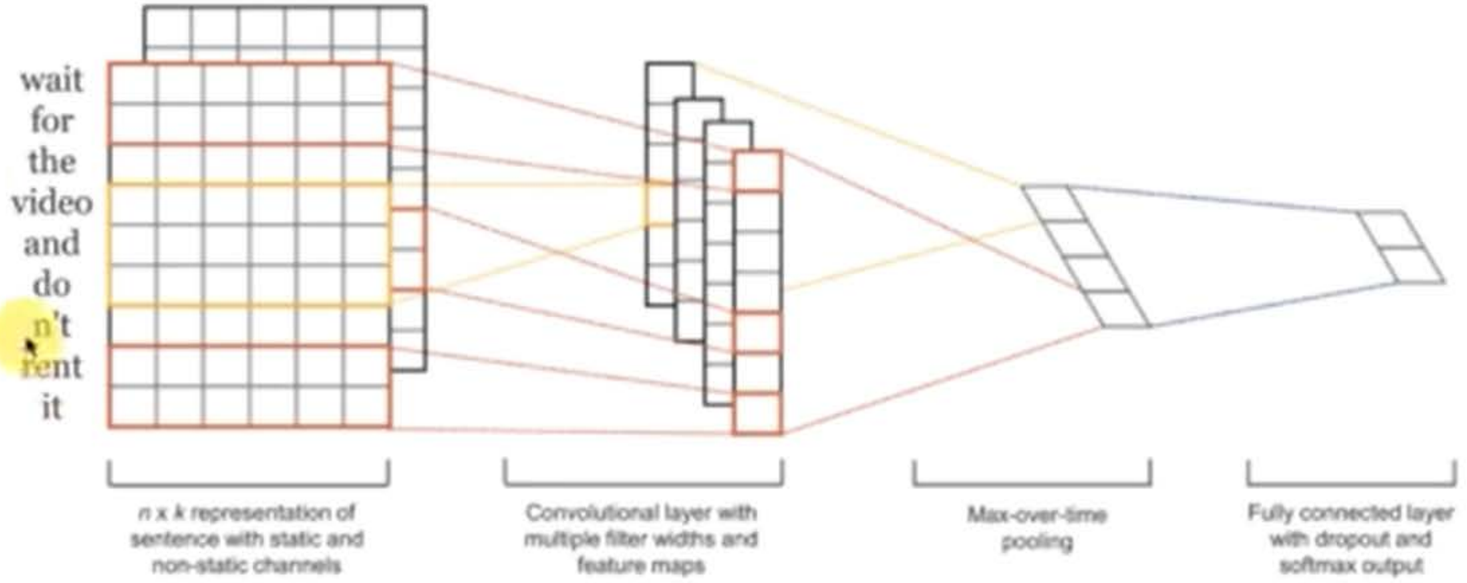
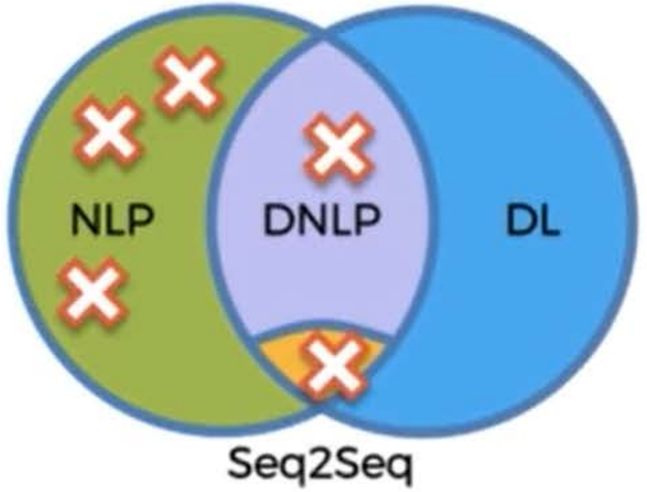


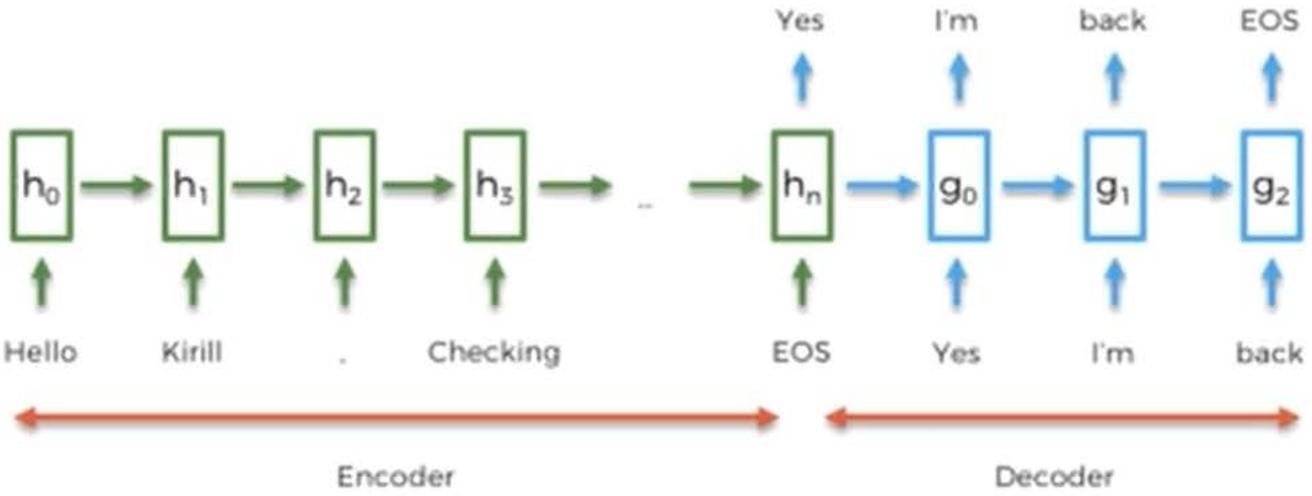
Image Source: www.wildml.com

Classical vs Deep Learning Models

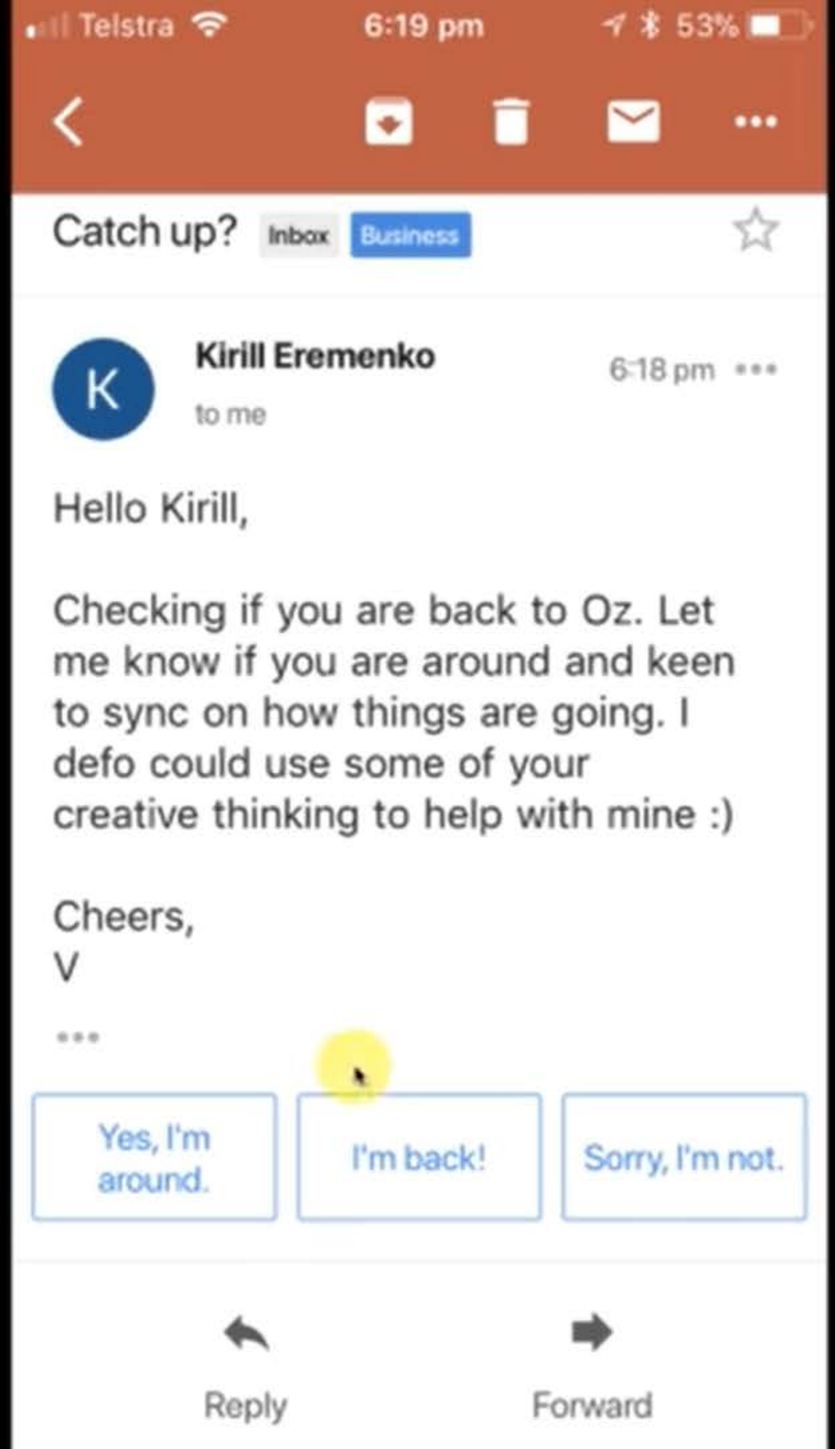
Some examples:

1. If / Else Rules (Chatbot)
2. Audio frequency components analysis (Speech Recognition)
3. Bag-of-words model (Classification)
4. CNN for text Recognition (Classification)
5. Seq2Seq (many applications)





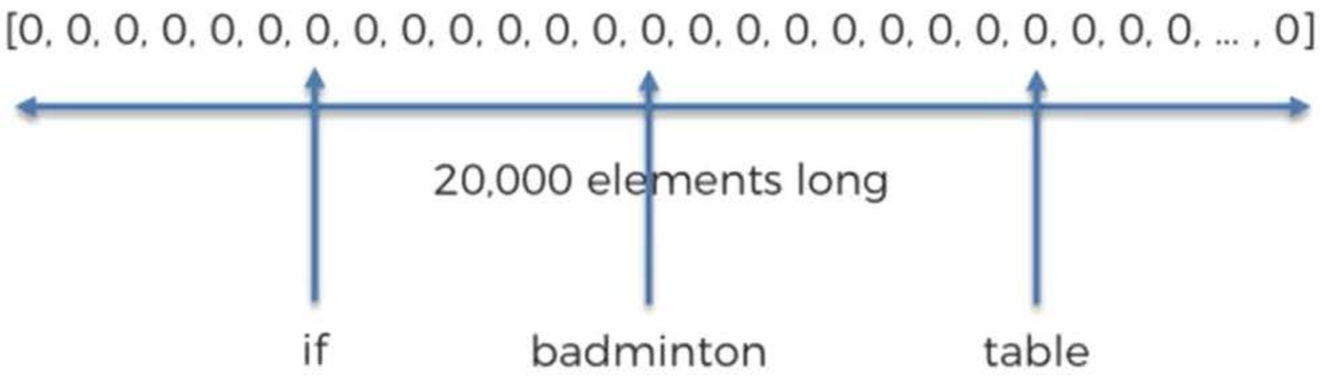
Bag-Of-Words

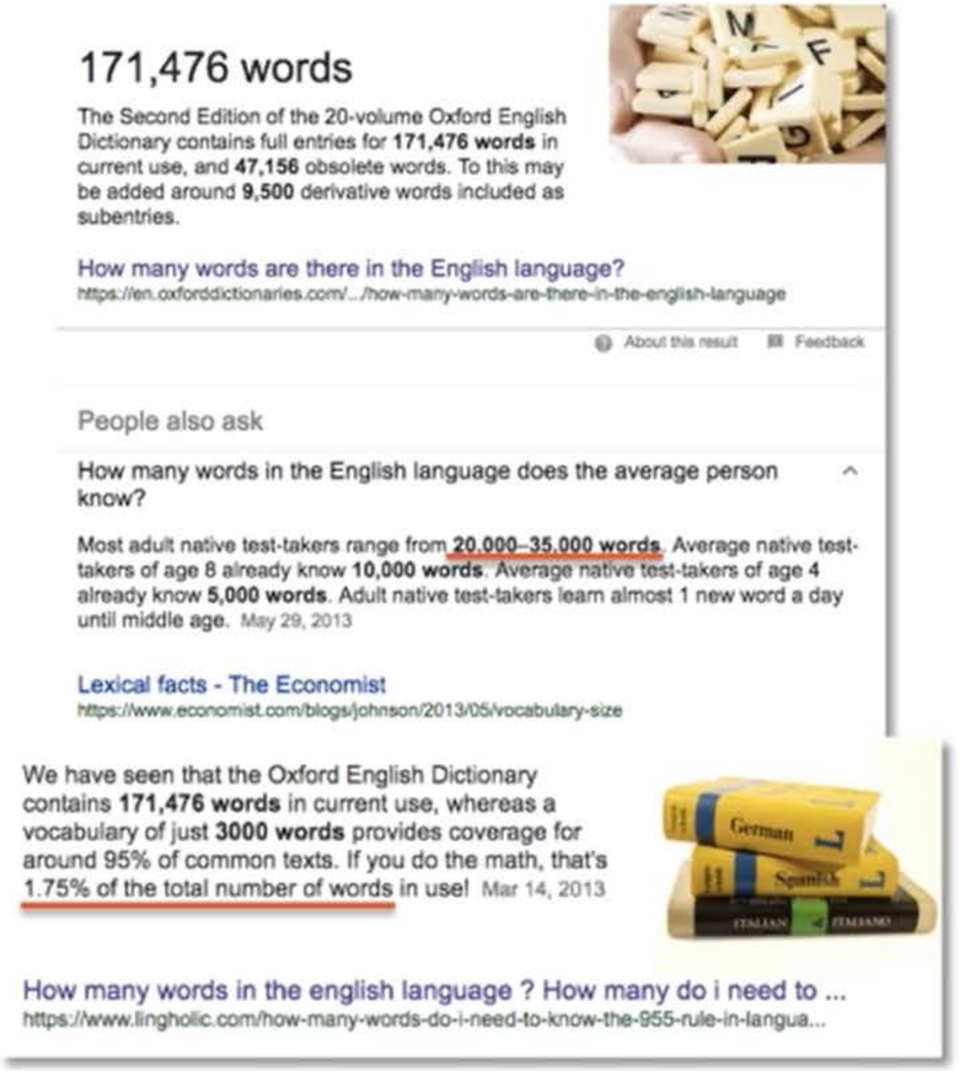

Telstra
6:19 pm
53%
Catch up?
Inbox
Business
K
Kirill Eremenko
6:18 pm
to me
Hello Kirill,
Checking if you are back to Oz. Let me know if you are around and keen to sync on how things are going. I defo could use some of your creative thinking to help with mine :)
Cheers,
V
Yes, I'm around.
I'm back!
Sorry, I'm not.
Reply
Forward

Bag-Of-Words

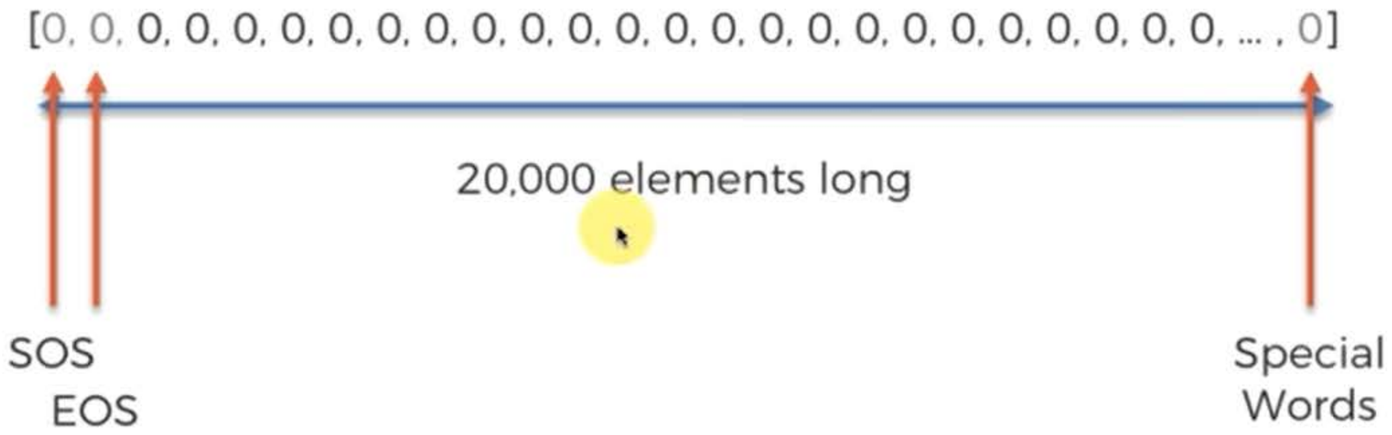
Yes / No

Bag-Of-Words

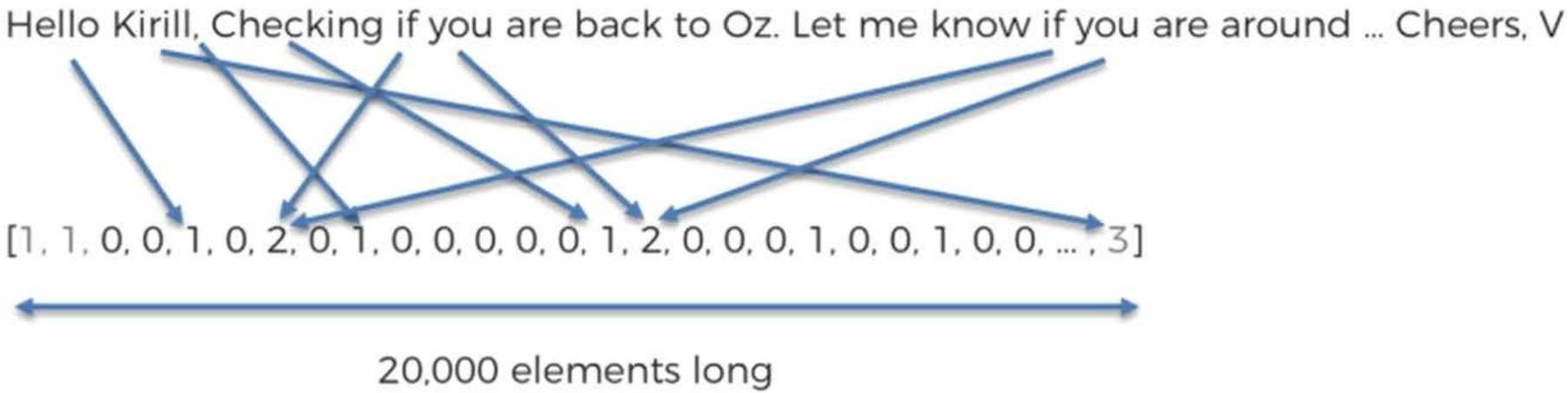




Bag-Of-Words



Bag-Of-Words



Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around … Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, … , 3] → Yes / No ?

20,000 elements long

Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around … Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, … , 3] → Yes / No ?

20,000 elements long

Training Data:

Hey mate, have you read about Hinton's capsule networks?	→	No
Did you like that recipe I sent you last week?	→	Yes
Hi Kirill, are you coming to dinner tonight?	→	Yes
Dear Kirill, would you like to service your car with us again?	→	No
Are you coming to Australia in December?	→	Yes
…	→	…

Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around ... Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, ... , 3] → Yes / No ?

20,000 elements long

Training Data:

[1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, ... , 2] → No

[1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, ... , 0] → Yes

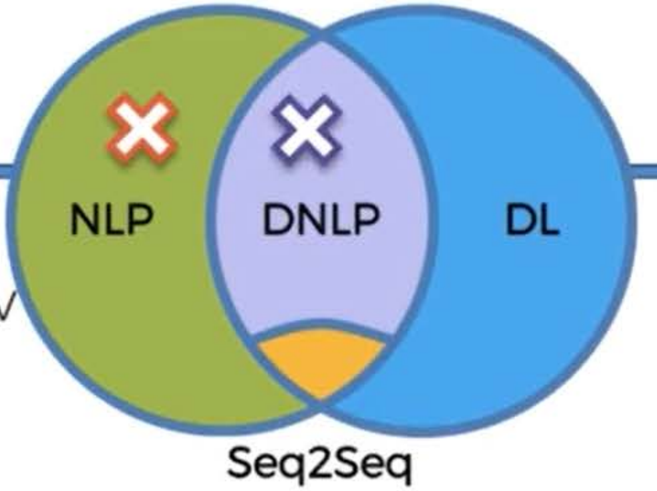
[1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, ... , 1] → Yes

[1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, ... , 1] → No

[1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, ... , 1] → Yes

... → ...

Bag-Of-Words



Hello Kirill, Checking if you are back to Oz. Let me know if you are around ... Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, ... , 3]

20,000 elements long

Yes / No ?

Training Data:

[1, 1, 0, 0, ... 0, 0, 1, 0, 1
[1, 1, 0, 0, ... 0, 0, 2, 0, 0
[1, 1, 0, 0, ... 0, 0, 1, 0, 0
[1, 1, 0, 0, ... 0, 0, 1, 1, 0
[1, 1, 0, 0, ... 0, 0, 1, 0, 0

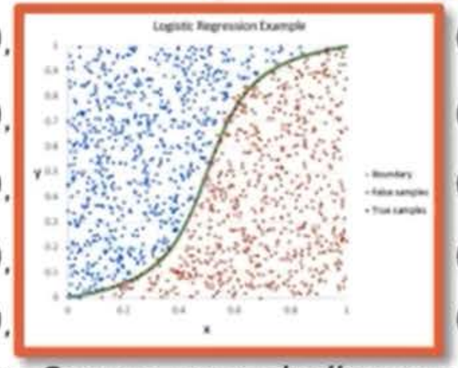
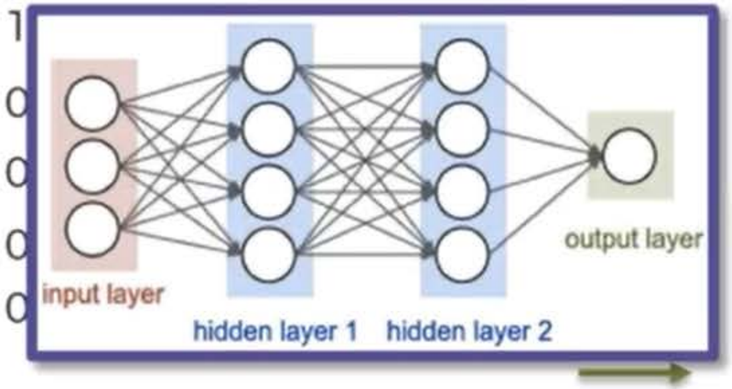


Image Source: www.helloacm.com



No
Yes
Yes
No
Yes
...

Cleaning the texts

In [3]:

```
import re

import nltk
#stopword are the words which are not going to affect the reviews(eg. the,a)
nltk.download('stopwords')
from nltk.corpus import stopwords

#here portstemmer keeps the root of word (eg. liked -> like)
from nltk.stem.porter import PorterStemmer

corpus = []
for i in range(0, 1000):

  #here ^ : not , and [^a-zA-Z]:all punctuations other than a-z,A-Z replaces by " "
  review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
  review = review.lower()
  review = review.split()

  #now we will do splitting
  ps = PorterStemmer()
  all_stopwords = stopwords.words('english')
  all_stopwords.remove('not')
  review = [ps.stem(word) for word in review if not word in set(all_stopwords)]
  review = ' '.join(review)
  corpus.append(review)
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\rajne\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Creating the Bag of Words model

In [0]:
```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, -1].values
```

Splitting the dataset into the Training set and Test set

In [0]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

Training the Naive Bayes model on the Training set

In [0]:
```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

Out[6]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```