NetApp Data Challenge

(Kshitij 2k19)

TEAM
DATA WARRIORS
(TM190904)

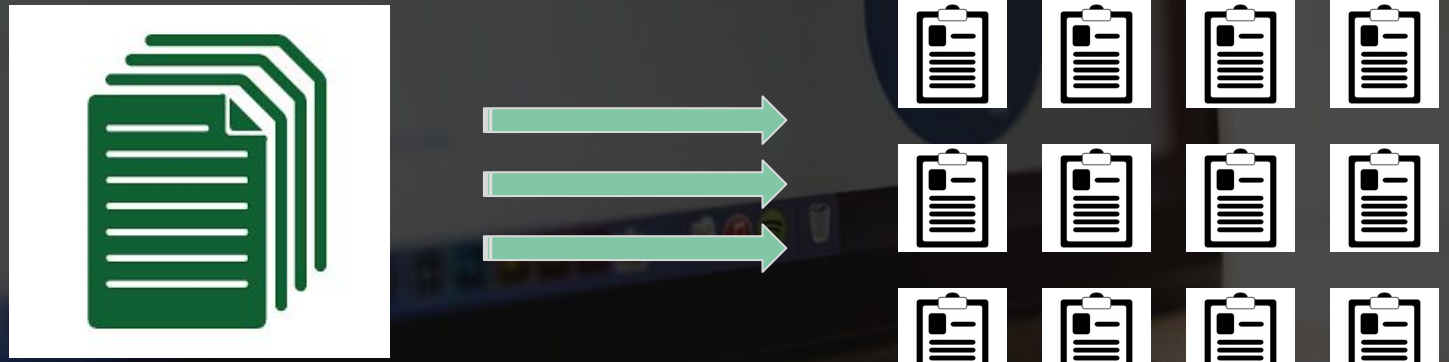
NEWS ARTICLES,
FB POSTS, BLOGS
TWEETS

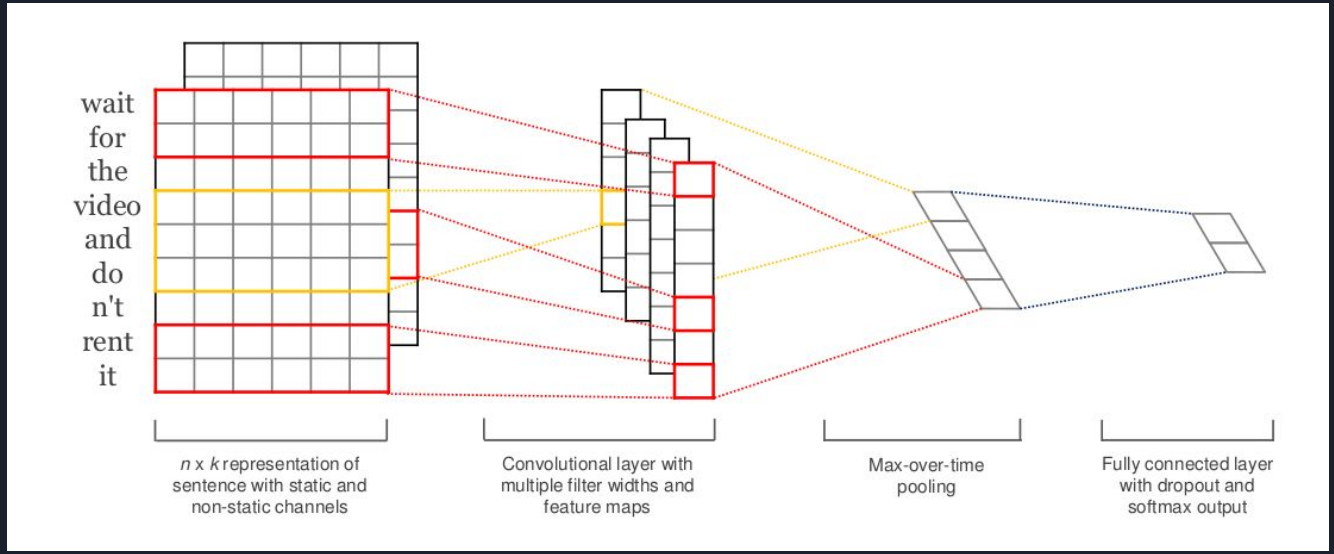
OPINIONS
(NEED TO ANALYZE)

PROBLEM STATEMENT

DOCUMENT CLASSIFICATION

CLASSIFYING DOCUMENTS INTO PREDEFINED **CATEGORIES**





CONVOLUTIONAL NEURAL NETWORKS FOR DOCUMENT CLASSIFICATION

CNN MODEL

DATA REPRESENTATION

- **INPUT :**
  - Dense representation of words needed.
  - Word Vectors trained using Word2Vec.
- **PREPROCESSING STEPS:**
  - Dynamic input.
  - Padding and Truncation of input sequence.
- **FINAL FORM**
  - Each training example : 100 X 100 matrix
  - Dimensions:
    - Sequence Length
    - Embedding Size

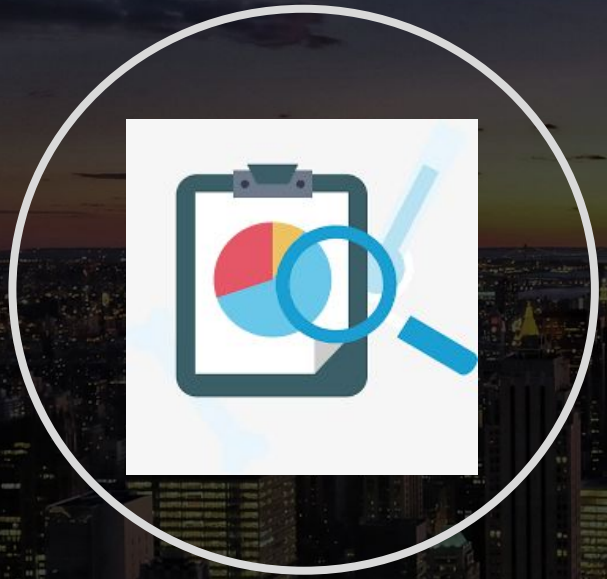
CNN MODEL

NEURAL NETWORK ARCHITECTURE

- **CONVOLUTIONAL LAYER**
  - Multiple Filters of varying window size
  - Feature Map Generation
  - Context Capture and Word Relation
- **MAX-POOLING LAYER**
  - One Feature per Filter
  - Capture the Most Important Feature
  - Variable Length Sentences
- **FULLY CONNECTED LAYER**
- **SOFTMAX LAYER**
  - Probability distribution over labels

INFERENCES

POSITIVES	NEGATIVES
• Simple yet accurate. • Using Pre-trained word vectors improves performance. • Fine-tuning word vectors further improves performance. • Gives good results with a limited number of parameters compared to very deep neural networks.	• Requires word embeddings trained on Word2Vec • Large computational time and resources required. • Hyperparameter tuning expensive due to large single training time.

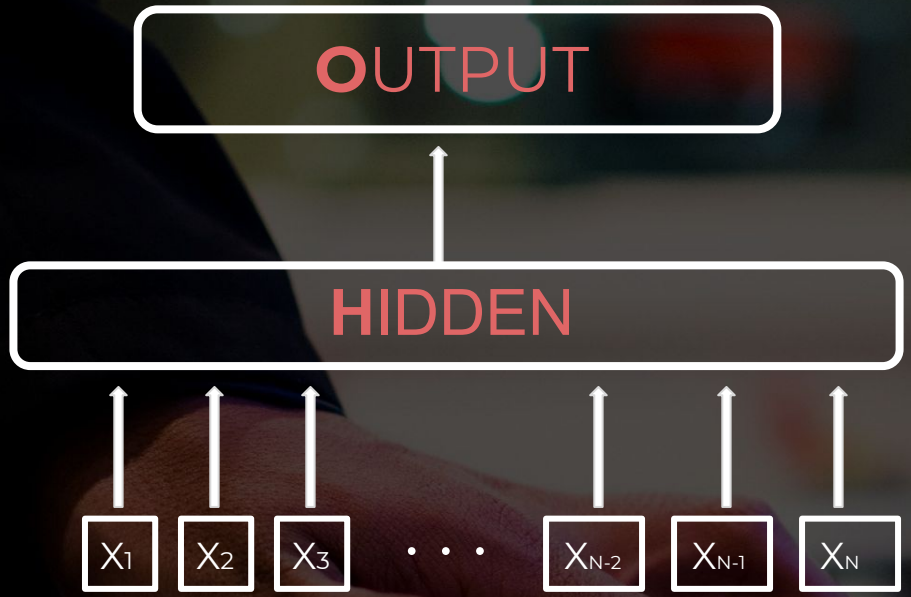


RESULTS

- 75-25 split of a part of the training data (10,000 examples)was done for the creation of the Validation Set.
- Hyperparameters tuned using grid search.

F1 - SCORE

TRAINING SET	**0.7848**
VALIDATION SET	**0.6234**



MODEL ARCHITECTURE
FAST DOCUMENT CLASSIFICATION

FASTTEXT FEATURES

- EXTENDS POWERFUL LINEAR MODELS.
- BAG OF N-GRAMS.
- IMPROVES GENERALIZATION.
- REDUCES TIME COMPLEXITY.

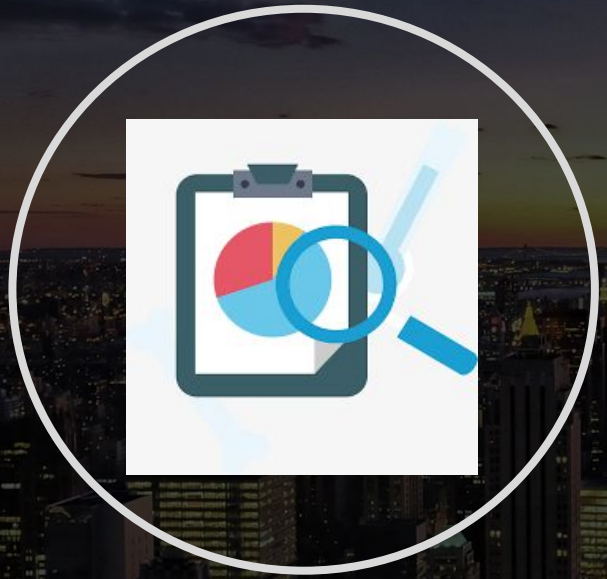
MODEL TRAINING : HIERARCHICAL CLASSIFIER

$$P(n_{l+1}) = \prod_{i=1}^{l} P(n_i).$$

- BINARY HEAP IMPLEMENTATION.
- NODE PROBABILITY OPTIMIZATION.
- TIME COMPLEXITY REDUCTION.

BRIDGING THE GAP TO DEEP LEARNING MODELS

- EXPLICIT WORD-ORDER CONSIDERATION IS COMPUTATIONALLY EXPENSIVE.
- LOCAL WORD ORDER THROUGH BAG OF N-GRAMS.
- HASHING FOR FAST MEMORY LOOKUP.



RESULTS

- 75-25 split of the training data was done for the creation of the Validation Set.
- fastText Parameters were optimized on the Validation Set.

F1 - SCORE

TRAINING SET	**0.8485**
VALIDATION SET	**0.7391**

ANY
QUESTIONS
?