# Reading Comprehension System using Natural Language Processing

Chinnu Pittapally
Mayuri Kadam
Shorabh Dhandharia

# Objective

- To answer questions based on a context paragraph
- Paragraphs are either stories or fact based sentence collection
- Focus: To predict the sentence containing right answer

**Significance:**

- Extension to the Information Retrieval
- Grading System

# Dataset

**Training Set:**

Total Paragraphs : 18896

Total  Question-Answer pairs : 87599

**Sample Data:**

{‘**context**’ : “Paragraph goes here…”,
 '**qas**': [{'answers': [{'answer_start': 248, 'text':
 'September 1876'}]
'**Id**':'5733bf84d058e614000b61be',
'**question**': 'When did the Scholastic Magazine
 of Notre dame begin publishing?'}]

Questions Counts by Category

| | | | |
|---|---|---|---|
| What | 48705 | Whom | 230 |
| Where | 3766 | Why | 1246 |
| Which | 6458 | How | 9509 |
| Who | 9034 | When | 6667 |

# Approach - N-gram Rule Based

- Read paragraph and each question; pre-process (remove stopwords, lower text, word_tokenize)
- Collect sentences for which the n-gram match is >=  threshold
- QuestToTag = {"when":'CD',

```
"who":['B-PERSON','I-PERSON'],
"whom":['B-PERSON','I-PERSON'],
"where":['B-GPE','I-GPE','B-ORG','I-ORG'],
"what":['B-PERSON','I-PERSON','B-ORG','I-ORG'],
"what place":['B-GPE','I-GPE','B-ORG','I-ORG'],
"how many":"CD"}
```

# Approach - WMD Similarity

- **Word Mover's Distance**

  Dissimilarity between two documents as minimum amount of distance needed to travel from words in one document to the other

- **How we use WMD?**

  We built two corpuses:

  1. All paragraphs in a topic to create a WMD model
  2. Paragraph against which the questions are asked

  We selected 4 sentences with highest WMD score with the question

$$\min_{\mathbf{T}\geq 0} \sum_{i,j=1}^{n} \mathbf{T}_{ij} c(i,j)$$

# Models

**Logistic Regression**

- {'C': 0.1, 'class_weight': 'balanced', 'penalty': 'l1', 'random_state': 2}

**MLP Classifier**

- Hidden Layer, Hidden Layer units => (10, 4)

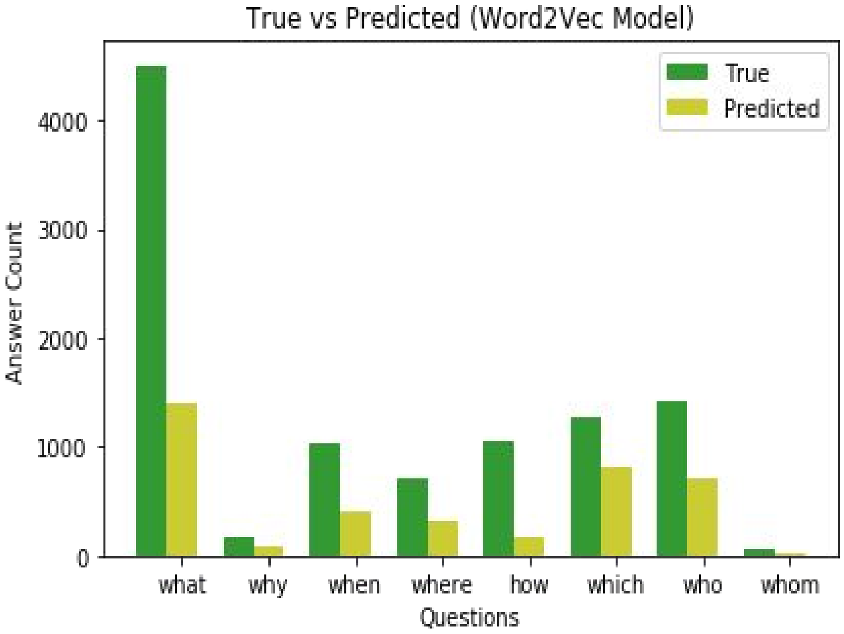
**Keras Sequential**

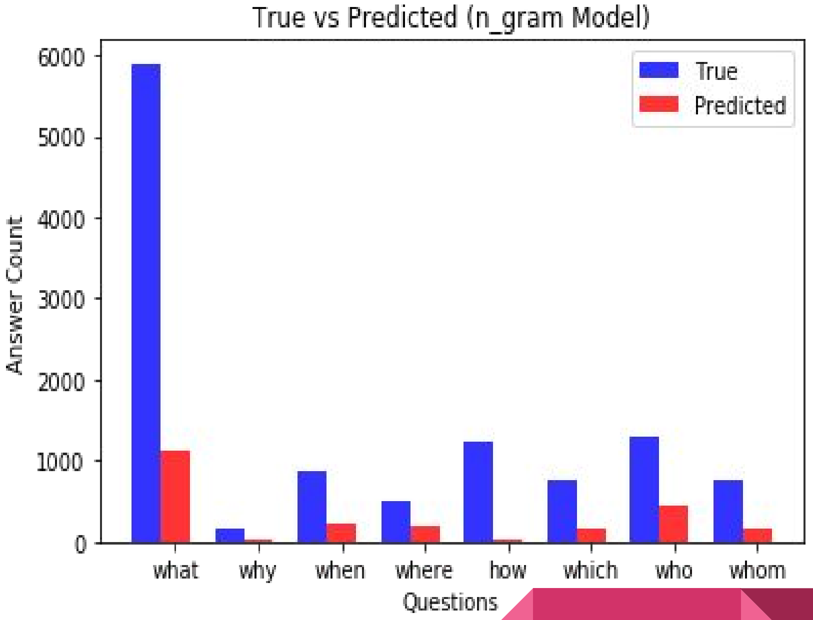
- Hidden Layer, Hidden Layer units => (1, 10)

# Results

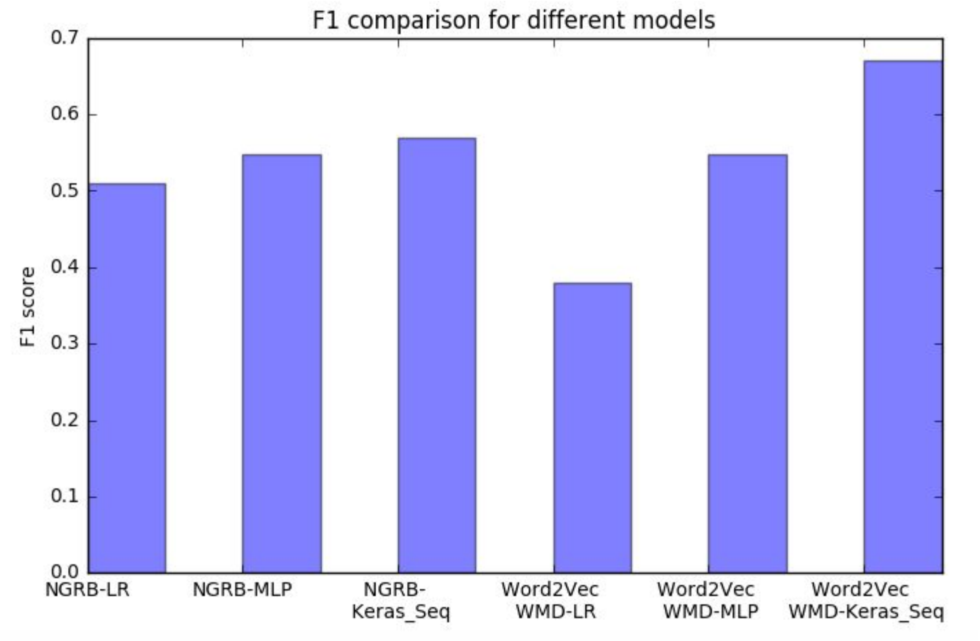
| Model Approach | Accuracy on Training | Precision on Test | Recall on Test | F1 on Test |
|---|---|---|---|---|
| N-gram rule based - Logistic Regression | 0.75411 | 0.7447 | 0.7447 | 0.5091 |
| N-gram rule based - MLP Classifier | 0.5626 | 0.4790 | 0.6374 | 0.5470 |
| N-gram rule based - Keras Sequential | 0.7216 | 0.59 | 0.55 | 0.57 |
| Word2Vec WMD - Logistic Regression | 0.5673 | 0.2919 | 0.5442 | 0.3800 |
| Word2Vec WMD - MLP Classifier | 0.5626 | 0.4790 | 0.6374 | 0.5470 |
| Word2Vec WMD - Keras Sequential | 0.7693 | 0.60 | 0.77 | 0.67(avg) |

# Results Contd.



# Comparison of Different Models



# Model Errors

**Q3 : Who was the game's top receiver?**

**A1 :** **Sanders** was his top receiver with six receptions for 83 yards.

**Predicted: True**

**A2 :** Anderson was the game's leading rusher with 90 yards and a touchdown, along with four receptions for 10 yards.

**Predicted: True**

**A3 :** Manning finished the game 13 of 23 for 141 yards with one interception and zero touchdowns.

**Predicted: True**

# Conclusion

- N-gram approach performed qualitatively better compared to all other models but it could rarely predict the ‘What’ category questions.
- WMD similarity enabled us to find the possible answers for ‘What’ categories
- N-gram approach shows a high performance using the Keras Sequential model
- ‘Which’, ‘What’ and ‘When’ categories performed better in the word2vec approach
- ‘Where’, ’Who’, ’When’ categories performed better using n-gram approach

# Future Work

Following implementation could improve our performance:

- Coreference between entities
- Deep Learning
- Ensemble Learning and classification

# Thank You !

## Questions??