# WebNLG Challenge 2019 - Phase 2

Bodhisatwa Chatterjee, Alexandar Devic, Sree Sai Teja Lanka, and Alexander Malis

# Overview of Presentation

1. Project Philosophy and Ideas
2. Phase A - (40 models)
    a. Overview and Results
3. Vocab Generation Overview for Phase B
4. Phase B - (60 models)
    a. Overview and Results
5. Phase C - (64 models)
    a. Overview and Results
6. Final Results
7. Future Work

# Project Philosophy and Idea

- Test many models
  - Three phases: A, B, C

# Project Philosophy and Idea

- Test many models
    - Three phases: A, B, C
        - Standard model with ideas brought from Phase 1 project
            - Train/Test split and non-delexicalized parsing

# Project Philosophy and Idea

- Test many models
  - Three phases: A, B, C
    - Standard model with ideas brought from Phase 1 project
      - Train/Test split and non-delexicalized parsing
    - Special Vocabulary generation and parsing methods
      - 5 Methods, 1 superior

# Project Philosophy and Idea

- Test many models
  - Three phases: A, B, C
    - Standard model with ideas brought from Phase 1 project
      - Train/Test split and non-delexicalized parsing
    - Special Vocabulary generation and parsing methods
      - 5 Methods, 1 superior
    - Further refined parsing methods and ideas
      - Delexicalization and advanced models

# Project Philosophy and Idea

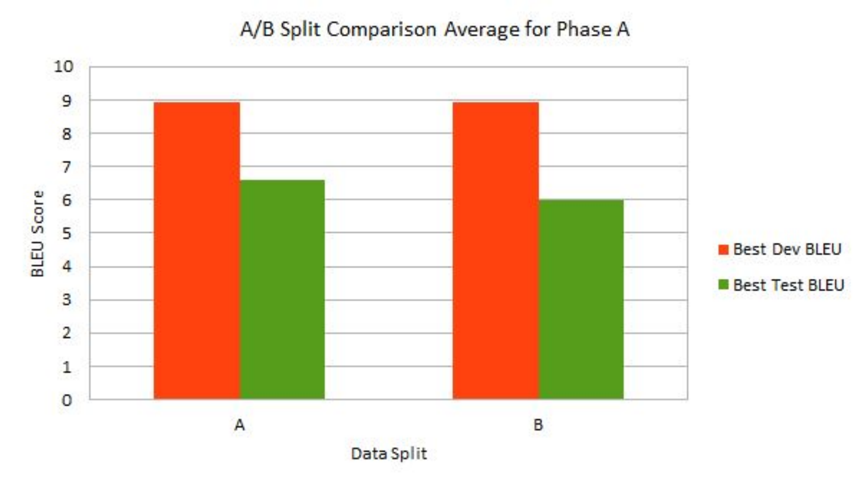
- Test many models
  - Three phases: A, B, C
    - Standard model with ideas brought from Phase 1 project
      - Train/Test split and non-delexicalized parsing
    - Special Vocabulary generation and parsing methods
      - 5 Methods, 1 superior
    - Further refined parsing methods and ideas
      - Delexicalization and advanced models
  - Each phase has a fundamentally different approach to the problem, but utilizes and carries the best models from the prior phase

# Project Philosophy and Idea

- Test many models
  - Three phases: A, B, C
    - Standard model with ideas brought from Phase 1 project
      - Train/Test split and non-delexicalized parsing
    - Special Vocabulary generation and parsing methods
      - 5 Methods, 1 superior
    - Further refined parsing methods and ideas
      - Delexicalization and advanced models
  - Each phase has a fundamentally different approach to the problem, but utilizes and carries the best models from the prior phase
- Picked the best
  - Tested over 170 models!

# Train/Test Split - Phase A

- Data was split two ways
  - A = 10% dev, 80% train, 10% test
  - B = 10% dev, 85% train, 5% test
- Split was done by iterating over entries and maintaining a count of modulo 20, with specific values assigned to either set.
- This ensured that A.Dev, A.Test, B.Dev and B.Test had no overlap.
- After Phase A, split A and B were close and were carried into testing for Phase B



# Vocab Generation

- Five parse methods were used. With the following example, the methods will use the following vocab words: ***"Alan Bean, a crew member of NASA's Apollo 12, was.born March 15, 1932."***

# Vocab Generation

- Five parse methods were used. With the following example, the methods will use the following vocab words: ***"Alan Bean, a crew member of NASA's Apollo 12, was.born March 15, 1932."***
  a. SPACE SPLIT (Method 0)

```
{"Alan", "Bean,", "a", "crew", "member", "of", "NASA's", "Apollo", "12,", "was.born", "March", "15,", "1932."
```

# Vocab Generation

- Five parse methods were used. With the following example, the methods will use the following vocab words: ***“Alan Bean, a crew member of NASA’s Apollo 12, was.born March 15, 1932.”***
  a. SPACE SPLIT (Method 0)

     `{"Alan", "Bean,", "a", "crew", "member", "of", "NASA’s", "Apollo", "12,", "was.born", "March", "15,", "1932.'`

  b. NO PUNCT (Method 1)

     `{"Alan", "Bean", "a", "crew", "member", "of", "NASA’s", "Apollo", "12", "was.born", "March", "15", "1932"`

# Vocab Generation

- Five parse methods were used. With the following example, the methods will use the following vocab words: ***"Alan Bean, a crew member of NASA's Apollo 12, was.born March 15, 1932."***
    a. SPACE SPLIT (Method 0)

    ```
    {"Alan", "Bean,", "a", "crew", "member", "of", "NASA's", "Apollo", "12,", "was.born", "March", "15,", "1932.'
    ```

    b. NO PUNCT (Method 1)

    ```
    {"Alan", "Bean", "a", "crew", "member", "of", "NASA's", "Apollo", "12", "was.born", "March", "15", "1932'
    ```

    c. NO PUNCT POSSESSIVE (Method 2)

    ```
    {"Alan", "Bean", "a", "crew", "member", "of", "NASA", "Apollo", "12", "was.born", "March", "15", "1932'
    ```

# Vocab Generation

- Five parse methods were used. With the following example, the methods will use the following vocab words: ***"Alan Bean, a crew member of NASA's Apollo 12, was.born March 15, 1932."***
    a. SPACE SPLIT (Method 0)

    `{"Alan", "Bean,", "a", "crew", "member", "of", "NASA’s", "Apollo", "12,", "was.born", "March", "15,", "1932.'`

    b. NO PUNCT (Method 1)

    `{"Alan", "Bean", "a", "crew", "member", "of", "NASA’s", "Apollo", "12", "was.born", "March", "15", "1932'`

    c. NO PUNCT POSSESSIVE (Method 2)

    `{"Alan", "Bean", "a", "crew", "member", "of", "NASA", "Apollo", "12", "was.born", "March", "15", "1932'`

    d. NO PUNCT FIX (Method 3)

    `{"Alan", "Bean", "a", "crew", "member", "of", "NASA’s", "Apollo", "12", "was", "born", "March", "15", "1932'`

# Vocab Generation

- Five parse methods were used. With the following example, the methods will use the following vocab words: ***"Alan Bean, a crew member of NASA's Apollo 12, was.born March 15, 1932."***
  a. SPACE SPLIT (Method 0)

     `{"Alan", "Bean,", "a", "crew", "member", "of", "NASA's", "Apollo", "12,", "was.born", "March", "15,", "1932."`

  b. NO PUNCT (Method 1)

     `{"Alan", "Bean", "a", "crew", "member", "of", "NASA's", "Apollo", "12", "was.born", "March", "15", "1932"`

  c. NO PUNCT POSSESSIVE (Method 2)

     `{"Alan", "Bean", "a", "crew", "member", "of", "NASA", "Apollo", "12", "was.born", "March", "15", "1932"`

  d. NO PUNCT FIX (Method 3)

     `{"Alan", "Bean", "a", "crew", "member", "of", "NASA's", "Apollo", "12", "was", "born", "March", "15", "1932"`

  e. NO PUNCT POSSESSIVE FIX (Method 4)

     `{"Alan", "Bean", "a", "crew", "member", "of", "NASA", "Apollo", "12", "was", "born", "March", "15", "1932"`

# Vocab Generation

- Five parse methods were used. With the following example, the methods will use the following vocab words: ***"Alan Bean, a crew member of NASA's Apollo 12, was.born March 15, 1932."***
    a. SPACE SPLIT (Method 0)

    `{"Alan", "Bean,", "a", "crew", "member", "of", "NASA's", "Apollo", "12,", "was.born", "March", "15,", "1932."`

    b. NO PUNCT (Method 1)

    `{"Alan", "Bean", "a", "crew", "member", "of", "NASA's", "Apollo", "12", "was.born", "March", "15", "1932"`

    c. NO PUNCT POSSESSIVE (Method 2)

    `{"Alan", "Bean", "a", "crew", "member", "of", "NASA", "Apollo", "12", "was.born", "March", "15", "1932"`

    d. NO PUNCT FIX (Method 3)

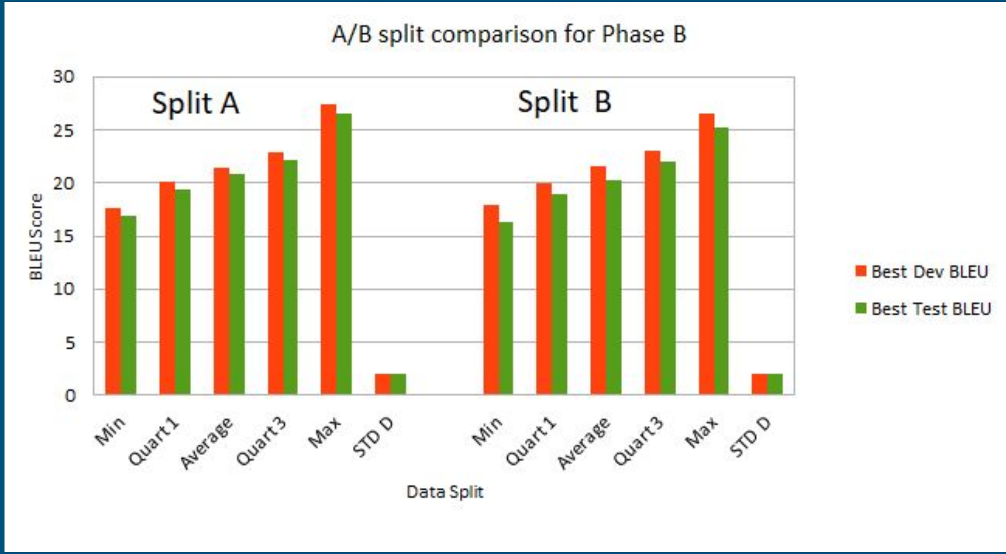
    `{"Alan", "Bean", "a", "crew", "member", "of", "NASA's", "Apollo", "12", "was", "born", "March", "15", "1932"`

    e. NO PUNCT POSSESSIVE FIX (Method 4)

    `{"Alan", "Bean", "a", "crew", "member", "of", "NASA", "Apollo", "12", "was", "born", "March", "15", "1932"`
- The SPACE SPLIT method yielded the best BLEU score surprisingly. RDF vocab generation was trivial.

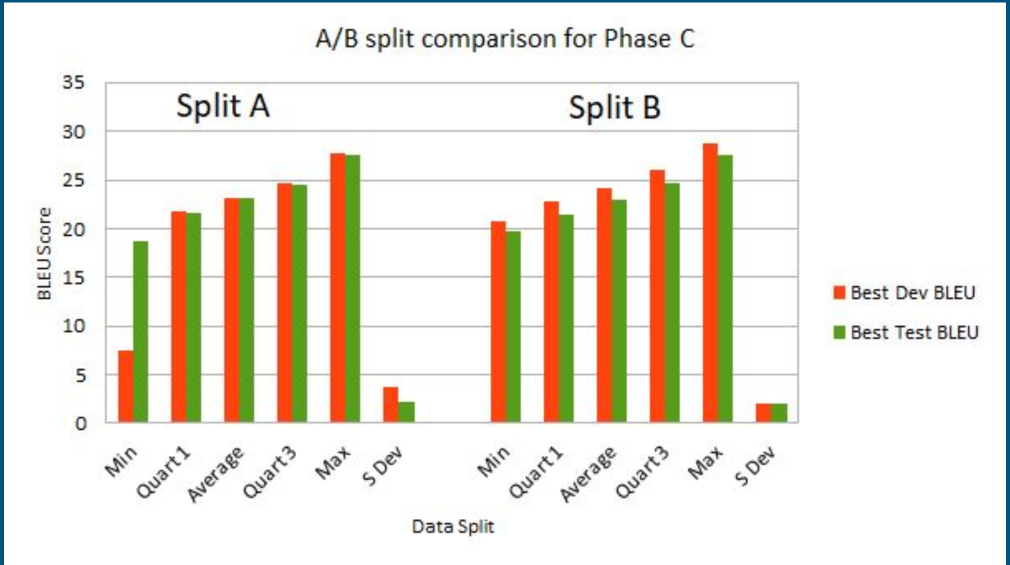
# Train/Test Split - Phase B

- In Phase B, split A and B were paired with vocab 0 to 4 and tested a total of 60 times. The results were compared and split B was slightly ahead of split A by an average value of .5 BLEU score.
- The top two models advanced to Phase C, which were V0-A and V0-B



# Train/Test Split - Phase C

- In phase C, the data was again very similar but for larger models B pulled ahead. Split A had poor performance in some of the more complex models leading to drastically reduced minimum scores.
- This leaves B as the prefered data split.



# Models - Phase B/C

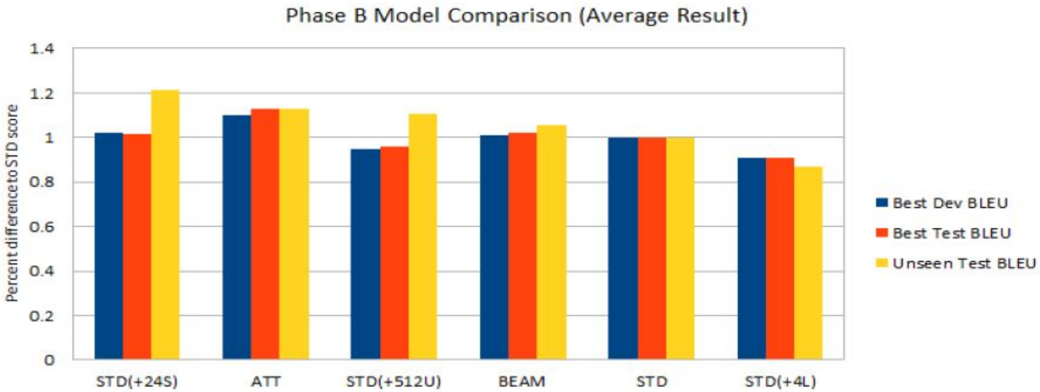
Phase B

- Tier 0: Base model
    - Test: 20.43, (None changed)
- Tier 1: One param changed
    - Max Test, 23.07 (ATT)

Phase C

- Tier 2: Two params changed
    - Max Test. 26.8 (ATT,24S)
- Tier 3: Three params changed
    - Max Test. 27.6 (BEAM,ATT,24S)
- Tier 4: Four params changed
    - Max test, 26.8 (BEAM,ATT,24S,512U)
- Tier 5: All params changed
    - Max test, 23.7 (All changed)

| Key % | Model (Avg) | Dev Perplexity | Best Dev BLEU | Test Perplexity | Best Test BLEU | Unseen Test |
|---|---|---|---|---|---|---|
| 1st | STD(+24S) | 109.16% | 102.36% | 110.34% | 101.52% | 121.09% |
| 2nd | ATT | 91.41% | 109.77% | 91.80% | 112.92% | 112.75% |
| 3rd | STD(+512U) | 123.37% | 94.77% | 123.23% | 95.79% | 110.90% |
| 4th | BEAM | 99.98% | 100.79% | 99.48% | 102.25% | 105.78% |
| 5th | STD | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| 6th | STD(+4L) | 109.53% | 90.83% | 113.35% | 90.80% | 87.02% |



# Progress till now.....

- Till now focus was to find the ‘best’ model

# Progress till now.....

- Till now focus was to find the ‘best’ model
- Model with beam search, attention-mechanism, 24,000 steps and 512 units.

# Progress till now.....

- Till now focus was to find the ‘best’ model
- Model with beam search, attention-mechanism, 24,000 steps and 512 units.
- Need to make this model ‘better’ in terms of generalized predictions

# Delexicalization - Phase C

- Objective is to generalize the ‘context’ of a word better

# Delexicalization - Phase C

- Objective is to generalize the ‘context’ of a word better
- Defined ‘context’ of word - specific category to which it belongs
- *‘Bob | loves | America’*
- *‘Villanelle | speaks | Russian’*

# Delexicalization - Phase C

- Objective is to generalize the 'context' of a word better
- Defined 'context' of word - specific category to which it belongs
- *'Bob | loves | America'*
- *'Villanelle | speaks | Russian'*

- Delexicalization means replacement of specific words, both in lex and triples, with the category it is associated with.
- *'Name | loves | Country'* and *'Name | speaks | Language'*

# Why is this non-trivial?

- Need to know categories for as many words as possible (complete delexicalization)

# Why is this non-trivial?

- Need to know categories for as many words as possible (complete delexicalization)
- Task of associating words with their categories is subjective
- *‘Godiva | Chocolate | Belgian’* and *‘Courtois | speaks | Belgian’*

# Why is this non-trivial?

- Need to know categories for as many words as possible (complete delexicalization)
- Task of associating words with their categories is subjective
- *'Godiva | Chocolate | Belgian'* and *'Courtois | speaks | Belgian'*

- Goal is to define categories which groups as many words as possible in itself

# Categories for De-lexicalization

- Languages and Countries : countries and languages list were built with python module pycountry. Any specific country occurring in either the lex comments or the RDF triple was replaced by the keyword 'COUNTRY'.

# Categories for De-lexicalization

- Languages and Countries : countries and languages list were built with python module pycountry. Any specific country occurring in either the lex comments or the RDF triple was replaced by the keyword 'COUNTRY'.
- Animals : comprehensive list of all animals was made from *Colorado State University Libraries* and all occurrences were replaced by keyword 'ANIMAL'.

# Categories for De-lexicalization

- Languages and Countries : countries and languages list were built with python module pycountry. Any specific country occurring in either the lex comments or the RDF triple was replaced by the keyword 'COUNTRY'.
- Animals : comprehensive list of all animals was made from *Colorado State University Libraries* and all occurrences were replaced by keyword 'ANIMAL'.
- Airports : specific category in our data, we delexicalized all names of airports occurring in our data by the keyword 'AIRPORT'. A list of airport names were prepared from Dbpedia.

# Categories for De-lexicalization

- Languages and Countries : countries and languages list were built with python module pycountry. Any specific country occurring in either the lex comments or the RDF triple was replaced by the keyword 'COUNTRY'.
- Animals : comprehensive list of all animals was made from *Colorado State University Libraries* and all occurrences were replaced by keyword 'ANIMAL'.
- Airports : specific category in our data, we delexicalized all names of airports occurring in our data by the keyword 'AIRPORT'. A list of airport names were prepared from Dbpedia.
- Astronauts : list of all the names of astronauts was made from Wikipedia and all the occurrences of names of astronauts were replaced by the keyword 'ASTRONAUT'.

# Performance of De-lexicalized Models

- ‘Incremental approach’ was followed

# Performance of De-lexicalized Models

- ‘Incremental approach’ was followed
- Delex_1 Score: 45.6 (Only Language and Countries were used)

# Performance of De-lexicalized Models

- ‘Incremental approach’ was followed
- Delex_1 Score: 45.6 (Only Language and Countries were used)
- Delex_2 Score: 44.9 (Language, Countries, Airports)

# Performance of De-lexicalized Models

- ‘Incremental approach’ was followed
- Delex_1 Score: 45.6 (Only Language and Countries were used)
- Delex_2 Score: 44.9 (Language, Countries, Airports)
- Delex_3 Score: 33.5 (Language, Countries, Airports, Animals)

# Performance of De-lexicalized Models

- ‘Incremental approach’ was followed
- Delex_1 Score: 45.6 (Only Language and Countries were used)
- Delex_2 Score: 44.9 (Language, Countries, Airports)
- Delex_3 Score: 33.5 (Language, Countries, Airports, Animals)
- Delex_4 Score: 31 (Language, Countries, Airports, Animals, Astronauts)

# Performance of De-lexicalized Models

- ‘Incremental approach’ was followed
- Delex_1 Score: 45.6 (Only Language and Countries were used)
- Delex_2 Score: 44.9 (Language, Countries, Airports)
- Delex_3 Score: 33.5 (Language, Countries, Airports, Animals)
- Delex_4 Score: 31 (Language, Countries, Airports, Animals, Astronauts)

- Q: Isn’t delex is supposed to boost the scores?

# Performance of De-lexicalized Models

- ‘Incremental approach’ was followed
- Delex_1 Score: 45.6 (Only Language and Countries were used)
- Delex_2 Score: 44.9 (Language, Countries, Airports)
- Delex_3 Score: 33.5 (Language, Countries, Airports, Animals)
- Delex_4 Score: 31 (Language, Countries, Airports, Animals, Astronauts)

- Q: Isn’t delex is supposed to boost the scores?
- A: With more and more categories expressed, less expressive sentences are generated (Bleu score ratio will decrease) (Re-lexicalization Solution)

# Final Results

| Team | Bleu Score |
|---|---|
| **Our Model** | **46.2** |
| MELBOURNE | 45.13 |
| TILB-SMT | 44.28 |
| PKUWRITER | 39.88 |
| UPF-FORGE | 38.65 |
| TILB-PIPELINE | 35.29 |
| TILB-NMT | 34.60 |
| BASELINE | 33.24 |
| ADAPT | 31.06 |
| UIT-VNU | 7.07 |

# Problems and Future Work

- Coverage of more categories in De-lexicalization

# Future Work and Conclusions

- Coverage of more categories in De-lexicalization
- Use of Re-lexicalization

# Future Work and Conclusions

- Coverage of more categories in De-lexicalization
- Use of Re-lexicalization

- Choice of model (nmt/pipeline/smt) should depend on dataset

# Future Work and Conclusions

- Coverage of more categories in De-lexicalization
- Use of Re-lexicalization

- Choice of model (nmt/pipeline/smt) should depend on dataset
- Each phase from data preprocessing to selecting model parameters is as important as the other.