SemEval 2017 Task10: ScienceIE - Extracting Keyphrases from Scientific Publications

CS 6961: Sagar Chaturvedi, Madhur Pandey

School of Computing, University of Utah

THE UNIVERSITY OF UTAH

SCHOOL OF COMPUTING THE UNIVERSITY OF UTAH

SemEval 2017 Task 10: ScienceIE

Two Subtasks:

A: Identification of keyphrases

B: Classification of identified keyphrases as
(i) PROCESS (ii) TASK (iii) MATERIAL

Our Implementation

Algorithms:

- Conditional Random Fields
- Neural Networks based approach using:
  (i) Simple RNN (ii) LSTM (iii) GRU
- Experimented with various dropout, activation and regularizations.

Feature Set:

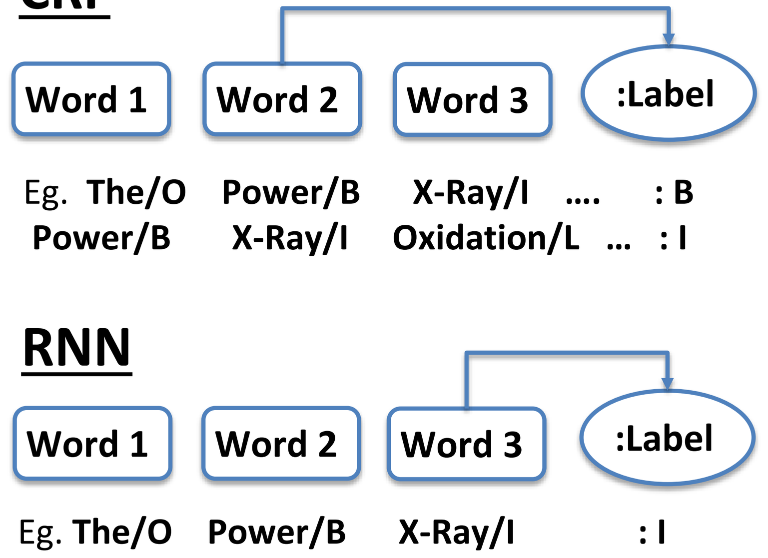
- We trained the CRF using the below features:

3/5 grams	Middle word contains digits or not
Pos tags of all n-grams	Length of middle word of 3/5 gram
For middle word of 3/5 gram, first letter cap or not	For middle word of 3/5 gram, all letters cap or not

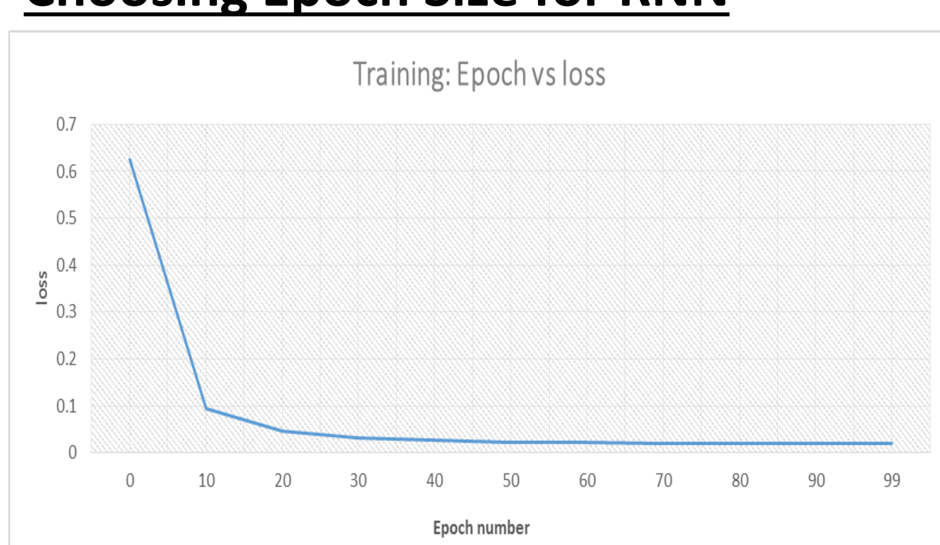
- We only used 3/5 grams feature for training Recurrent Neural Networks

Sub Task A

- We assign the BILO label to each of the words in the article as below:
  Eg. The/O Power/B X-Ray/I Oxidation/L is/O ....
- Generate the above mentioned features based on n-grams
- Label for each n-gram is BILO tag of the central word in case of CRF and the last word in case of RNN.

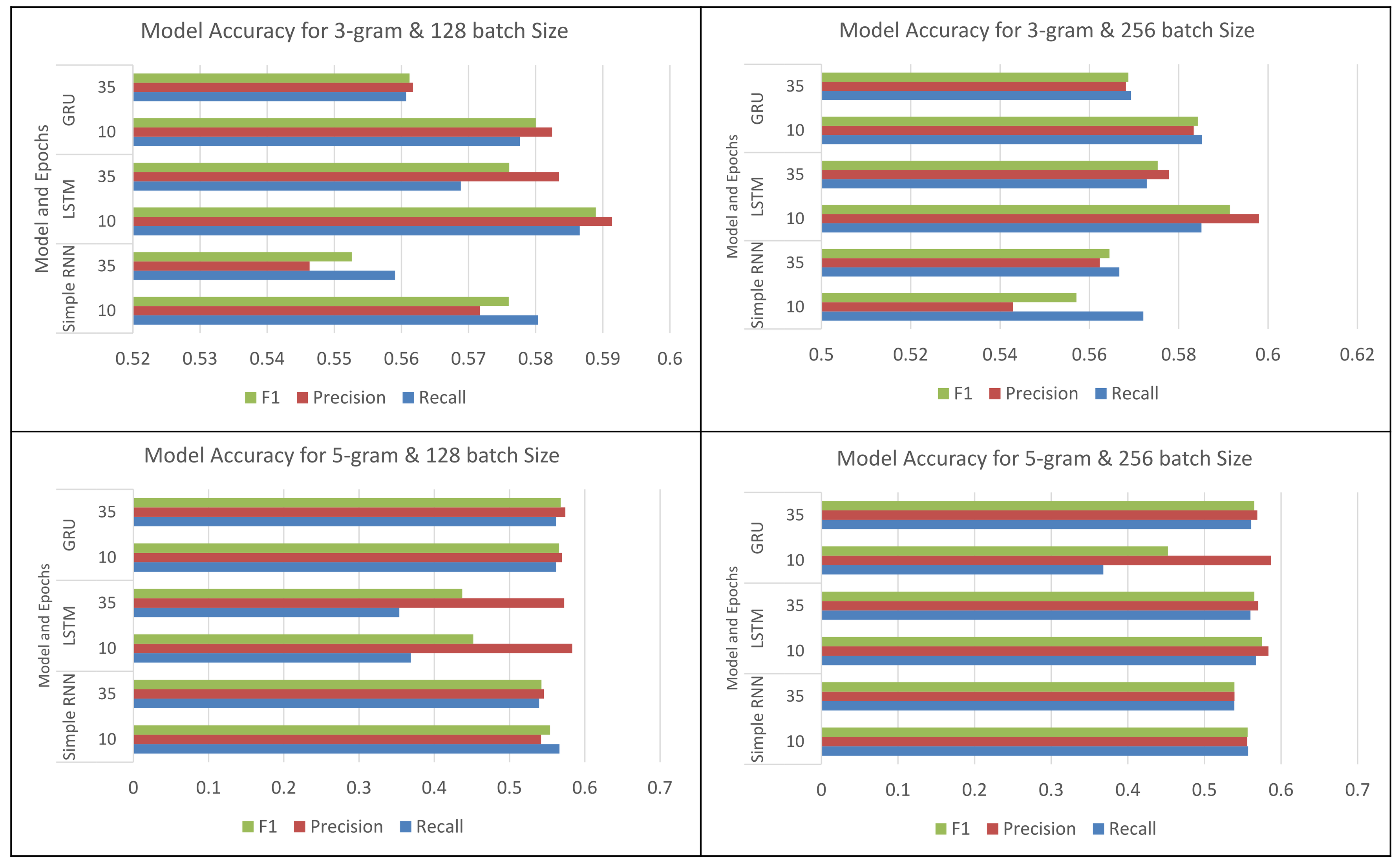


Choosing Epoch Size for RNN



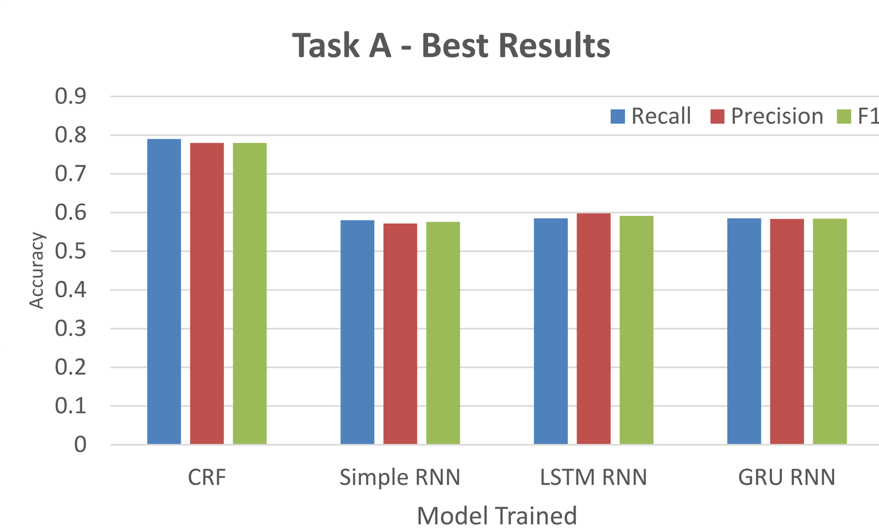
Results for RNNs:

- The 3-gram features performed better than the 5-gram features
- LSTM gave the best results in most cases, but was comparatively slower
- 10 epochs were faster to train and had accuracy close to 35 epochs



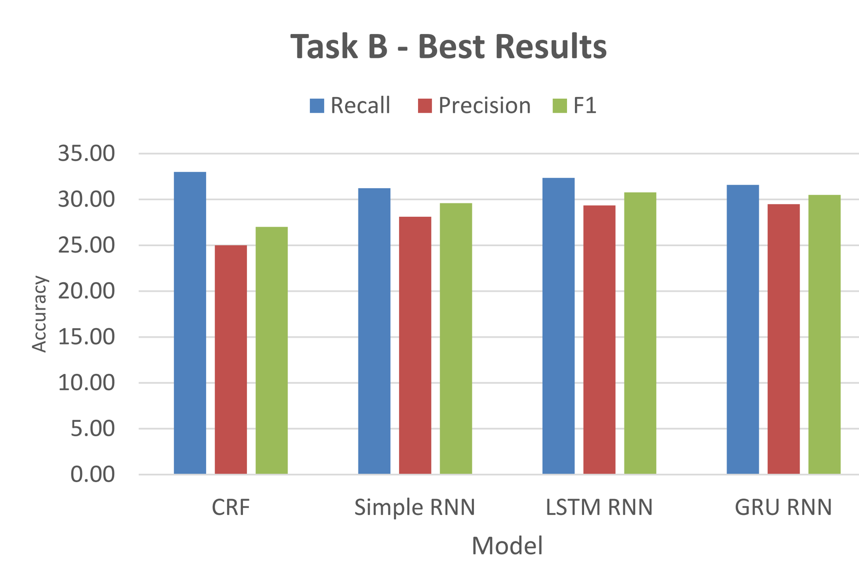
Performance:

- Our results stand 7th as per current leaderboard
- CRF results are for all tags



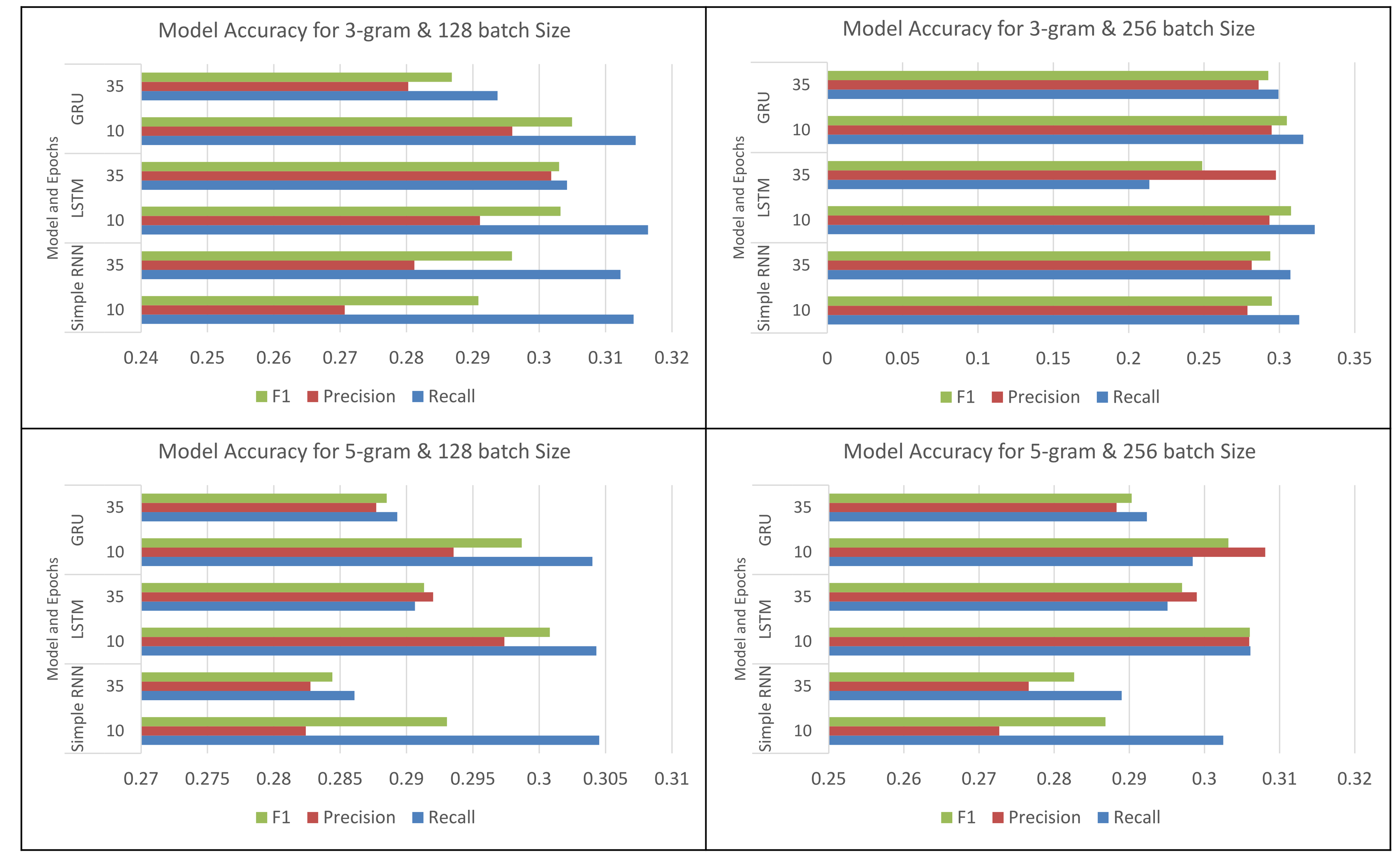
Sub Task B

- Instead of assigning BILO label to each of the words in the article, we assign BILO label to each keyphrase type:
  Eg. The/O Power/B-PROC X-Ray/I-PROC Oxidation/L-PROC is/O ....
- Features and labels are created in similar manner as in Sub Task A



Performance:

- LSTM model gave us the best F1 Score of 30.77 for sub task B
- However, CRF model had the best recall of all the methods.



Analysis

- When trained an RNN separately for named entity type "process", the F1 was 0.5 which is 66% higher than the F1 trained on all named entities together
- 3-gram features gave better results in case of RNN, however, 5-grams worked well for CRF
- Once we removed the training/test words that do not contain any alpha-numeric characters, there was a slight improvement (1.5-2%) in accuracy
- In all the RNN models, 3-gram features outperformed 5-grams feature in terms of both accuracy and speed. Batch size=256 and Epoch=10 worked best

Future Work

- Using CRF on top of RNN to improve accuracy
- Experiments with bi-directional RNNs
- Training separate models for each keyphrase types i.e Process, Material and Task and then combine them
- Train RNN with arbitrary features

Tools

- CRF Suite (https://python-crfsuite.readthedocs.io/en/latest/)
- NLTK (http://www.nltk.org/)
- Keras RNN (https://keras.io/layers/recurrent/)

References

- SemEval2017 Task10 (https://arxiv.org/pdf/1704.02853.pdf)
- CRF for Keyphrase (https://core.ac.uk/download/pdf/11884499.pdf)
- SemEval2010 Task5 (http://www.aclweb.org/anthology/S10-1004)
- Keyphrase Extraction Approaches (http://www.aclweb.org/anthology/W09-2902)