Quantum Twitter Sentiment Analysis

Group Member: Pei-Yong Wang, Rong-Xin Zhu and Rui Xing

Data Preprocessing

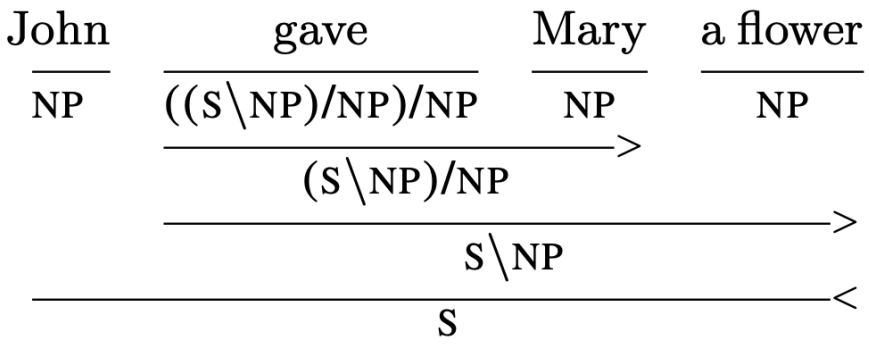
- Remove punctuations;
- Remove emojis;
- From here, we can have three different types of data sets:
  - Doing nothing, no lemmatization (services->service, is->be, …), no stemming (service->servic)
  - Just lemmatization
  - The full suite, stemming after lemmatization

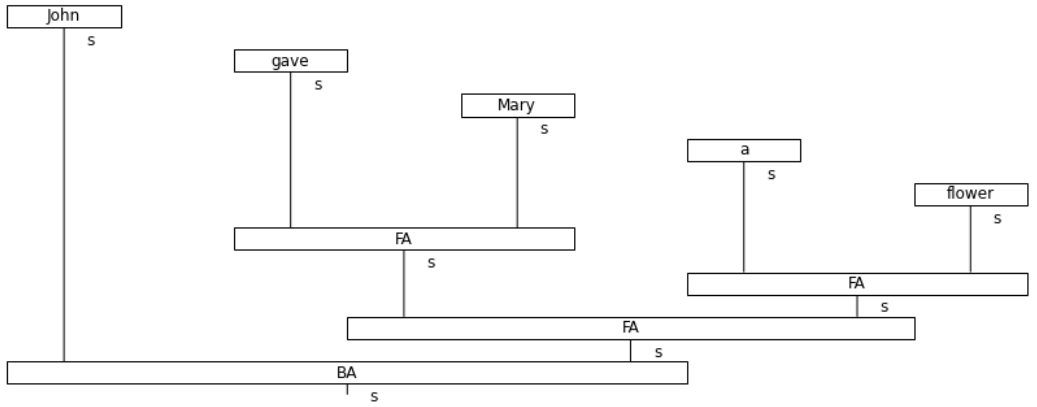
Data Preprocessing

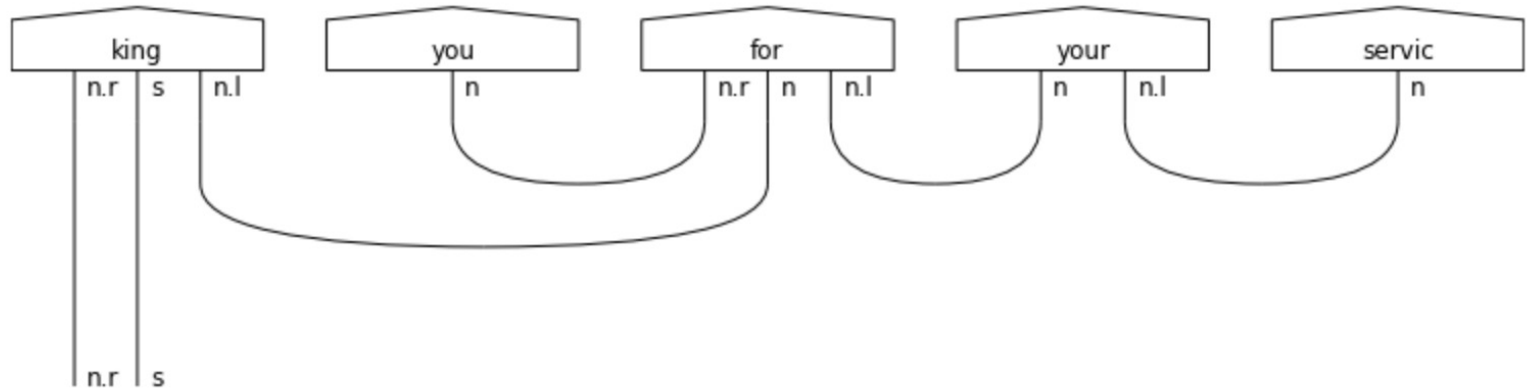
- First we chose 2578 tweets of the whole data set which contains more than 40k tweets, with equal number of positive and negative tweets;
- Due to limited time and computational resources, we randomly selected 206 tweets for training and 26 tweets for validation. The length of each tweet is generally no more than 10;

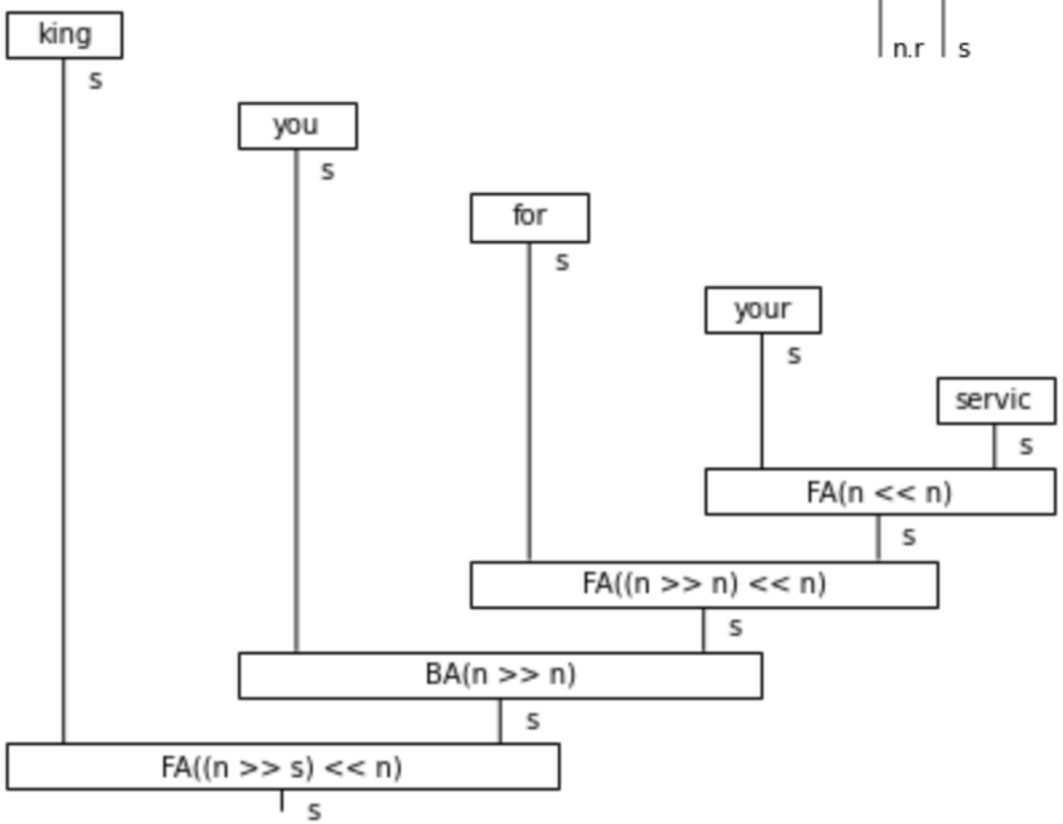
From Sentences to Diagrams

- We apply TreeParser(), a CCG (combinatorial categorical grammar)based parser from the lambeq package for parsing sentences and convert sentences to diagrams, instead of the BobcatParser(), which explicitly apply the pregroup grammar, since BobcatParser() is not good at handling sentences without object.

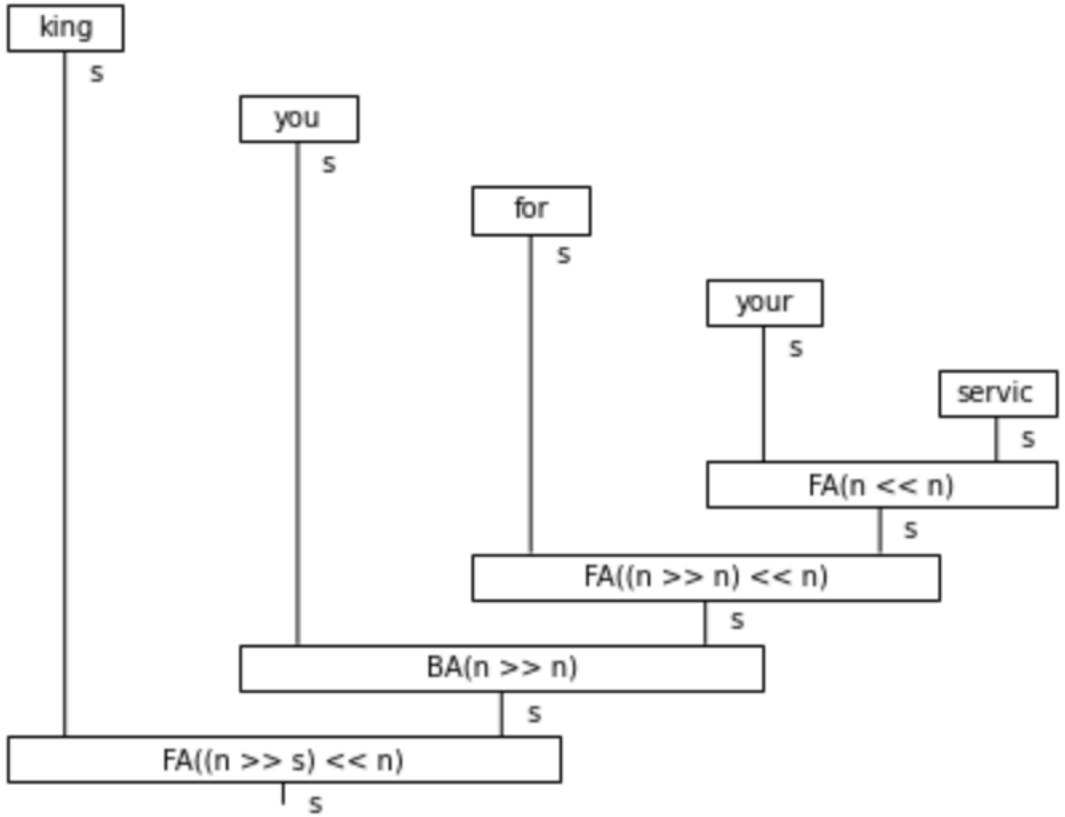


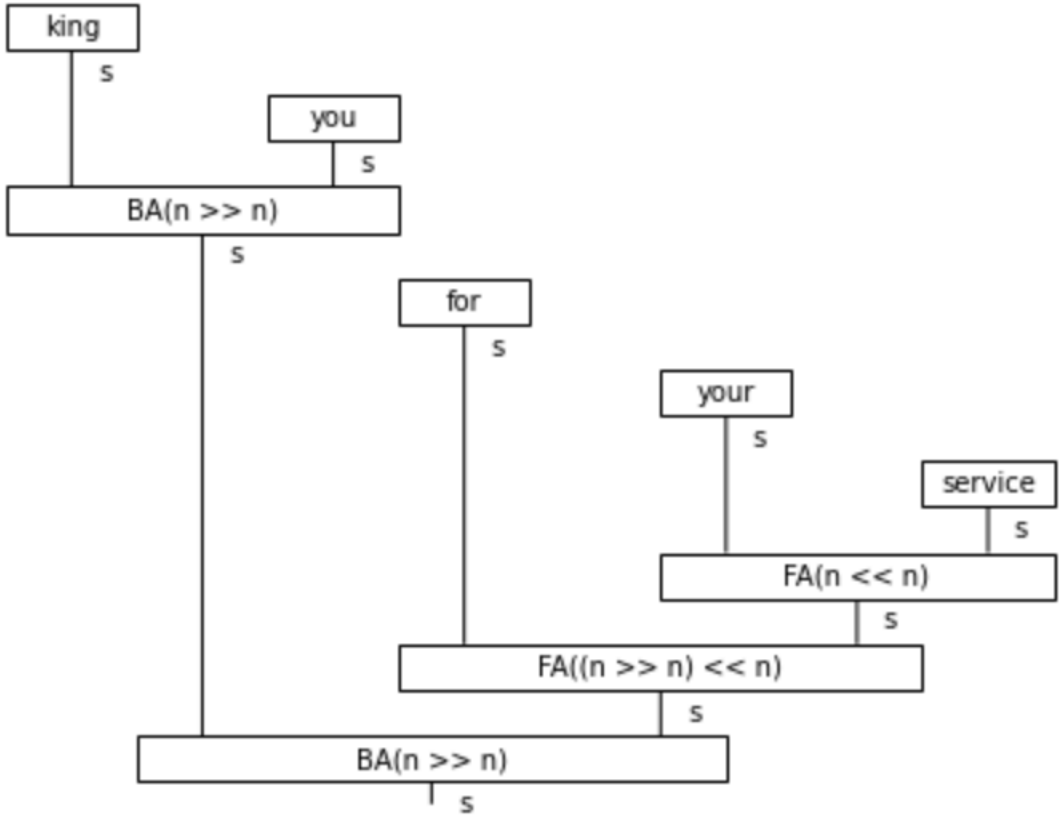



king
you
for
your
servic
n.r
s
n.l
n
n.r
n
n.l
n
n.l
n
n.r
s


king
s
you
s
for
s
your
s
servic
s
FA(n << n)
s
FA((n >> n) << n)
s
BA(n >> n)
s
FA((n >> s) << n)
s

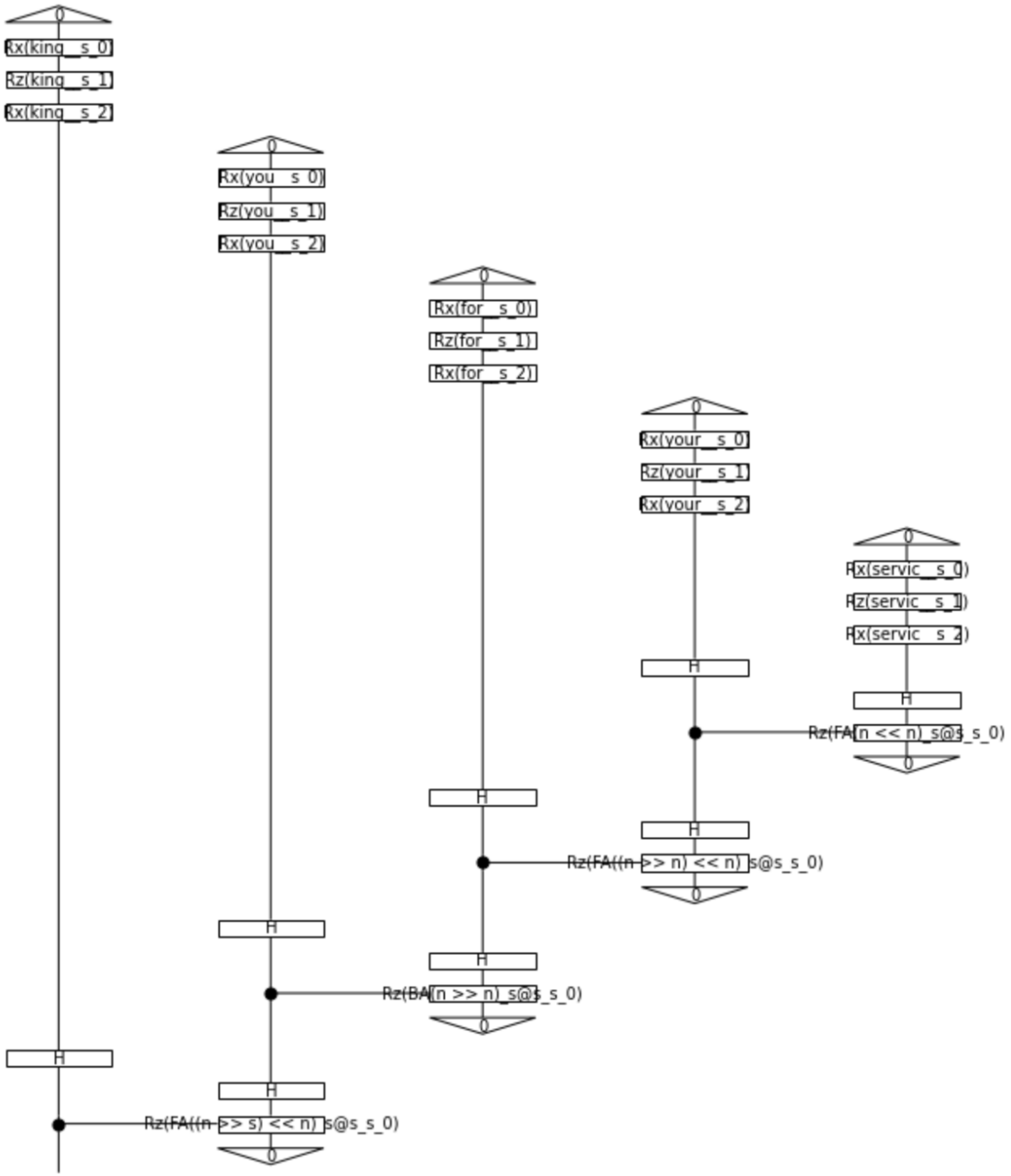
Stemming Can Change Parsing



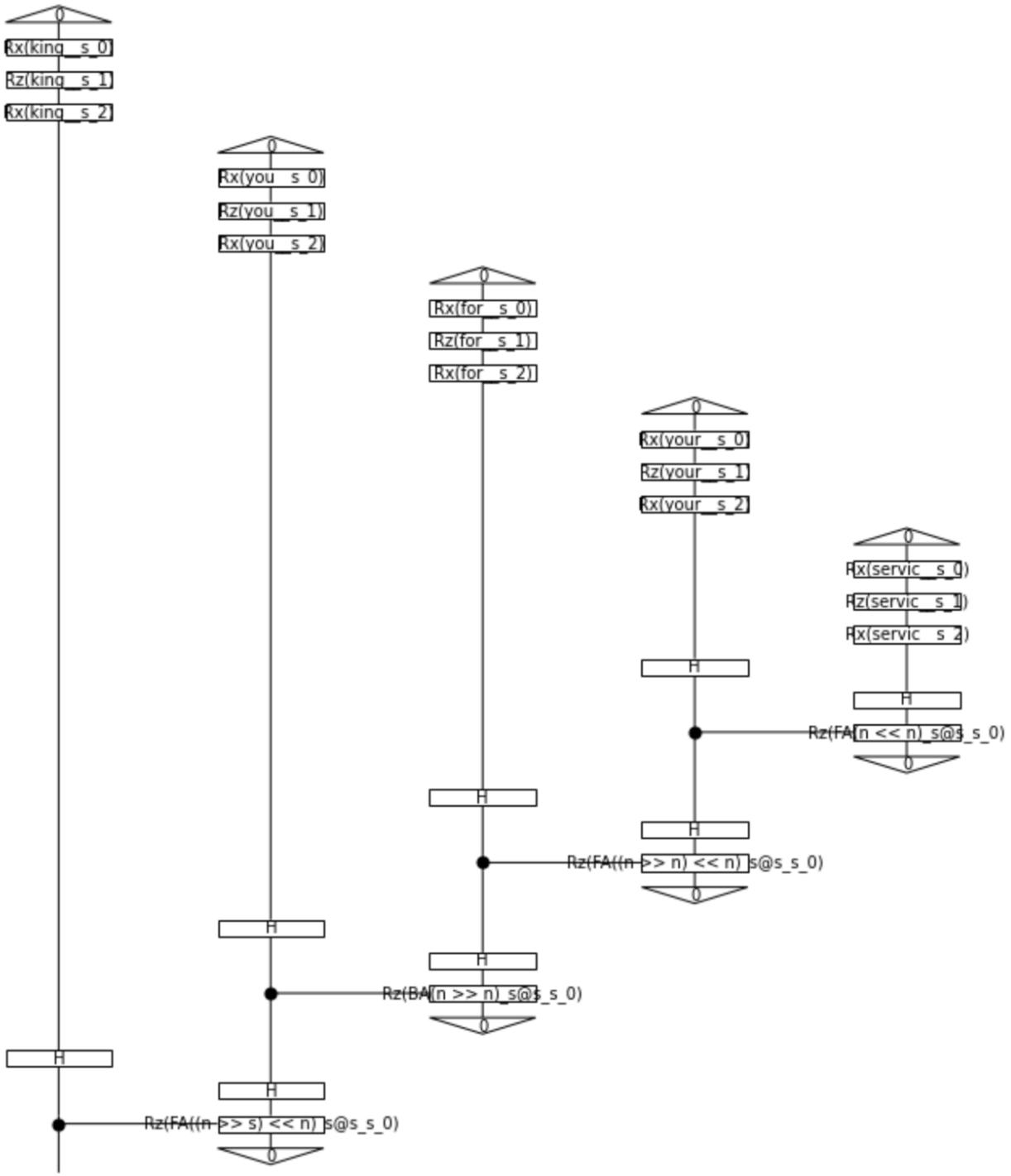


Diagrams to Circuits

- We assign one qubit for each of these following types in lambeq. AtomicType, and then apply the IQP ansatz
  - NOUN, SENTENCE, PREPOSITIONAL_PHRASE, NOUN_PHRASE, CONJUNCTION



In the circuit, each word is represented with a single-qubit unitary RxRzRx, which has three (trainable) parameters. In NLP jargon, these three rotation angles can be viewed as “word embeddings”. The type of composed phases, like “your servic”, are represented by parameterized two-qubit unitaries.

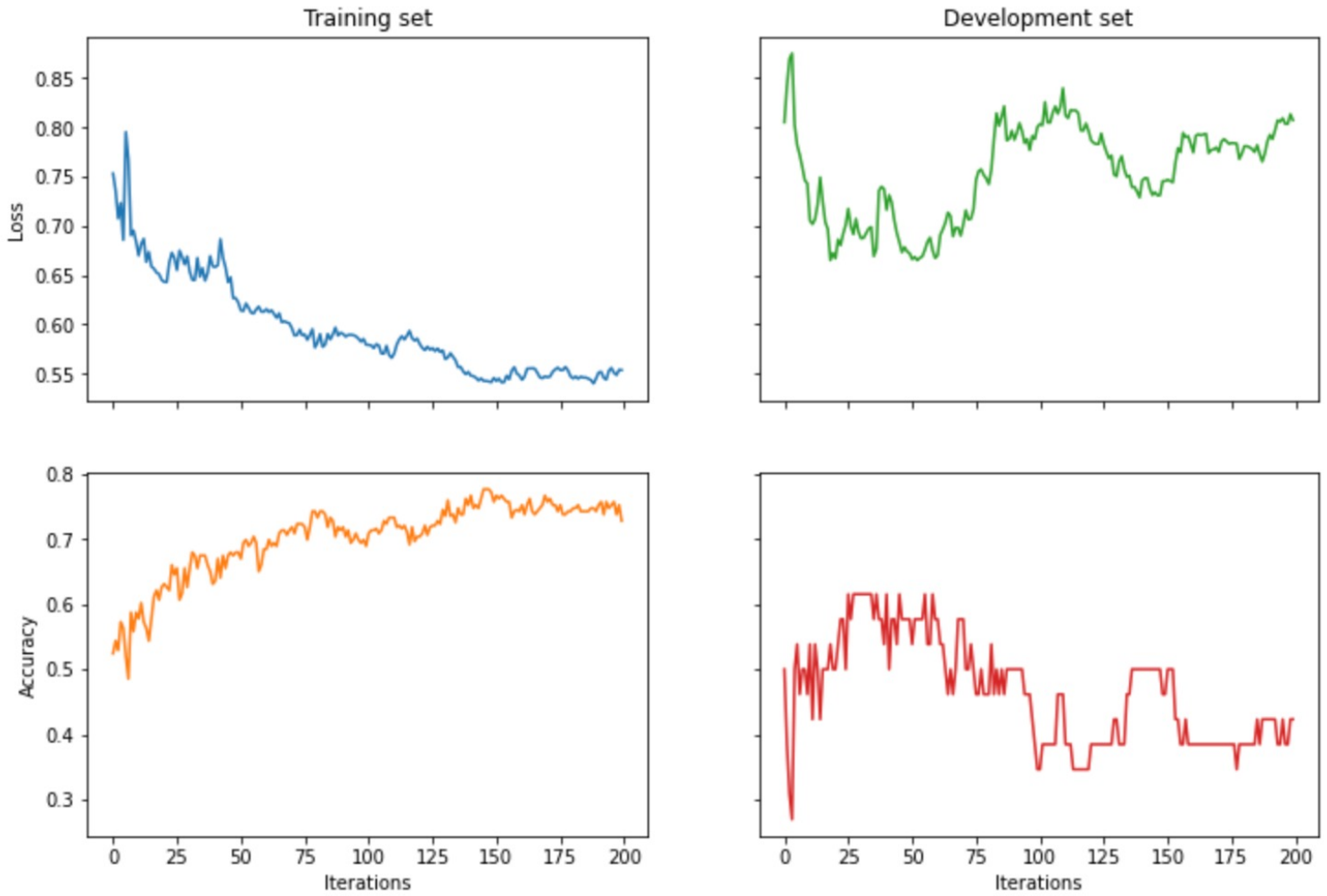


To retrieve classifications, we'll need the probability from the only dangling qubit (the left most one)

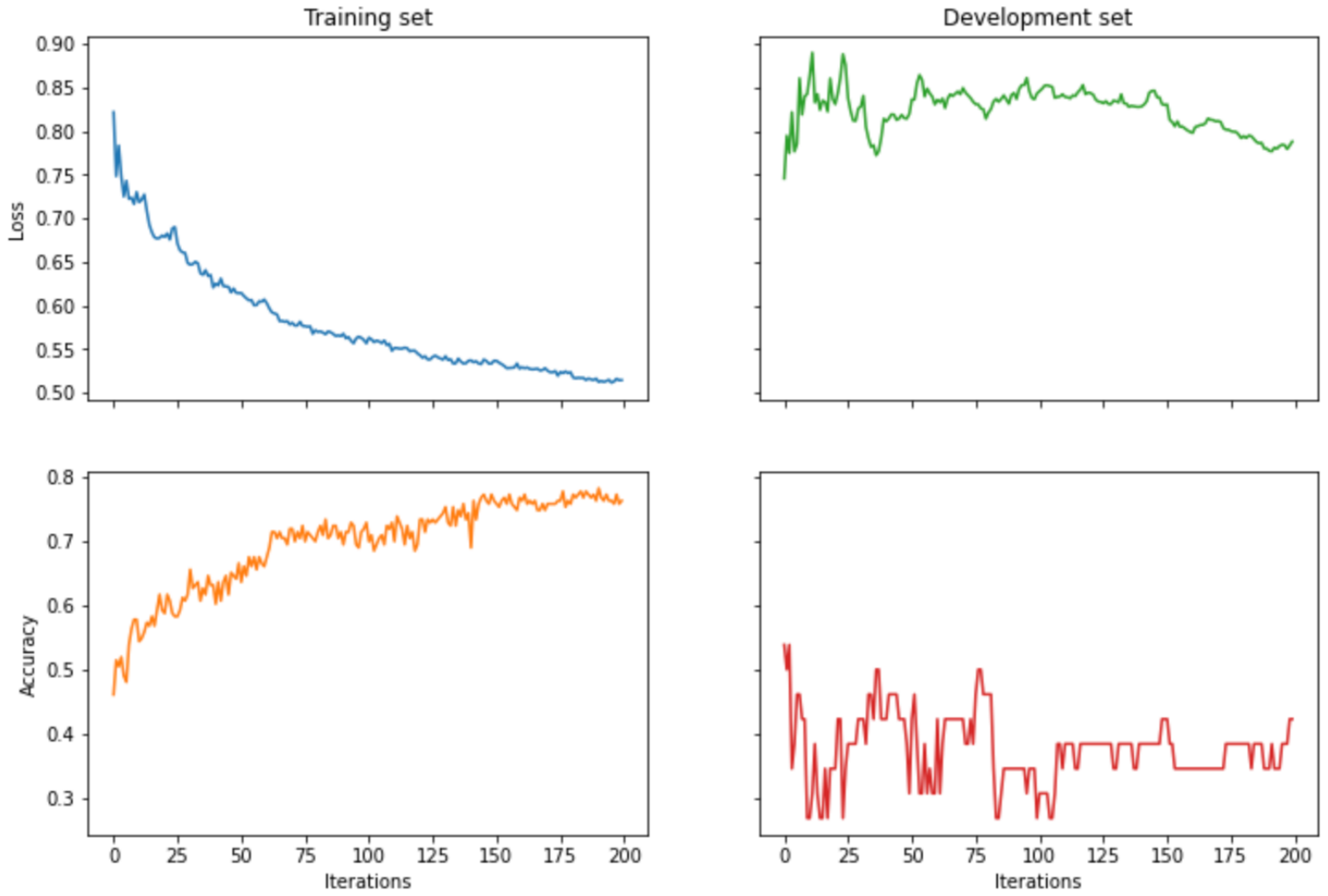
Training the Model

- Loss function: binary cross entropy;
- Parameters for words and types are updated with SPSA;
- Circuit simulated with Jax/tensornetworks (inside lambeq).

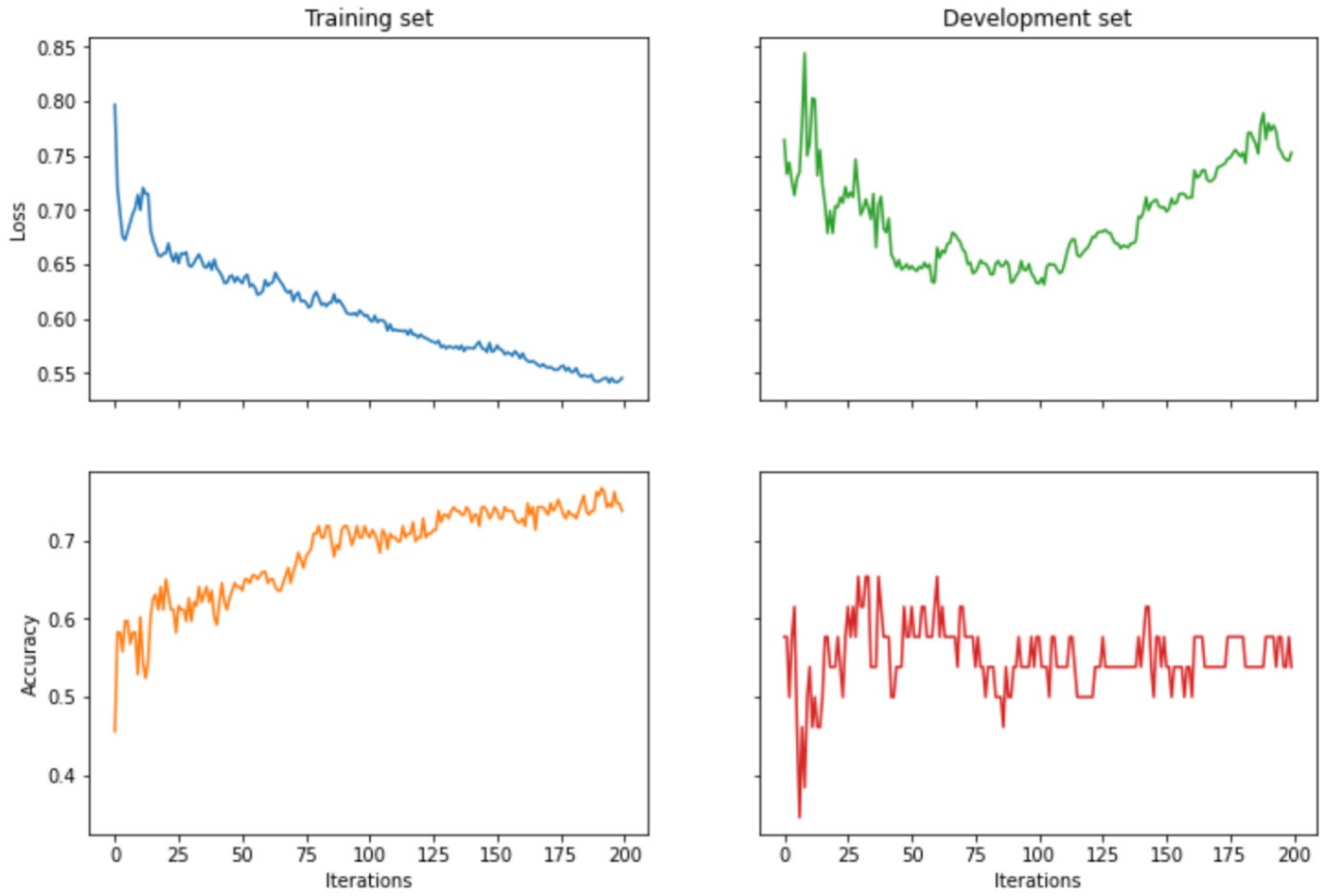
Results: Full-Suite Preprocessed Data



Results: Only Lemmatized Data



Results: No Stemming or Lemmatization



Results

- Size of training data severely impacted on our model performances on the validation set due to limited vocabulary in the training data -> some (maybe most) word embeddings in the validation set are not trained;
- Need to increase training data size;
- Need to increase the speed of applying parameters to circuit in lambeq;
- Another future direction could be combining parsers (and types) from industry-grade NLP software, like SpaCy, with lambeq, to provide more accurate results.