WebNLG Challenge

by: Bayu Distiawan T

A. Data:

Data training contains of set of triples and sentences represent the triples. Example:

```
Triple-1: [(subject: “Donald Trump”), (property: “birthDate”), (object:
“1946-06-14”)]
```

```
Triple-2: [(subject: “Donald Trump”), (property: “president”), (object:
“United States of America”)]
```

```
Sentences: Donald Trump was born on 14 June 1946 is the president of
United States of America
```

B. Pre-processing Step:

1. Determining the type of entity
   Subject and object of each triples will be mapped into its type. For example:
   - Donald Trump: PERSON
   - 1946-06-14: DATE
   - United States of America: PLACE

   This information is gathered from DBPedia. We use special treatment for detecting number and date using regular expression.

   For unknown entity, we use UNKOWN type.

2. Creating training data:
   The deep learning model used for this task is encoder-decoder architecture that usually used in machine translation, so the input and output is a pair of sequence (source sequence and target sequence)
   a. SOURCE SEQUENCE:
      Source sequence is generated from the triples. Each entities (subject and object) is encoded into “ENTITY-[NUMBER]”. For example on the triples above, we encoded the subject and object as follows:
      - Donald Trump: ENTITY-1
      - 1946-06-14: ENTITY-2
      - United States of America: ENTITY-3

      The type of each entity is appended to the sequence, while the cammelCase value of property is splited and appended to the sequence to add the information for the Deep Learning model. We concatenate all triples into one sequence, from the example above, the source sequence will be:

      ```
      ENTITY-1 PERSON birth date ENTITY-2 DATE ENTITY-1 president
      ENTITY-3 PLACE
      ```

a. TARGET SEQUENCE:
The target sequence is de-lexicalized by each entity (subject and object of triple). The de-lexicalization process is done automatically by finding the most similar entity on n-gram sequence of the sentence. This is the most critical part because better de-lexicalization process give cleaner target sequence since the entity is not written in exact match on the target sentence. For example, if we will not find entity "1946-06-14" on the target sentence.
After applying our de-lexicalization procedure, we will get the target sentence:

```
ENTITY-1 was born on ENTITY-2 is the president of ENTITY-3
```

3. Word-vector of vocabulary
The last part of pre-processing is creating the word-vector for the vocabulary of the training data. We use pre-trained glove word vector (http://nlp.stanford.edu/data/glove.6B.zip). We are not using all glove entry, but only using word vector that exist in the training data vocabulary.

C. Deep learning
Deep learning model specification:

- Encoder-decoder architecture with attention
- RNN type: Bidirectional LSTM with 512 hidden unit
- Dropout: 0.5
- Embedding size: 300

D. Re-lexicalization
The re-lexicalization process is simply done by replacing the encoding "ENTITY-[NUMBER]" with the related entity.