Paraphrasing

Rewrite sentences from author1 to author2 writing style

Topics In Information Retrieval

Team Members:

Deeksha Singh Thakur & Utsav Chokshi
MTech CSE, IIIT Hyderabad

Overview

Paraphrases are textual expressions that convey the same meaning using different surface forms. Capturing the variability of language, they play an important role in many natural language applications including question answering, machine translation, and multi-document summarization.

Our task is to convert the text written in Author1 writing style to a paraphrase adapted from some other author's writing. By training on Author1 text we have a model that captured Author1 writing style. Used Author2 text as test data to predict paraphrases.

Implementation

I. DATA COLLECTION

Data for paraphrasing task consists of books written by two authors, having their own different style of writing, different set of vocabulary and different genres. The two authors J K Rowling and Jane Austen books are used to train and test our paraphrasing models. Jane Austen and JK Rowling are referred to as JA and JK respectively in the further report.

Trained model on text of JKRowling and Jane Austen books text. One of the author's text used as training dataset and the others text is test dataset for our paraphrasing task.

II. DATA PREPROCESSING

- Collecting book of each author, available in pdf format. Converting them to text.
- Invalid symbols in text are transformed to valid utf-8 symbols
- Gensim word-2-vec models are used to find word embedding generated for each word in text.
- POS tagging of sentences to recognize NNPs, to allow us to treat all proper nouns in same way in our model. In our dataset, all proper nouns are intended to be replaced by word "TOKEN".
- prepared word vectors for each word in corpus of rowling+austen text, using gensim libraries for word2vec, each word is a represented using 100 dimension word embedding.

III. ALGORITHMS

Since our task is of unsupervised sequence to sequence learning, LSTM encoder-decoder model comes out to be best choice. The model consists of two LSTMs – the encoder LSTM and the decoder LSTM. The input to the model is a sequence of vectors (word embeddings). The encoder LSTM reads in this sequence. After the last input has been read, the decoder LSTM takes over and outputs a prediction for the target sequence. The target sequence is same as the input sequence.

Following are the approaches we experimented for our problem statement :

1. **Single Autoencoder for seq-2-seq mapping**

Train LSTM autoencoder on austen's text data. The model is trained on Jane Austen's text. The sentences of author are converted into sentence vectors and back to original sentence using autoencoder model.

Insight into the approach :

The word embeddings used for the task of training our seq-2-seq model are generated on the total data corpus, i.e. words of both Author's text combined. Both authors in consideration write in english, the words in there text belong to the same vector space. Our approach proceeds taking this as basis of our experiment. Since word embeddings used for training task are commonly built from vocabulary of both authors, there is a high probability that the sequence encoding space of the two authors also aligns.

Implementation details :

A sequence or sentence in our task is considered to be of length 10 words/sentence. For training the model, the training is done for sequences of length 10 words per input sentence. Thus the input dimension as well as output dimension for auto_encoder is (10,100).

Model trained using JK Rowling text. Input sentence of Jane Austen given as input for testing.

Output:

Autoencoder was built by trying various variations in types of input, format of input, loss function, number of hidden layers, prediction models for testing.

-> Attempt1:

Autoencoder with **categorical_crossentropy** loss function. With 10 hidden layer. Observation : After 50 epochs of training,

Seed:

- " can . but these , i suppose , are precisely "
- " really really really really really you you revenge revenge hunt "

Seed:

- " . i do not advise the custard . mrs. goddard "
- " you really really you information information information goals goals d.a "

Seed:

- " herself that he thought of her dancing , but if "
- " thrown tent tent tent he he he he he it "

Observation & Result of Experiment : Model not trained properly.

-> Attempt2:

Autoencoder with **cosine_proximity** loss function, and 100 hidden layers, tried for 63 epochs, the observations were quite convincing,

Seed:

- " the carriage was sent for them now . "
- " the he was sent visit him , . "

Seed:

- " from the idea it suggested of something more "
- " hit the memory it hid surrounding sense motive "

Result & Observation : Improved results. The model has started to capture semantics of sentence using some sort of word matching. Eg. ‘sent for them’ was translated to ‘visit him’.

-> Attempt 3:

Increase the number of hidden layers=300 and no. of epochs of training=120 [error stabilized after 120 epochs, so terminated training]

Seed :

- " i should probably be off in five minutes . at "
- " i should really return exams within twelve february . anticipating "

Seed:

- " the while . `` how good it was in you "
- " the chaise . `` really that it was required you "

Seed:

- " `` to yield without conviction is no compliment to "
- " `` to censure immediate nature is fair to to "

Observation & Result : The model trained using this approach is finding phrasal similarities, and syntactic similarities. The semantic space of input and output sentences still could not be aligned.

2. **Two Autoencoders - plugging one's encoder with other's decoder**

Trained two autoencoders parallelly. One trained on JA text and the other on JK text. The weights of both models were saved. And a new model was built using the encoder of JA and decoder of JK.

Insight into the approach :

Each author has his own set of vocabulary. Hence aligning of 'thought vectors' is more important than aligning word meanings. We experimented keeping in mind that thought vectors of both authors may fall into same vector space as both authors write in english, and word embedding of input are trained by taking text of both authors together.

Implementation Details :

The training autoencoder model was divided into distinguished layers of-

1. Input Sentence (100 dimension word embedding * 15 words)
2. Encoder(Input) -> Thought vector
3. Decoder (Thought vector) -> Output Sentence = Input Sentence(100dimension word embedding * 15 words)

Then a final model was prepared for testing with the following layers -

1. Input 1 - [Input from text of author1]
2. Encoder 1 - [Encoder for text of author 1]
3. Decoder 2 - [Decoder for text of author 2]

4. Output 2 - [Output generated by decoder of author 2]

Output :

Test

- Input : SENT_START to elizabeth , however , he voluntarily acknowledged that the necessity of his absence had been self-imposed . SENT_END
- Output : ascertaining done done done done done done done scared scared scared scared scared scared scared thinking thinking thinking thinking thinking "

Test

- Input : SENT_START but now i have no doubt of seeing him here about the second week in january . '' SENT_END
- Output : -impossible "

Result & Observation :

The model did not work as intended. The plugging in of two different encoders could not align and gave garbage results even after 100 epochs of training each model.

3. Skip Thought vector

The core problem in previous approach was :

1) Thought Vectors were not aligned i.e. thought vectors of JK and JA may have same set of concepts but they may represent same concept in different dimension.

2) JK and JA has different genres of writing. Some concepts which are present in JK strongly (like magic) may not be present in JA at all !

Insight into the approach :

1) Represent sentence using skip vector.
    a) Skip Thought Vector is generalized sentence embedding prepared by training sentence encoder on different genres of books.
    b) Skip Thought Vectors are prepared just like skip-gram model of word2vec. An encoder- decoder model is trained that tries to reconstruct the surrounding sentences of an encoded passage.
    c) https://arxiv.org/pdf/1506.06726.pdf

2) Training decoders : Give skip vector of JA sentence to JA decoder and skip vector for JK sentence to JK decoder.
3) Testing : represent JA sentence using skip vector and input it to JK decoder. Output sentence from JK decoder is paraphrased sentence.

Implementation Details :

1) We used pre -trained model for skip-thought vector to generate skip thought vectors for sentence. (https://github.com/ryankiros/skip-thoughts)
2) We trained decoder model on JA text.
3) We tested decoder model on JA by giving input sentences from JK text.

Output :

Input JK Sentence -> Ouput JA Sentence

```
Input Sentence :
what could he have been thinking of ?
Output Sentences :
['what could he have been in thinking of ?']
------
Input Sentence :
it must have been a trick of the light .
Output Sentences :
['it must have been a very simple of the case .']
------
Input Sentence :
mr. dursley blinked and stared at the cat .
Output Sentences :
['mr. knightley looked up at his door .']
------
Input Sentence :
it stared back .
Output Sentences :
['it regarded her .']
------
```

Input JK Sentence -> Ouput JK Sentence

```
Saving... Done
Truth  0 :  hagrid and harry made for the counter .
Sample ( 0 )  0 :  hagrid and harry .
Truth  1 :  the oldest boy came striding into sight .
Sample ( 0 )  1 :  the giant chuckled and stared at the cat .
Truth  2 :  harry followed hagrid out onto the rock .
Sample ( 0 )  2 :  harry stared at the kitchen .
Truth  3 :  harry felt a great leap of excitement .
Sample ( 0 )  3 :  harry felt a start leap of excitement .
Truth  4 :  mr. dursley sat frozen in his armchair .
Sample ( 0 )  4 :  mr. dursley stood rooted to the spot .
Truth  5 :  he had a bored , drawling voice .
Sample ( 0 )  5 :  he had a bored , grubby-looking pub .
Truth  6 :  harry did n't trust himself to speak .
Sample ( 0 )  6 :  harry did n't move .
Truth  7 :  `` the great humberto 's on tonight .
Sample ( 0 )  7 :  `` the 's name 's name again .
Truth  8 :  he 'd never even seen the boy .
Sample ( 0 )  8 :  he had n't remember him .
Truth  9 :  those silvery eyes were a bit creepy .
Sample ( 0 )  9 :  a single wand was a very odd watch .
Epoch  3 Update  337 Cost  95.659072876 UD  16.4188809395
Saving... Done
```

Result & Observation :

- This model works excellently when it has to decode same author's text.
- Compared to previous models it captures syntactic similarity very well.
- For small sentences, it captures semantic similarity but fails for large sentences.

Conclusion

Our approaches could only reach as far as phrasal similarity and aligning sequences on basis of word meanings and syntax.

They failed to capture semantic similarities between sequences effectively.