Use of Recurrent Neural Network for grammatical error detection

Piotr Knapczyk, Mateusz Nowak, Michał Pluta, Włodzimierz Funika

AGH, Faculty of Computer Science, Electronics and Telecommunication,
Department of Computer Science

Motivation

- need for a solution to detecting article-bound errors in texts in English
- self-trained - based on a given dataset
- collecting statistics
- providing the user with a simple graphical interface
- use Recurrent Neural Network for solution implementation

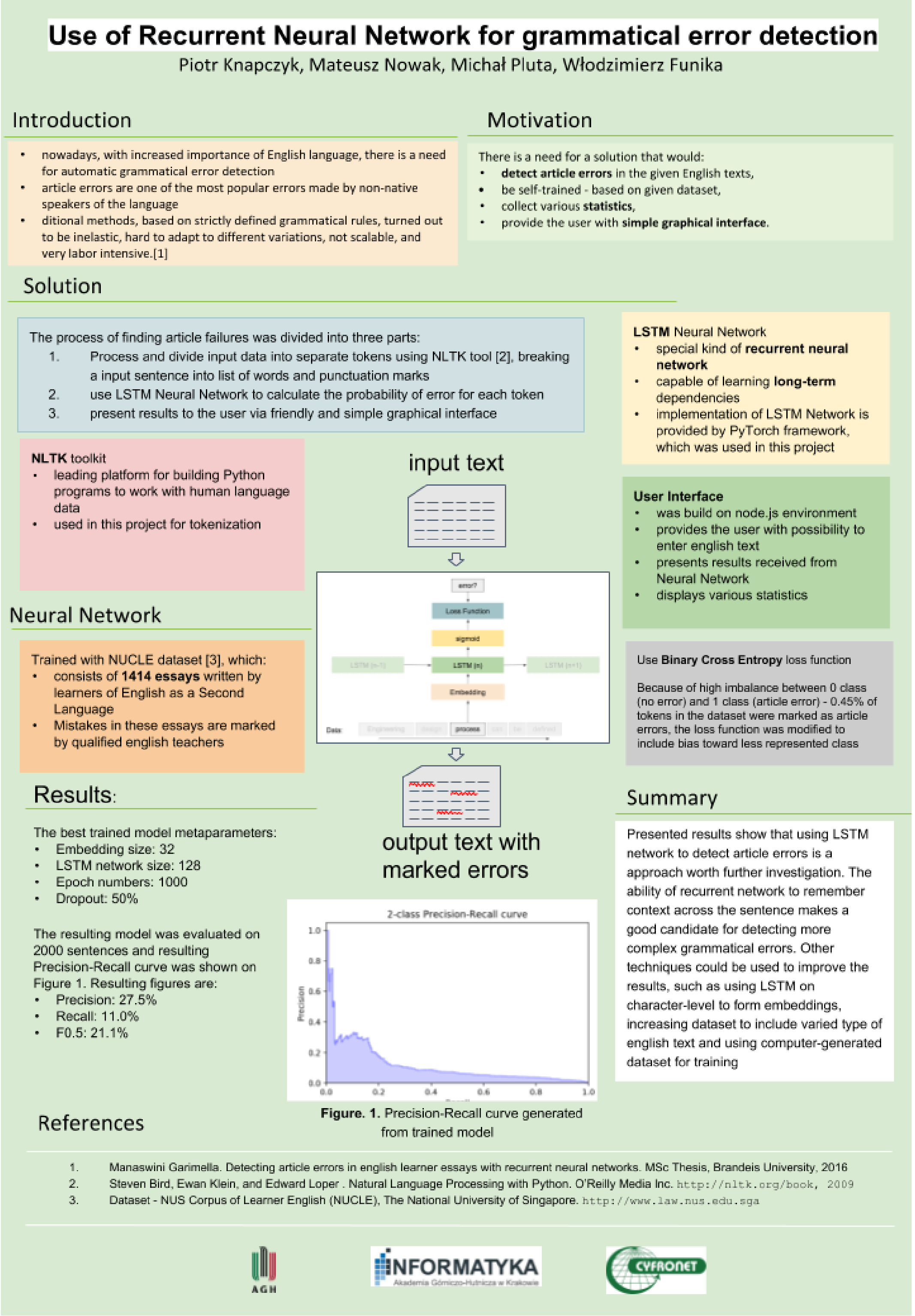
Solution

The process of finding article-bound failures was divided into three parts:

- process and split the input data into separate tokens using the NLTK tool, breaking an input sentence into a list of words and punctuation marks
- use LSTM Neural Network to calculate the probability of error for each token
- present results via a user-friendly and simple graphical interface

Poster No 2

We invite you to get acquainted with our poster which contains a detailed description of our research and proposed solution.



error?
Loss Function
sigmoid
LSTM (n-1)
LSTM (n)
LSTM (n+1)
Embedding
Data:
Engineering
design
process
can
be
defined
2-class Precision-Recall curve
Precision
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Use of Recurrent Neural Network for grammatical error detection

Piotr Knapczyk, Mateusz Nowak, Michał Pluta, Włodzimierz Funika

Introduction

- nowadays, with increased importance of English language, there is a need for automatic grammatical error detection
- article errors are one of the most popular errors made by non-native speakers of the language
- ditional methods, based on strictly defined grammatical rules, turned out to be inelastic, hard to adapt to different variations, not scalable, and very labor intensive.[1]

Motivation

There is a need for a solution that would:

- **detect article errors** in the given English texts,
- be self-trained - based on given dataset,
- collect various **statistics**,
- provide the user with **simple graphical interface**.

Solution

The process of finding article failures was divided into three parts:

1. Process and divide input data into separate tokens using NLTK tool [2], breaking a input sentence into list of words and punctuation marks
2. use LSTM Neural Network to calculate the probability of error for each token
3. present results to the user via friendly and simple graphical interface

NLTK toolkit

- leading platform for building Python programs to work with human language data
- used in this project for tokenization

Neural Network

Trained with NUCLE dataset [3], which:

- consists of **1414 essays** written by learners of English as a Second Language
- Mistakes in these essays are marked by qualified english teachers

Results:

The best trained model metaparameters:

- Embedding size: 32
- LSTM network size: 128
- Epoch numbers: 1000
- Dropout: 50%

The resulting model was evaluated on 2000 sentences and resulting Precision-Recall curve was shown on Figure 1. Resulting figures are:

- Precision: 27.5%
- Recall: 11.0%
- F0.5: 21.1%

input text

output text with marked errors

LSTM Neural Network

- special kind of **recurrent neural network**
- capable of learning **long-term** dependencies
- implementation of LSTM Network is provided by PyTorch framework, which was used in this project

User Interface

- was build on node.js environment
- provides the user with possibility to enter english text
- presents results received from Neural Network
- displays various statistics

Use **Binary Cross Entropy** loss function

Because of high imbalance between 0 class (no error) and 1 class (article error) - 0.45% of tokens in the dataset were marked as article errors, the loss function was modified to include bias toward less represented class

Summary

Presented results show that using LSTM network to detect article errors is a approach worth further investigation. The ability of recurrent network to remember context across the sentence makes a good candidate for detecting more complex grammatical errors. Other techniques could be used to improve the results, such as using LSTM on character-level to form embeddings, increasing dataset to include varied type of english text and using computer-generated dataset for training

Figure. 1. Precision-Recall curve generated from trained model

References

1. Manaswini Garimella. Detecting article errors in english learner essays with recurrent neural networks. MSc Thesis, Brandeis University, 2016
2. Steven Bird, Ewan Klein, and Edward Loper . Natural Language Processing with Python. O'Reilly Media Inc. `http://nltk.org/book, 2009`
3. Dataset - NUS Corpus of Learner English (NUCLE), The National University of Singapore. `http://www.law.nus.edu.sga`