# Indonesian Essay Scoring using Bi-LSTM with Word Embedding Representation

**Ilham Firdausi Putra**
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
ilhamfputra31@gmail.com

## Abstract

This paper presents the solution that placed 2nd at UKARA 1.0 Challenge 2019. UKARA 1.0 Challenge is an Indonesian automatic essay/short-answer scoring competition held by Universitas Gadjah Mada. We combine Bi-LSTM with pretrained word embedding vector to achieve F1-score of 0.81. The code and pretrained Word2vec word embedding will be made publicly available[1].

## 1 Introduction

Automatic essay scoring as one of the topics in natural language processing has been greatly developed by the demand to make the assessment process faster. Despite the fast advancement in automatic essay scoring, research in this area for Bahasa Indonesia has been very limited and only recently emerged as a topic. The use of informal language and the diversity of local languages was the main challenge in developing automatic essay scoring for Bahasa Indonesia.

UKARA 1.0 Challenge aims to encourage more ideas and studies for developing automatic short-answer scoring specifically for Bahasa Indonesia. In this challenge, participants will be given access to datasets in the form of students short-answers in two phase. In the first phase, the participant developed their solution with the development set for 43 days from July 29 - September 10, 2019. Finally, the participant can submit their final solution on the second phase with the test set for 3 days from September 16 - September 19, 2019.

## 2 Indonesian Essay Scoring

In this solution, we cast the challenge as a binary classification problem. Given a short-answer to the stimulus, we build a model that tries to predict whether the answer was relevant to the stimulus or not. We process the inputs as a sequence of word. Each word represented as a low dimensional vector and processed sequentially by bidirectional LSTM (Hochreiter and Schmidhuber, 1997).

| Type of Set | Data A | Data B |
|---|---|---|
| Training | 268 | 305 |
| Development | 215 | 244 |
| Test | 855 | 974 |

Table 1: The total of short-answer for each set type and data type.

### 2.1 Dataset

The dataset is a short-answer from 2 different stimuli (For the size detail, see Table 1). The short-answer and stimulus consist of a total 36,930 word with 2,816 unique vocabularies. The label for each short-answer was a binary with 1 representing relevant answer and 0 representing non-relevant answer. The only text preprocessing done was converting character to lowercase and removing non-alphanumeric character.

### 2.2 Word Embedding

We pretrained Word2vec (Mikolov et al., 2013) 100 dimension word embedding using Gensim (Řehůřek and Sojka, 2010) on Indonesian text from Wikipedia dump[2], Opensubs (Lison and Tiedemann, 2016), and the preprocessed UKARA dataset (For the word count detail, see Table 2). The addition of text from Opensubs and UKARA dataset helps in providing informal words that usually absent in Wikipedia article. With this dataset, we ended up with a total of 420,024 unique vocabularies.

[1] https://github.com/ilhamfp/ukara-1.0-challenge

[2] https://dumps.wikimedia.org

| Data Source | Word Count |
|---|---|
| Opensubs | 105348108 |
| Wikipedia | 101251643 |
| Ukara | 36930 |
| Total | 206636681 |

Table 2: The count of word for each data source.

| Stage | Known Word Count |
|---|---|
| 1: Raw Word | 2310 |
| 2: Stemmed | 65 |
| 3: Normalized | 48 |
| 4: Stemmed | 3 |

Table 3: The count of known word found in each stage of building the word embedding layer.

### 2.3 Model

We use Keras (Chollet et al., 2015) with Tensorflow (Abadi et al., 2015) as the backend to build the model. The text was tokenized and padded into maximum length of 43 (90th percentile of all short-answer length) before goes into the model. In order to build the embedding layer, we perform a multi-stage text processing using PySastrawi[3] stemmer and a normalizer function (removing duplicate adjacent character) to minimize the amount of unknown vocabulary (For the count of known word found in each stage, see Table 3). This multi-stage process yields a total of 2.426 known vocabularies and 390 unknown vocabularies. We finally fit the model with an EarlyStopping and ReduceLROnPlateau callback.

### 2.4 Experiment

We run the experiment on RepeatedStratifiedKFold with 10 split and 10 repeats. For each split and repeat, we perform prediction to the validation and test set. We later normalize the result according to how many predictions made, essentially performing ensemble of 100 different model fit. We choose a threshold of 0.5 for Data A and 0.48 for Data B in predicting the label. See Table 4 for the result.

[3] `https://github.com/har07/PySastrawi`

| Type of Data | F1 Score | Precision | Recall |
|---|---|---|---|
| Data A CV | 0.89277 | 0.85238 | 0.93717 |
| Data B CV | 0.77083 | 0.68519 | 0.88095 |
| Data Test | 0.81 | 0.75 | 0.89 |

Table 4: The cross-validation and final test result.

## 3 Conclusion

In this work, we present the effectiveness of BiLSTM and pretrained Word2vec in Indonesian essay scoring problem. In addition to that, the inclusion of Opensubs data helps in providing informal words that usually absent in Wikipedia article. Finally, we maximize the amount of known word in building word embedding layer by performing multi-stage text processing.

## References


Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

François Chollet et al. 2015. Keras. `https://keras.io`.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

P. Lison and J. Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.


## A Hyperparameter Detail

### A.1 Gensim Hyperparameter

We use `gensim.models.word2vec.Word2Vec` default parameter as of version 3.8.1.

### A.2 Model Hyperparameter

We use the default parameter as of Keras version 2.3.0 and Tensorflow version 1.14.0 as the backend with the exception of the following:

**Bi-LSTM**:

- `units`: 50
- `return_sequences`: True
- `return_dropout`: 0.1
- `return_recurrent_dropout`: 0.1

**Dropout**:

- `rate`: 0.1

**EarlyStopping**:

- `monitor`: 'val_f1'
- `min_delta`: 0.0001
- `patience`: 8
- `mode`: 'max'
- `baseline`: None
- `restore_best_weights`: True

**ReduceLROnPlateau**:

- `monitor`: 'val_f1'
- `factor`: 0.5
- `patience`: 3
- `mode`: 'max'
- `min_lr`: 1e-6