Supplementary of Numeracy-600K: Learning Numeracy for Detecting Exaggerated Information in Market Comments

1 Details of Model Setting

In our models, each word in the input sentence is represented as a d -dimensional vector with word embeddings, and all the words are concatenated in as a $d \times l$ matrix, where l denotes the sentence length. Some preprocessing was performed on the data. We transformed all characters to lowercase. The sentence representation was padded to the maximum length of an instance. The target numeral to be inferred is replaced with a special token <TRT>.

1.1 Convolutional Neural Network (CNN)

We construct a CNN model for numeracy. Modified from the CNN for sentence classification (**?**), in our model, each word in the input sentence is represented as a $d-$dimensional vector, and all the words are concatenated in as a $d \times l$ matrix, where l denotes the sentence length. The target numeral to be inferred is replaced with a special token <TRT>. The output of our CNN model is a softmax layer that generates the probability distribution over the magnitudes for the target numeral.

The details of our CNN model are described as follows. The size of the first layer, the embedding layer, is set as d = 300. We set l = 73, which is the longest sentence in the dataset. Padding is performed for shorter sentences. The second layer is a convolutional layer with filter size 8. The third layer is a fully connected layer with dimension 32, which functions as a max-pooling layer. To avoid overfitting, a dropout layer is added with a dropout rate of 0.3. Finally, two activation functions, the rectified linear unit (ReLU) and softmax, are used in the last two layers. We chose to use the Adam optimizer.

1.2 Gated Recurrent Unit (GRU)

We construct an RNN-based model for numeracy with GRU. The tokens in the sentence are input as a sequence. Each token is represented as a d-dimensional vector. The target numeral is replaced with the special token <TRT>. The architecture of the GRU model in this paper consists of a 300-dimensional embedding layer, a 64-dimensional GRU layer, and a dropout layer with a dropout rate of 0.3. The final two layers and the optimizer are the same as those in the CNN model.

1.3 Bidirectional GRU (BiGRU)

The bidirectional RNN model, BiGRU, merges the outputs from both directions of the GRU model. Because units of measurement provide the important clues for numeral, a bidirectional architecture is expected to be useful with the right to left inputs. For example, the difference between (C1) and (C2) is the unit of measurement (i.e., *POINTS* and *PERCENT*), and it leads to different results of the magnitude of numerals.

> **(C1)** *DOW JONES <.DJI> UP 8.70 POINTS*
>
> **(C2)** *DOW JONES <.DJI> UP 0.05 PERCENT*

1.4 Convolutional Recurrent Neural Network (CRNN)

In our CRNN model, a CNN layer extracts features for each segment. Then, a max-pooling layer in the CNN model is replaced by an RNN layer and aggregates the extracted features. To examine whether replacing the pooling layer with the RNN layer can improve performance in our task, we keep the other components of the CRNN model the same as those in the CNN model, and replace the max-pooling layer with the 64-dimension BiGRU layer.

1.5 CNN-capsule

We also introduce one of the latest architectures, capsule network, to the task of numeracy. We combine the capsule network with either of the CNN and the GRU models. The structure of the CNN-capsule model begins with a 300-dimensional embedding layer. The second layer is a convolutional layer having a kernel size of 9 and using the ReLU activation function. The third layer, called the primary layer, is used to retain the order of context information, including one convolutional layer with 32 channels. Finally, the capsule layer outputs an $n \times dim$ matrix, where n is the number of classes, set to 8 for this paper, and dim is the dimension of each capsule, set to 16.

1.6 GRU-capsule

The GRU-capsule model begins with a 300-dimensional embedding layer, followed by a 64-dimensional GRU layer, which returns the full sequence of outputs. To compare the impacts of the CNN and RNN frameworks in the CapsNet architecture, we keep the primary and capsule layers the same as those in the CNN-capsule model.

1.7 BiGRU-capsule

We further explore the bidirectional GRU model with the addition of capsule network. The BiGRU-capsule model consists of a 300-dimensional embedding layer, bidirectional GRU layers with a 64-dimensional hidden state, and the primary and capsule layers described above.