Paper 11 | Finki at SemEval-2016 Task 4: Deep Learning Architecture for Twitter Sentiment Analysis by Dario Stojanovski et. al – published in 2016

Abstract

1. Deep learning methods for sentiment analysis handle the feature extraction automatically which provides for robustness and adaptability
2. The proposed a fusion model named Finki, which employs both CNN and gated RNN (GRNN) to obtain a more diverse tweet representation
3. The network trained on top of GloVe word embeddings pre-trained on the Common Crawl dataset
4. Both NN are used to obtain a fixed length representation of variable sized tweets and the concatenation of these vectors is supplied to a fully connected softmax layer with dropout regularisation
5. The model achieved the best and second highest results on the 2-point and 50point quantification subtask respectively (without relying on any hand-crafted features)

Models

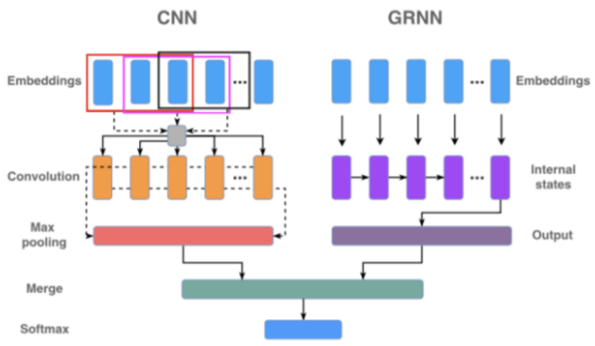

Figure 1: Deep neural network architecture.

- The CNN has a single filter with window size of 3
- The model is implemented using Keras on a Theano backend

1. Preprocessing
   - All URLs and HTML entities are removed
   - Removed punctuation except question and exclamation marks
   - Remove user mentions
   - Emoticons and Twitter specifics are kept
   - Lowercase all words
   - Each appearance of an elongated word is shortened to a maximum of three-character repetitions
2. Pre-trained word embeddings
   - For words in the dataset not present in the lookup table, the authors use random initialisation of word embeddings
   - This is effective in encoding syntactic and semantic regularities of words but they are still oblivious to the words' sentiment characteristics
   - Therefore, the authors allow the word embeddings to continuously update during network training

3. CNN
   - Only employed one convolutional and max pooling layer
   - The convolutional layer is used to extract local features around each word window while the max pooling layer is used to extract the most important features in the feature map
   - A tweet is represented as a concatenation of the word embeddings of the words within the tweet
4. Gated RNN
   - RNNs make use of sequential data. They perform the same task for every element in a sequence with the output being dependent on previous computations
   - RNNs suffer from exploding and vanishing gradient problem. Two proposed methods:
     - LSTM networks
     - Gated Recurrent Unit (GRU)
   - The authors decided to use GRU because of the fewer model parameters, therefore, potentially needing less data to generalise and enabling faster training
   - GRU has gating units that modulate the flow of information inside the unit
     - The current activation of the GRU at time t is a linear interpolation between the previous activation at t-1 and the candidate activation:

       $$s_t^j = (1 - z_t^j)s_{t-1}^j + z_t^j \hat{s}_t^j,$$

       where an update gate decides how much the unit updates its activation or content. The update gate is computed as:

       $$z_t^j = \sigma(W_z x_t + U_z s_{t-1})^j.$$

       The candidate activation is computed as:

       $$\hat{s}_t^j = tanh(W x_t + U(r_t \odot s_{t-1}))^j,$$

       Where r_t is a set of reset gates and reset gate is computed as:

       $$r_t^j = \sigma(W_r x_t + U_r s_{t-1})^j.$$
5. Network fusion
   - The outputs from both networks are concatenated to form a single feature vector and feed into a fully connected softmax layer. The softmax regression classifier gives probability distribution over the labels in the output space
6. Regularisation and model parameters
   - Dropout to counter overfitting issue. The dropout is set to 0.25
   - The output size of CNN and GRU network is set to 100 and the network is trained using SGD over shuffled mini-batches using RMSprop update rule

Results

Measure	Baseline	Score	Rank
Acc	0.778	0.848	4
AvgF1	0.438	0.748	7
AvgR	0.5	0.72	10
MAE^{μ}	0.537	0.672	6
$\mathbf{MAE^{M}}$	1.2	0.869	5
AE	0.184	0.074	1
RAE	2.11	0.707	3
KLD	0.175	0.034	1
EMD	0.474	0.316	2
AvgRank			4.5

Table 3: Results and ranks for Subtask B, C, D and E respectively

1. The systems are ranked by the macroaveraged recall for the Subtask B where higher scores are better
2. Other subtasks, the systems are ranked by the error functions
3. The proposed model performs best on the quantification subtasks and manages second highest average rank on the considered subtasks

Conclusion

1. For future work, the authors would like to pre-train word embeddings on a large set of distantly labelled tweets
2. It would be interesting to see the effects of using bi-directional GRNN