**CS 482/682 Final Project Midterm Report**

Cong Gao, Zhaohao Fu, Liujiang Yan

1. **Problem Statement**

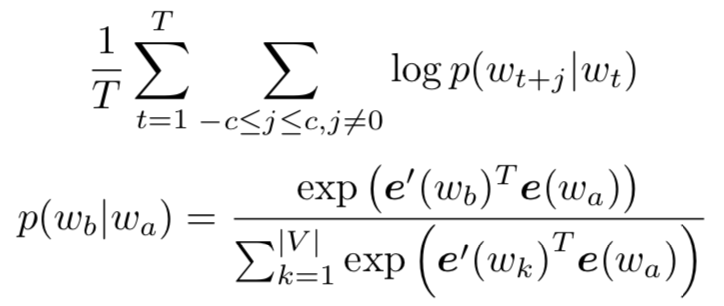
The task we aim to solve here is text categorization, which is to classify the type or source given the text. For example, given the tweets, let the system to classify the person who published them; or given the news title or abstract, let the system to classify the types of the news. It is essentially a supervised learning problem and classification problem.

In this project, the language will be restricted to English, while the approach should be easily expanded to other languages with little effort. We compare the performance as well as efficiency from several deep learning based approaches. The dataset we start with comes from the instruction guide: (1) Tweets from Hilary Clinton and Donald Trump; (2) UCI dataset of news from multiple topics.

1. **Technical Approach**
2. Word-level RNN

**Word Embedding** Different from computer vision tasks, which come with numerical representation and are straightforward in numerical operation defined by the networks. For natural langrage processing tasks, the input are string of characters with varying length, which is not trivial for processing straightforward. Therefore, we need a representation for given text that we could further perform numerical operation, leading to the following two different representations and their corresponding network architectures.

The first method we are using here is inspired by Lai et al[1]. The representation here uses word embedding that represents a given word by a vector of weights from a pretrained model that stands for the semantics information. In our project we do not train our own model from ground, but use the present Skip-tram pretrained model, which is the state-of-art in many NLP tasks[1]. The Skip-gram model trains the embeddings of words by maximizing the average log probability,



**Architecture** Figure 1. Shows the architecture of the Word-level RNN, which consists of several components. The first one is a bidirectional recurrent network, where the sequence input is the sequence of word embeddings given by the text, and the hidden variables stand for left and right contextual information correspondingly. Then we stack the corresponding left and right contextual hidden variables with the input and form a feature vector. Then a fully connected layer connect the feature vector to a hidden layer and a max pooling layer is performed to capture the max responses at each position. The output is same size with the classes’ number and a log softmax activation function is performed to get the final output. Since it is a classification problem and the final layer is log softmax, we use negative log likelihood as our loss function.

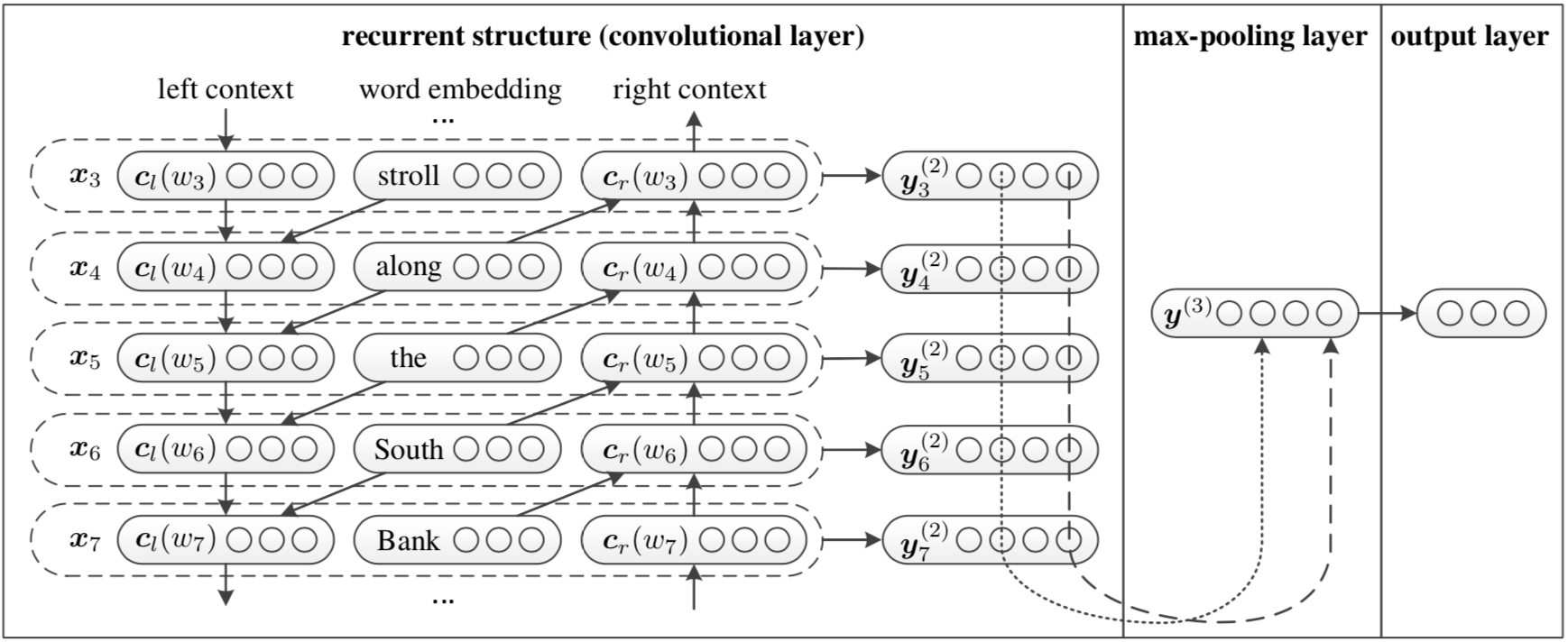


Fig 1. Architecture of the Word-level RNN

1. Character-level RNN

**Character quantization** The second method we used here is designed from Zhang et al[2]. The representation here is character level. For each given text, it forms a fixed size binary matrix that each column is one hot vector size 1-of- with one for the character’s index in a given alphabet and zero otherwise; and the total row number is fixed to , any character exceeding length is ignored. Thus, the sequence of characters is transformed to a sequence of such sized vectors with fixed length .

**Architecture** Figure. 2 presents the architecture of Character-level RNN, which is similar to Alexnet used for image classification, consisting of convolutional layers, max pooling layers and fully connected layers. As discussed above, the final layer is a log softmax for multiple class classification task, and we use negative log likelihood as our loss function.

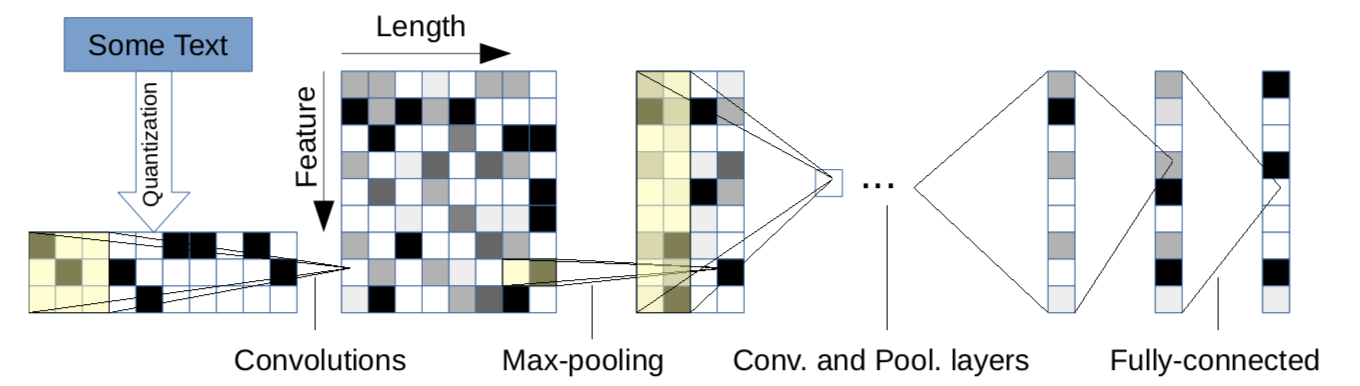


Fig 2. Architecture of the Character-level RNN

1. **Implementation Status**

We have implemented the pipeline for text categorization that consists of

(1) Datasets for character level and word level, including tweets text by Donald Trump and Hilary Clinton;

(2) Configuration loading component;

(3) Model architecture definition for word-level RNN and character-level RNN;

(4) Training and evaluation framework. Specifically, for word embedding we use pretrained model from GloVe by Stanford University. The whole pipeline is implemented in PyTorch, and version control is done by using GitHub private repo.

1. **Future Works**
   1. Personal Dataset

Besides the dataset we have now, we propose to collect more data. Specifically, we will collect the tweets from Hilary and Trump up to date as extension to our current tweets dataset. We may also utilize some existed public dataset like AG news.

* 1. Architecture
     1. Multiple RNN structures

The recurrent part of the architecture could be vanilla RNN or LSTM, and we will compare the performance of these two variants. The outputs of RNN connect with hidden layer by shallow fully connected layer, it is interesting to know whether deeper connections will lead to better accuracy. It would also worth a try to replace the fully connected layer with convolutional layers.

* + 1. Improvement of Character-level RNN

We shall see that the architecture proposed in paper is similar as AlexNet for image classification task. Inspired by current popular architecture for vision tasks, we would like to vary the architecture by:

* Use smaller kernels;
* Use residual layers;
* Remove fully connected layers.

We will compare the performance as well as the efficiency of each variant.

* 1. Hyper parameter tuning

1. **Reference**

[1] Lai, S., Xu, L., Liu, K. and Zhao, J., 2015, January. Recurrent Convolutional Neural Networks for Text Classification. In AAAI (Vol. 333, pp. 2267-2273).

[2] Zhang, X., Zhao, J. and LeCun, Y., 2015. Character-level convolutional networks for text classification. In Advances in neural information processing systems (pp. 649-657).