Sai Teja Karnati
UIN: 659 365 999

Review of the paper:

Character-Aware neural language models- Yoon Kim, Yacine, David, Rush

Summary:

The paper employs a CNN and a Highway network over characters whose output is given to the LSTM RNN model instead of the word embedding given at input layer. This helps in obtaining a better neural language model which relies on character-level input.

Neural Language Models (NLM) have outperformed count-based n-gram language models and also solved the problem of sparsity issue through parametrization of words as vectors. But they are blind to subword information (e.g. morphemes).This leads to high perplexities for rare words which is very problematic in morphologically rich languages. The model described in the paper understands the subword information through Character-level Convolutional Neural Network (CNN) and uses its output as input for the RNN-LM. The model doesn't need any morpheme tagging or word embeddings. The model performs on par with state of the art results with 60% lesser parameters.

Architecture of the model involves a RNN (LSTM RNN), Character level CNN and a Highway Network. A typical LSTM RNN takes input embedding X^K at time *t* and outputs a probability distribution of output embeddings. Training involves minimizing the negative log-likelihood by truncated backpropagation through time. In the model, X^K input embeddings are replaced by the output from the CNN. Character level CNN (CharCNN) has input matrix C^K on which a filter/kernel H of width w is applied which gives us output Y^K. This output Y^K goes through a Highway network (allows training of deep networks by adaptively carrying some dimensions of the input directly to the output). This output Z^K is used instead of X^K word embedding in LSTM RNN input layer.

Performance of the model is evaluated using Perplexity, PPL = $e^{NLL/T}$,where NLL is calculated over the test set. Hyperparameters are trained on Penn Treebank (PTB) and applied on various morphologically rich languages. The preprocessed data is obtained from the authors and their baselines are used as comparisons. Training is done on a small and big dataset (Data-S, Data-L) and the <UNK> token is used to replace OOV.

Optimization is done by truncated backpropagation through time for 35 times using SGD with learning of 1.0 which halves if perplexity doesn't change by more than 1. Batch size of 20 was used on Data-S and 100 was used on Data-L. Parameters of the model were initialized randomly. Training on Data-L is sped up using hierarchical Softmax. LSTM-Char-Small uses 200 hidden units while LSTM-Char-large uses 650.

The model established in the paper outperforms most of the State of the art models with 60% lower parameters (19m vs 52m). Even significant results can be seen in other morphologically rich languages when compared against MLBL model and its LSTM version.

Sai Teja Karnati
UIN: 659 365 999

****

Things I liked in the papers:

- ✓ Significant improvement in morphological rich languages with lesser dimensions solving the issues with resource scaling.
- ✓ In depth evaluation of model both quantitatively and qualitatively against some state-of-the-art models and outperforming them often.
- ✓ Concept of using CNN, Highway network and also using RNN together may be helpful in other areas of research as well.
- ✓ Paper neatly describes the previous methods and issues briefly.

Things that could be improved in the papers:

- ✓ The paper mentions issues with scaling and GPU resources and doesn't mention clear solutions or results in upgraded GPU cases.
- ✓ The word representations seem to still have a little bit of edit-distance based errors like (his, hhs) despite semantic feature encoding from Highway network

Future:

- ✓ Better understanding and exploration of models with mixed feature extraction techniques.
- ✓ Use of characters, extraction of morphemes and its value.
- ✓ Obtaining good results without lot of parameters using task specific models.

Major difference between this Character-aware NLM and morphological log-Bilinear model (or its LSTM version) is the reduced parameters and avoidance of morpheme embeddings.

The loss function used is SGD on the negative log-likelihood done by truncated backpropagation over time. We discussed softmax in class and then discussed SGD which solves computational problems. We also discussed Backpropagation in RNN and CNN. We haven't discussed Highway Networks in class but they are similar to LSTM in concept.

Data sets used were Penn Treebank for English, ACL Workshop on Machine Translation Data for other languages and Arabic data from News-Commentary corpus.