Character Level Convolution Network for Text Classification

Project Id: 14

Team Name: Apriori

Gayatri Madduri (2019102043)

Kawshik Manikantan (2019111004)

Srividya Srigiri (2019102041)

Sai Akarsh (2019111017)

Problem statement

Text classification is currently the emerging area in Natural language processing (NLP). Text classification is a machine learning technique which assigns a set of pre-defined tags or categories to free-text. The main objective of text classification is to categorize or organize any kind of text from files over the web. Text classification can be applied to many applications like sentiment analysis, text labelling and spam detection.

Convolutional Neural Networks(CNNs) are effective in extracting information in many areas like text recognition and classification. In our model, ConNets are applied at character level. This approach uses words as the basis in which character-level features are extracted at word level and form a distributed representation. The model accepts a sequence of encoded characters as input and then quantize that character sequence in backward order which makes it easy in case of a fully connected network.This method of classification actually works on different languages without going into the semantic details of it. Especially, this model is very efficient on large-scale datasets which is not possible in other previous approaches.

Goals and Approaches

- Replicating the CREPE model that has been proposed in the paper.
- Building the other models used for comparison that have been specified in the paper.

- Improving the proposed model using the latest developments and technologies available.
- Building an app to showcase the model we have built using StreamLit. (This is an extra objective that will be done if time permits)

- Each of us have divided the multiple models that need to be implemented for the project. The breakup of the work is given below.
- The paper used eight datasets and five models along with their model to test and compare. We will first try to use only one dataset for our models and then try to improvise them. In the end we will try to expand this to around five datasets. The timeline for this has been provided below.
- We noticed that the proposed model did not perform the best in some datasets where the bad of words approach worked better. We will try to see if we can figure out what makes that better than our model and try to incorporate that feature. Given the fact that this paper is over seven years old, there is a lot of new technologies out there that we can use to improve the model. We will try to see all of these in order to improve our model and give a better comparison and result.

Datasets

- **DBPedia:**
  - 14 non-overlapping classes from DBpedia 2014
  - 40,000 training samples
  - 5,000 testing samples

```
from datasets import load_dataset

dataset = load_dataset("dbpedia_14")
```

- **Yelpreview full:**
  - The Yelp reviews dataset consists of reviews from Yelp. It is extracted from the Yelp Dataset Challenge 2015 data.

- 130,000 training samples and 10,000 testing sample
- 5 classes

```
from datasets import load_dataset

dataset = load_dataset("yelp_review_full")
```

- **Amazon polarity:**
  - 35 million reviews classified into 2 classes
  - 1,800,000 training samples and 200,000 testing samples

```
from datasets import load_dataset

dataset = load_dataset("amazon_polarity")
```

Expected Deliverables

- Code
  - Code for the model proposed in the paper along with the other models for comparison
  - Code for the experimental models prepared by the team for the improvement of the given model or models used in the paper.
- Report
  - Report consists of the details of the experiment along with the metric comparisons for all of the experiments.
- Presentation
  - Presentation consists of intuitive explanation of the thought process used during the entire project by the team members
- App based on the model (If time permits)
  - An app that shows the working of the models designed as a part of the project.

Milestones and Deadlines

Weeks	Milestones
Week 1: (Nov 1 – Nov 7)	Learning necessary concepts and frameworks for the model to be designed.
Week 2: (Nov 7 – Nov 14)	Building the models assigned to each member on a single common dataset.
Week 3: (Nov 14 – Nov 21)	- Improving performance of the main model. - Extending to multiple datasets
Week 4: (Nov 21 – Nov 28)	- Buffer for completion of open issues. - Compiling observations into a report and a presentation. - Building the app

Work Distribution

The following work distribution has been planned out for the initial stages of the project. The effort is to understand the variety of models and probably come up with an idea that is based on the better elements of the individual models.

The final improved would be a true team effort that involves brain-storming for the better features and approaches

Partition 1: --- Gayatri Madduri

- Bag-of-words and its TFIDF
- Bag-of-ngrams and its TFIDF.
- Bag-of-means on word embedding

Partition 2: --- Srividya Srigiri

- Word-based ConvNets.

Partition 3: --- Sai Akarsh

- Long-short term memory.

Partition 4: --- Kawshik Manikantan

- Paper Implementation

References

- Xiang Zhang, Junbo Jake Zhao, Yann LeCun: Character-level Convolutional Networks for Text Classification. CoRR abs/1509.01626 (2015)
- K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 28(1):11–21, 1972
- T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Wein-berger, editors, Advances in Neural Information Processing Systems 26, pages 3111–3119. 2013
- S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Comput., 9(8):1735–1780, Nov. 1997
- A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks, 18(5):602–610, 2005.