On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis

**Research Question & Objectives:** The primary questions of the research are: (1) Are neural network architecture (especially CNN) affected by small preprocessing decisions in the input text. (2) Does the preprocessing of the embeddings’ underlying training corpus have an impact on the final performance of a state-of-the-art neural network text classifier, so, this study aims to identify the influence/role of preprocessing input data (text) on the performance of standard text classification models-based CNN.

**Research Methodology:** Four preprocessing data techniques are used, and they include tokenization, lowercasing, lemmatizing, and multiword grouping. Two classification models are used in this experiment. The first is the standard CNN with RELU activation function. The second model is a CNN with a recurrent layer (LSTM) with the idea of replacing multiple layers of convolution. To evaluate the outcome of the research, the two tasks considered are topic categorization and polarity detection. The datasets used for the topic categorization are the BBC news dataset, 20 News, Reuters, and Ohsumed. The dataset used for the polarity detection includes PL04, PL05, RTC, IMBD, and the Stanford sentiment dataset.

**Results:**Table 1 shows the accuracy of CNN and CNN+LSTM models for both topic categorization and polarity detection. Table 2 shows similar output but with cross-preprocessing (differently processed training data)

**Table 1: Preprocessing effect**

**Table

Description automatically generated**

**Table 2: Cross-preprocessing**

**Table

Description automatically generated**

From table 1, the lowest variance was observed in datasets with the largest training data, implying that preprocessing decisions are not so important in large training datasets. Tokenization only (Vanilla) appeared to be consistent across all datasets and tasks in both models and outperforms other techniques, with the only exception in the Ohsumed (medical domain) which requires fine-grained distinctions. Table 2 shows that multiword grouping outperformed other techniques generally. In conclusion, simple tokenization works the same way or better than complex preprocessing techniques like lemmatization except for domain-specific datasets. **On the IMDB dataset, the model has an accuracy of 88.9**. Also, the experiments show that preprocessing choice affects the variance of the result especially when the training data is not large enough for generalization.

References

Camacho-Collados, J., & Pilehvar, M. T. On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis. arXiv 2017. *arXiv preprint arXiv:1707.01780*.

Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2016). Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*.

Github Code: https://github.com/pedrada88/preproc-textclassification