Executive Summary: Linear Models for Text Understanding

Sheldon Benard
260618386

Elias Stengel-Eskin
260609642

Galen Bryant
260602804

1. Introduction

In our investigation, Linear Models for Text Understanding, we sought to answer: *Can models of significantly less complexity perform similarly (or outperform) complex models?* We explored the power of simple baseline classifiers in the context of text understanding, as these models typically require significantly less computation power and time, juxtaposed against the temporal CNN models in [1]. Ultimately, we showed that, given a thorough tuning of the hyper-parameters and proper utilization of optimization techniques, a logistic regression baseline classifier can match the performance of, and often outperform the CNN models.

2. Methodology

[1] utilized multinomial Logistic Regression for their Bag of Words and Bag of Centroids baseline models, and thus, we chose to explore the Logistic Regression classifier (one-vs-rest and multinomial) in our investigation.

For the replication task, our feature selection procedure mimicked that of [1]. For Bag of Words, the 5000 most frequent words were utilized to encode each data point as a 5000-dimensional vector. For Bag of Centroids, centroids were computed, by running the k-means algorithm on pre-trained Google News word embeddings with k = 5000, and used to encode each data point. The paper did not specify any additional pre-processing, and thus, we did no pre-processing for this task.

For the improvement task, we focussed on improving both the quality of the features selected and the efficiency and optimization of execution.

For feature selection, we introduced pre-processing, n-grams, and skips. Pre-processing, which included punctuation and stop-word removal and lower-casing the input strings, eliminated redundancy in the tokenization of the data points. N-grams and skips allowed for structural and contextual information to be taken into consideration by the model.

To increase efficiency and optimization, feature hashing, online learning, and early termination were explored. Feature hashing is a scalable solution to increasing the quantity of features which we used to expand the total number of features that our model could explore. Online learning ensured that gradient descent updates would take significantly less time, as single data points were used rather than the whole dataset. Lastly, early termination prevented overfitting and provided us with the optimal number of passes to use.

Vowpal Wabbit, of Microsoft Research [2], was utilized to combine all these considerations in one solution.

3. Results

Table 1 shows the test accuracy (percentage) for our models replicating [1]'s bag-of-words logistic regression baseline. All experiments achieved within 3.17% of the original paper. Our results are reported in blue.

Dataset	Z+L BOW	OVR	Multi
DBpedia	96.19	97.50	97.54
AmazonFull	54.17	52.12	52.36
AmazonPolarity	89.86	88.81	88.81
Yahoo	66.62	68.04	65.48
AG	86.69	90.47	89.86
Sogou	91.35	92.98	92.21

TABLE 1.

Dataset	Z+L BOC	OVR	Multi
DBpedia	89.09	93.66	94.12
AmazonFull	36.50	37.00	36.99
AmazonPolarity	72.86	73.71	73.72
Yahoo	56.47	57.50	58.27
AG	76.73	88.99	88.71

TABLE 2.

Table 2 similarly shows the test accuracy (percentage) for our models replicating [1]'s bag-of-centroids logistic regression baseline. All experiments except AG achieved within 5.03% of the original paper. It was unclear why AG outperformed the original baseline by so much, but it is worth remarking that it also outperformed the BOW baseline without any tuning. Our results are reported in blue.

Table 3 shows a comparison of [1]'s best CNN and our best logistic regression model (with optimal passes), for each dataset. Best results are reported in bold. Our models outperform the best CNN-based model for four of the six datasets.

4. Conclusion

By tuning the hyper-parameters of an online logistic regression baseline model, we were able to match and beat [1]'s performance. We were able to outperform temporal CNNs in 4 of the 6 datasets, and we were within 1.119% and 0.7625% for those that we did not beat. Crucially, all of our models could be trained in under 10 minutes (on a laptop), versus several hours to several days for the temporal CNNs (utilizing Tesla K40 GPUs). We find that models of significantly less complexity can outperform complex models for text understanding at a much smaller computational cost.

Dataset	Z+L best	Our best
DBpedia	98.40	**98.53**
AmazonFull	**59.57**	58.45
AmazonPolarity	**95.07**	94.31
Yahoo	71.10	**71.97**
AG	87.18	**90.47**
Sogou	95.12	**96.93**

TABLE 3.

References

[1] X. Zhang and Y. LeCun, “Text understanding from scratch,” *arXiv preprint arXiv:1502.01710*, 2015.

[2] A. Agarwal, O. Chapelle, M. Dudík, and J. Langford, “A reliable effective terascale linear learning system,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1111–1133, 2014.