OhioState at IJCNLP-2017 Task 4: Exploring Neural Architectures for Multilingual Customer Feedback Analysis

Dushyanta Dhyani
The Ohio State University OH, USA
dhyani.2@osu.edu

Abstract

This paper describes our systems for IJCNLP 2017 Shared Task on Customer Feedback Analysis. We experimented with simple neural architectures that gave competitive performance on certain tasks. This includes shallow CNN and Bi-Directional LSTM architectures with Facebook's Fast-text as a baseline model. Our best performing model was in the Top 5 systems using the Exact-Accuracy and Micro-Average-F1 metrics for the Spanish (85.28% for both) and French (70% and 73.17% respectively) task, and outperformed all the other models on comment (87.28%) and meaningless (51.85%) tags using Micro Average F1 by Tags metric for the French task.

1 Introduction

Customer Feedback Analysis (CFA) aims to analyze the feedback given by customers to various products/organizations. A primary component of CFA is to identify what the feedback is discussing so that further processing can be carried out appropriately. This requirement serves as a motivation for this shared task, which aims to classify user feedback in multiple languages into predefined categories and automate the process using machine learning methods for document classification.

2 Related Work

Document Classification is a well-studied problem in the NLP community with applications like sentiment analysis (Chen et al., 2016), language identification (Lui and Baldwin, 2012), email/document routing and even adverse drug reaction classification (Huynh et al., 2016). However, the problem and various proposed solutions are highly domain and use-case specific. State of the art sentiment analysis models/architectures that perform well for news articles fail to perform well for Twitter sentiment analysis. Moreover, the Twitter sentiment analysis models have to be redesigned for tasks like target dependent sentiment analysis (Vo and Zhang, 2015). This shows that the type of models used for a particular domain depends a lot on the data and the granularity of the categories. Recent efforts (Kim, 2014; Zhang et al., 2015; Conneau et al., 2016; Yang et al., 2016; Joulin et al., 2016) show the applicability of a single (generally neural) model over a variety of datasets, showing their capability to model text classification tasks in a domain and language agnostic way.

3 Task Description

The goal of the shared task is: Given customer feedback in four languages (English, French, Spanish and Japanese) the participants should design systems that can classify customer feedback into pre-defined categories (comment, request, bug, complaint, meaningless and undetermined). Evaluation is done on per-language basis using a variety of metrics.

3.1 Dataset

The contest organizers provided customer feedback data in four languages. The size of train/dev/test samples for each of the sub task is shown in Table 1. About 5% of the samples across the data splits for English, French and Japanese task have multiple labels, while each sample in the Spanish task has only one label. For the data samples with a single label, the distribution was highly biased towards certain classes, with the comment and complaint categories covering 80%-95% across all data splits for each sub-task.

Proceedings of the 8th International Joint Conference on Natural Language Processing, Shared Tasks, pages 170–173,
Taipei, Taiwan, November 27 – December 1, 2017. ©2017 AFNLP

The contest organizers also provided a relatively larger corpus of unlabeled data. While this could be used in different ways like learning domain-specific word embeddings, we exclude it in our experiments.

Language	Train	Dev	Test
English(en)	3066	501	501
Spanish(es)	1632	302	300
French(fr)	1951	401	401
Japanese(jp)	1527	251	301
Total	8176	1455	1503

Table 1: Number of Training Samples for each sub-task

3.2 Evaluation

The contest organizers use 3 metrics to evaluate the submitted systems

- **Exact Accuracy**: All tags should be predicted correctly.
- **Micro-Average F1**: As discussed in (Manning et al., 2008), micro-average F1 gathers document level decisions across classes, thus giving more weight to large classes, which is the case across all the sub-tasks
- **Micro-Average by Tags**: Label specific micro-average F1.

4 Proposed Approach

Motivated by the success of a variety of architectures for document classification task, we use multiple methods for the given challenge. We used a recently released open source tool called Fasttext as our baseline. In addition to that, we used a commonly used CNN architecture and multiple LSTM based architectures. In this section we discuss various components of our systems.

4.1 Pre-processing

We used minimal text pre-processing by using inbuilt tokenizer's from TensorFlow (Abadi et al., 2015) and Keras (Chollet et al., 2015) across all our architectures. In addition to that, we applied some elementary text cleaning to the English data only, given our lack of understanding of other languages.

4.2 Models

4.2.1 fastText (OhioState-FastText)

Given its ease of use, we used the fastText (Joulin et al., 2016) tool[1] as our baseline model. At its core, fastText is a linear model with a few neat tricks to make the training fast and efficient. It takes individual word representations and averages them to get the representation of the given text. This representation is then passed through a softmax to get class distribution. Training is performed using Stochastic Gradient Descent to minimize the negative log-likelihood over all the classes. We used most of the default parameters as in the original tool. We, however, found that the model performs best on the dev set when the embedding dimension is set to 200 and the model is trained for 100 epochs. Since the size of training data and number of training labels were small, we used the softmax loss function (and not the hierarchical softmax and negative sampling methods) as training time was not a constraint.

4.3 Convolution Neural Networks (OhioState-CNN)

We also performed some basic experiments with CNN's given their applicability to text classification (Kim, 2014; Zhang et al., 2015; Conneau et al., 2016; Kalchbrenner et al., 2014) problems. We used a simplified version of the architecture from (Kim, 2014) as discussed here[2]. We set the word embedding size to 100 and trained the architecture for 10 epochs (after which it starts overfitting) . We used 128 filters of filter width 3,4 and 5 and added a dropout layer with retention probability of 0.5. We trained the model using Adam (Kingma and Ba, 2014) and the sigmoid cross entropy loss.

4.4 Bi-Directional LSTM

LSTM's have been shown to be extremely effective for learning representations for text, not only for sequence to sequence labeling tasks, but for general classification tasks (Yang et al., 2016) as well as language modeling (Li et al., 2015). We use Keras' ability to plug and play layers to experiment with a couple of architectures.

- **OhioState-biLSTM1** : A single layer Bi-directional LSTM with an embedding layer

[1] We used the Python wrapper from pypi

[2] http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/

	Sub-Tasks				
	EN		ES	FR	
Models	**EA**	**MAF**	**Both EA & MAF**	**EA**	**MAF**
OhioState-FastText	**63.4**	**65.36**	82.94	68	70.49
OhioState-CNN	54.20	56.13	81.27	65	67.8
OhioState-biLSTM1	61.2	63.79	82.61	**70**	**73.17**
OhioState-biLSTM2	61.6	63.98	**85.28**	68.5	71.71
OhioState-biLSTM3	62.8	64.97	79.93	65	67.56

Table 2: Performance of Various Models for Exact Accuracy (EA) and Micro-Average F1 (MAF) score

ES		FR			
Both EA & MAF		**EA**		**MAF**	
Plank-multilingual	88.63	Plank-monolingual	73.75	Plank-monolingual	76.59
Plank-monolingual	88.29	IITP-CNN-entrans	71.75	IITP-CNN-entrans	74.63
IIIT-H-biLSTM	86.29	Plank-multilingual	71.50	Plank-multilingual	74.39
IITP-RNN	85.62	**OhioState-biLSTM1**	70.00	**OhioState-biLSTM1**	73.17
OhioState-biLSTM2	85.28	IIIT-H-SVM	69.75	ADAPT-Run1	72.68

Table 3: Top-5 Performing systems for the Spanish and French Sub-Tasks

for the vocabulary and a dense layer with a sigmoid activation for the class labels. We also added a Dropout layer (with retention probability of 0.3) after the LSTM layer.

- **OhioState-biLSTM2** : We added a 1D Convolutional (with ReLU activation) and MaxPooling layer after the word embedding layer which has shown to better represent n-gram like characteristics in text.

- **OhioState-biLSTM3** : We also added a Batch Normalization layer after the Convolution layer in the above architecture(though it decreased the performance)

Note that we did not make use of any pretrained embeddings. We used the same training parameters for the 3 Bi-LSTM variants discussed above: Word embedding dimension, LSTM unit size and Batch Size were set to 64. We used the Adam (Kingma and Ba, 2014) optimizer with binary cross entropy loss.

A few points worth mentioning: While the CNN and Bi-LSTM architectures were trained in a multi-label setting, at prediction time, we only predict the label with the maximum score. Also, Japanese text in the corpus either has no or a single space and thus tokenization is not effective. So even though we achieve some (unconvincing) results for the Japanese task, we do not consider them as relevant to this sub-task which requires more sub-word level treatment.

5 Results

We report the performance on 3 sub-tasks (leaving out Japanese for reasons previously discussed) for our models in Table 2 and comparison with systems designed by other teams in Table 3 using the exact accuracy and micro-average F1 metric.

While there is a considerable difference between our best performing system and the top systems for the English sub-task, we obtain competitive performance for the Spanish and French subtasks. Moreover, our LSTM based models outperform other systems for **comment and meaningless** category when evaluated using Micro Average by Tags metric for the French sub-task with an F-1 accuracy of 87.28% and 51.85% respectively. However, as shown in Table 4, our neural models failed to generalize to the infrequent labels as compared to a shallow model like fastText which is an expected behavior.

6 Conclusion

We propose some simple but effective neural architectures for customer feedback analysis. We show the effectiveness of LSTM based models for Text Classification in French and Spanish subtasks without any prior information like heavy pretrained embeddings, thus making it easy to perform fast and effective hyper-parameter tuning and

Task	Comments	Complaint	Meaningless	Bug	Request
EN	BiLSTM2 (77.8)	BiLSTM1 (63.4)	fastText (48.3)	fastText (16.7)	fastText (53.9)
FR	**BiLSTM2 (87.3)**	BiLSTM1 (57.4)	**BiLSTM1 (51.9)**	fastText (20)	fastText (15.4)
ES	BiLSTM2 (92.6)	BiLSTM2 (68.9)	0	0	fastText (31.6)

Table 4: Our best performing models (F1) for each label of the English, French and Spanish sub-task (Scores in bold perform best amongst all submitted systems)

architecture exploration.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *EMNLP*, pages 1650–1659.

François Chollet et al. 2015. Keras. `https://github.com/fchollet/keras`.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781*.

Trung Huynh, Yulan He, Alistair Willis, and Stefan Rüger. 2016. Adverse drug reaction classification with deep neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 877–887.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *IJCAI*, pages 1347–1353.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J Smola, and Eduard H Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*, pages 1480–1489.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.