CLASS-BASED PREDICTION ERRORS TO CATEGORIZE TEXT WITH OUT-OF-VOCABULARY WORDS

Joan Serrà[1], **Ilias Leontiadis**[1], **Dimitris Spathis**[2], **Gianluca Stringhini**[3], **& Jeremy Blackburn**[1]

[1] Telefónica Research, Barcelona, Spain – firstname.lastname@telefonica.com
[2] Aristotle University of Thessaloniki, Greece – sdimitris@csd.auth.gr
[3] University College London, United Kingdom – g.stringhini@ucl.ac.uk

ABSTRACT

Common approaches to text categorization essentially rely either on n-gram counts or on word embeddings. This presents important difficulties in highly dynamic or quickly-interacting environments, where the appearance of new words and/or varied misspellings is the norm. To better deal with these issues, we propose to use the error signal of class-based language models as input to text classification algorithms. In particular, we train a next-character prediction model for any given class, and then exploit the error of such class-based models to inform a neural network classifier. This way, we shift from the 'ability to describe' seen documents to the 'ability to predict' unseen content. Preliminary studies using out-of-vocabulary splits from abusive tweet data show promising results, outperforming competitive text categorization strategies by 4–11%.

1 INTRODUCTION

The first steps in automatic text categorization rely on the description of the document content. Typically, such content is characterized by some sort of word, stem, and/or n-gram counts (e.g. Wang & Manning, 2012) or, more recently, by unsupervised or semi-supervised word/sentence/paragraph embeddings (e.g. Arora et al., 2017). While the common pipeline performs well for a myriad of tasks, there are a number of situations where a relatively large ratio of seen vs. unseen tokens threatens the performance of the classifier. This is the case, for instance, with abusive language in online user content or microblogging sites (Nobata et al., 2016; Mehdad & Tetreault, 2016). In such cases, the volume of annotated data is not massive, and the vocabulary changes rapidly and non-consistently across sites and user communities. Sometimes these changes and inconsistencies are intentional, so as to hide the real meaning from traditional automatic detectors using hate-word dictionaries or blacklists. For example, the notorious online community *4chan* launched "Operation Google", which aimed to replace racial slurs on social media with the names of prominent tech companies in a sort of adversarial attack on Google's Jigsaw project (Hine et al., 2016).

To avoid relying on specific tokens, existing text classification approaches can incorporate so-called linguistic features (Brody & Diakopoulos, 2011), such as number of tokens, length of tokens, number of punctuations, and so on, or syntactic features (Nobata et al., 2016), based on part-of-speech tags, dependency relations, and sentence structure. Distributional representations (Le & Mikolov, 2014) and recurrent neural network (RNN) language models (Mikolov, 2012) are also used, together with more classical approaches involving word or character n-grams (Wang & Manning, 2012).

2 PROPOSED APPROACH

In essence, we propose to perform text categorization following a two step approach (Fig. 1). First, a language model for next-token prediction is trained with the data corresponding to each single category. Second, we use a normalized, sequential measurement of the performance of those language models as input to a neural network classifier. In a sense, we aim to shift from 'what is said' (ability to describe) to 'the way of saying' (ability to predict). The idea is that the language model should be tailored to a given category, but without developing a strong dependence on particular tokens that

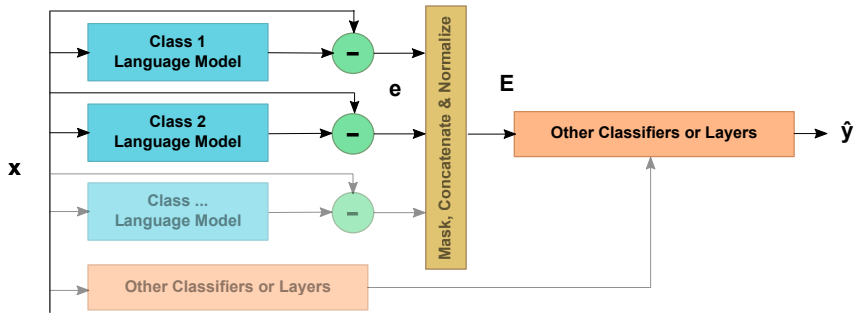

Figure 1: General schema of the proposed model.

are frequent or representative for that category. The concept of class-based errors is reminiscent of universal background models in speaker verification (Reynolds et al., 2000) and a few document attribution strategies in information retrieval (Serrà et al., 2012). Using error signals for classification also relates to derivative-based similarity and classification in time series (Keogh & Pazzani, 2001).

For the class-conditioned language model we use a character-based RNN. Sequences of one-hot encoded characters $\mathbf{X}$ are passed through a time-distributed dense layer with parametric rectified linear unit (PReLU) activation, which form a preliminary embedding. This is shared with the subsequent layers. For each class i, one gated recurrent unit (GRU) is used, followed by a time-distributed dense layer with softmax output, yielding the next-character prediction sequence $\hat{\mathbf{X}}^{(i)}$. With that and the delayed version of $\mathbf{X}$, we calculate a class-conditioned error sequence $\mathbf{e}^{(i)}$ for each character (and zero-mask it according to the sequence length, if needed). In our experiments with one-hot encoded character inputs, $\mathbf{e}^{(i)}$ corresponds to categorical cross-entropy. The final loss is taken to be

$$\mathcal{L}\left(y, \mathbf{e}^{(1)}, \dots \mathbf{e}^{(c)}\right) = \sum_{i=1}^{c} \left(\frac{\mathbb{1}_{\{i\}}(y)}{n-1} \sum_{j=1}^{n-1} e_j^{(i)} \right),$$

where y is the class label of the instance, c is the total number of classes, n is the sequence length of the instance, and $\mathbb{1}()$ is an indicator function.

To perform classification, we take the class-conditioned error sequences $\mathbf{e}^{(i)}$ provided by the language model, and concatenate them into a matrix $\mathbf{E} = [\mathbf{e}^{(1)}, \dots \mathbf{e}^{(c)}]$ for each instance. We then use an instance normalization layer

$$\bar{\mathbf{E}} = \sigma\left(g\mathbf{E}/\mu\right),$$

where $\sigma()$ is an activation function, g is a gain constant, and μ is the average over all the elements of $\mathbf{E}$ (taking sequence masking into account). In principle, we could learn g (or extended vector/matrix versions of it) and use any activation. However, we empirically found $g = 1/3$ and a sigmoid activation to work well with character-based cross-entropy sequences. After computing $\bar{\mathbf{E}}$, we input it to a two-layer neural network with PReLU activations, dropout, and a softmax output.

We train our models using Adam until we do not see an improvement on the validation set for 4 epochs. For the language model, we use layer dimensionalities 250 (embedding) and 500 (GRU). For the classification model, we use a dropout of 0.5 and dimensionality 100. At classification time, the language model's weights remain frozen.

3 Evaluation setup

To evaluate the suitability of the proposed concepts, we choose the task of detecting abusive tweets with out-of-vocabulary words, using a sample of 2 million tweets from the Twitter 1% feed. Half of the sample is selected based on their use of 'hate words' from a crowdsourced dictionary[1], while the other half is selected randomly. We manually filter the dictionary to remove overly common and ambiguous words like "india", a localized slur for dark skinned people.

[1] http://hatebase.org

Table 1: AUC scores for the considered baselines (see text) and the proposed approach. A null randomized model yields 0.514 (0.082) and 0.503 (0.090) for the Easy and Hard setups, respectively.

Setup	Word NB	Char NB	Word NB-LR	Char NB-LR	Char RNN	Proposed
Easy	0.900 (0.091)	0.849 (0.052)	0.912 (0.113)	0.902 (0.102)	0.858 (0.141)	**0.951** (0.080)
Hard	0.634 (0.109)	0.663 (0.080)	0.580 (0.089)	0.679 (0.038)	0.619 (0.101)	**0.705** (0.059)

To simulate data with new or heavily misspelled tokens, we decide to create semi-synthetic splits with out-of-vocabulary words. We randomly perform 10 train/validation/test splits, compute the area under the receiver operating characteristic curve (AUC), and report the mean and standard deviation over the 10 test splits. Two setups are considered:

- Easy – For the positive class, we take the hate-word list and split it into 70% for training and 15% for validation and testing. In performing such split, we force words with the same stem to end up in the same split. We then select tweets from the entire corpus that contain at least one stemmed word from each split. For the negative class, we just select randomly from the remaining tweets until we have balanced train/validation/test sets.
- Hard – Besides the list of hate words, we also consider a list of common words (top 1,000 to 3,000 most frequent English words[2]). We proceed as with the positive class of the Easy setup, and generate balanced train/validation/test splits for both abusive and non-abusive tweets. In addition, to increase difficulty, we remove tweets with list words appearing in more than one split (that is, we ensure that the intersection of listed words is null between train/validation/test).

4 RESULTS

We compare the proposed approach with 5 of the most common and competitive baselines in abusive language detection (Djuric et al., 2015; Mehdad & Tetreault, 2016). The first two are based on a naïve Bayes classifier using the TF-IDF of both word and character n-grams (NB). The next two are based on the approach proposed by Wang & Manning (2012): similarly, word and character n-grams are constructed, but NB ratios are calculated, and logistic regression is used as a classifier (NB-LR). For the previous baselines, we use all combinations of $\{1,2,3\}$-grams with cutoff frequencies of 20 (NB) and 100 (NB-LR). The fifth baseline is a character-based RNN using a time-distributed embedding layer, followed by a PReLU activation, a GRU, a dense layer with PReLU activation, and a dense layer with sigmoid output. We try different values for the dimensionality of the aforementioned layers and finally use 200 (embedding), 400 (GRU), and 200 (dense).

The result of the comparison is reported in Table 1. We see that, among the 5 baselines, there is no clear winner for the two setups. Word-based NB-LR performs best in the Easy setup and character-based NB-LR performs best in the Hard setup. Nonetheless, the proposed approach outperforms them by 4.2 and 3.8%, respectively. Compared to the average baseline performance, we observe a relative improvement of 7.5 and 11.0%. We also note that the standard deviation of the proposed approach (across runs) is comparatively low with respect to the baselines.

5 FUTURE WORK

We envision a number of potential extensions for the proposed approach: adding an 'all-class' predictor to the language models, improve (or learn) the error sequence normalization, studying the effect of adding further classifiers in parallel to the proposed classification model, ways of fusing those, play with class-based sentence or paragraph embeddings, etc. Generalizing the architecture to longer sequences is a main task for further research, perhaps considering recurrent, quasi-recurrent, or convolutional networks for the classification stage. We also plan to shift our focus to real-world, curated data sets of abusive language.

ACKNOWLEDGMENT: Work supported by the EU grant H2020-MSCA-RISE-691025 (ENCASE).

[2] http://www.ef.com/english-resources/english-vocabulary/

REFERENCES

S. Arora, Y. Liang, and T. Ma. A simple but tough to beat baseline for sentence embeddings. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2017.

S. Brody and N. Diakopoulos. Cooooooooooooooollllllllllllll!!!!!!!!!!!!!! Using word lengthening to detect sentiment in microblogs. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 562–570, 2011.

N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati. Hate speech detection with comment embeddings. In *Proc. of the Int. Conf. on World Wide Web (WWW)*, pp. 29–30, 2015.

G. E. Hine, J. Onaolapo, E. De Cristofaro, N. Kourtellis, I. Leontiadis, R. Samaras, G. Stringhini, and J. Blackburn. A longitudinal measurement study of 4chan's politically incorrect forum and its effect on the web. *ArXiv*, 1610.03452, 2016.

E. Keogh and M. Pazzani. Derivative dynamic time warping. In *Proc. of the SIAM Int. Conf. on Data Mining (SDM)*, 2001.

Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, volume 32, pp. 1188–1196, 2014.

Y. Mehdad and J. Tetreault. Do characters abuse more than words? In *Proc. of the Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pp. 299–303, 2016.

T. Mikolov. *Statistical language models based on neural networks*. PhD thesis, Brno University of Technology, 2012.

C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang. Abusive language detection in online user content. In *Proc. of the Int. Conf. on World Wide Web (WWW)*, pp. 145–153, 2016.

D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

J. Serrà, H. Kantz, X. Serra, and R. G. Andrzejak. Predictability of music descriptor time series and its application to cover song detection. *IEEE Trans. on Audio, Speech and Language Processing*, 20(2):514–525, 2012.

S. Wang and C. D. Manning. Baselines and bigrams: simple, good sentiment and topic classification. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pp. 90–94, 2012.