Using PPM for Health Related Text Detection

Victoria Bobicev
Technical University of Moldova
victoria.bobicev@ia.utm.md

Victoria Lazu
Technical University of Moldova
victoria.lazu@ia.utm.md

Daniela Istrati
Technical University of Moldova
daniela.istrati@ia.utm.md

Abstract

This paper describes the participation of the LILU team in SMM4H challenge on social media mining for health related events description such as drug intakes or vaccinations.

1 The Tasks and the Data

The challenge included four tasks (Weissenbacher, 2018); we participated in Task 1: Automatic detection of posts mentioning a drug name — binary classification; and Task 4: Automatic detection of posts mentioning vaccination behavior — binary classification.

The data included medication-related posts on Twitter. The training data was available on the challenge site[1].

For the Task 1 the organizers provided 9624 annotated tweets' id numbers; 9130 tweets we downloaded using this data. The data was comparatively balanced: 4730 tweets that mention drug names and 4400 tweets that do not mention any drug or dietary supplement. The evaluation set consisted of 5384 tweets.

For the Task 4 8180 annotated tweets' id numbers were provided. Only 6941 tweets we downloaded and the data was less balanced: 1979 tweets that mention influenza vaccination behavior and 4962 tweets that do not. The evaluation was performed on 161 tweets.

2 Method

We explored the PPM (Prediction by Partial Matching) model for automatic analysis of tweets. Prediction by partial matching (PPM) is an adaptive finite-context method for text compression that is a back-off smoothing technique for finite-order Markov models (Bratko et al., 2006). PPM produces a statistical language model which can be used in a probabilistic text classifier. Treating a text as a string of characters, a character-based PPM deals with different types of documents in a uniform way. PPM is based on conditional probabilities of the upcoming symbol given several previous symbols. A blending strategy for combining context predictions is to assign a weight to each context model, and then calculate the weighted sum of the probabilities:

$$P_{PPM}(x) = \sum_{i=1\ldots m} \lambda_i p_i(x), \qquad (1)$$

where $P_{PPM}(x)$ is the probability of the current character calculated using PPM method; $p_i(x)$ are conditional probabilities of this character on the base of the context of length i; λ_i are weights assigned to each conditional probability $p_i(x)$.

PPM is a special case of the general blending strategy. The PPM models use an escape mechanism to combine the predictions of all character contexts of length up to m, where m is the maximal length of the context; more details can be found in (Bobicev, 2007). The maximal length of a context equal to 5 in PPM model was proven to be optimal for text compression (Teahan, 1998) thus we used maximal length of a context equal to 5.

For example, the probability of character '***m***' in context of the word '***algorithm***' is calculated as a sum of conditional probabilities dependent on different context lengths up to the limited maximal length:

$$P_{PPM}(\text{'}\boldsymbol{m}\text{'}) = \lambda_5 \cdot p(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{orith}\text{'}) + \lambda_4 \cdot p(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{rith}\text{'}) + \\ + \lambda_3 \cdot p(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{ith}\text{'}) + \lambda_2 \cdot p(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{th}\text{'}) + \\ + \lambda_1 \cdot p(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{h}\text{'}) + \lambda_0 \cdot p(\text{'}\boldsymbol{m}\text{'}) + \lambda_{-1} \cdot p(\text{'esc'}),$$

Where λ_i is the normalization weight; 5 is the maximal length of the context; p('esc') is so

[1] https://healthlanguageprocessing.org/smm4h/social-media-mining-for-health-applications-smm4h-workshop-shared-task/

Proceedings of the 3rd Social Media Mining for Health Applications (SMM4H) Workshop & Shared Task, pages 65–66
Brussels, Belgium, October 31, 2018.

called ‘escape’ probability, the probability of an unknown character.

As a compression algorithm PPM is based on the notion of *entropy* introduced as a measure of a message uncertainty (Shannon, 1948).

Cross-entropy is the entropy calculated for a text if the probabilities of its characters have been estimated on another text (Teahan, 1998):

$$H_d^m = -\sum_{i=1}^{n} p^m(x_i)\log p^m(x_i) \qquad (2)$$

where n is the number of symbols in a text d, $H_d{}^m$ is the entropy of the text d obtained by model m, $p^m(x_i)$ is a probability of a symbol x_i in the text *d* obtained by model m.

The cross-entropy can be used as a measure for document similarity; the lower cross-entropy for two texts is, the more similar they are. Hence, if several statistical models had been created using documents that belong to different classes and cross-entropies are calculated for an unknown text on the basis of each model, the lowest value of cross-entropy indicates the class of the unknown text.

On the training step, we created PPM models for each class of posts; on the testing step, we evaluated cross-entropy of previously unseen posts using models for each class. Thus, cross-entropy was used as similarity metrics; the lowest value of cross-entropy indicated the class of the unknown posts.

PPM can be applied at the word level; however in most cases character level model better classify noisy texts with misspellings and slang (Bobicev, 2007).

3 Results

We performed a 10-fold cross-validation of the PPM based classification method on 6941 tweets, 1978 of which were from the positive class and 4963 from the negative class, and obtained: Precision = 0.839, Recall = 0.838, F-score = 0.839.

In order to improve the results we decided to remove less important words from the text before the model creation. The importance of words had been calculated using Gain Ratio (Quinlan, 1993):

$$GR = \frac{H(C)-\sum_{v\epsilon Vi} P(v)*H(C|v)}{-\sum_{v\epsilon Vi} P(v)\ \log \mathrm{P}(v)} \qquad (3)$$

where *H(C)* is class entropy; *Vi* are features (in our case words); *v* are feature values (in our case 0 or 1; presence or absence of the word) and *P(v)* are probabilities of these values. Then, we removed a small number of words with the smallest Gain Ratio and repeated the experiment obtaining Precision = 0.861, Recall = 0.858, F-score = 0.859. The final result on the blind test set was as follows: Precision = 0.841, Recall = 0.860, F-score = 0.850. The mean result for all participating teams: P=0.890, R=0.872, F=0.880.

We proceeded in the same way for the task 4 and obtained: Precision = 0.842, Recall = 0.814, F-score = 0.828. The final result on the blind test set was as follows: Precision = 0.829, Recall = 0.808, F-score = 0.818. The mean result for all participated teams: P=0.826, R=0.858, F=0.840.

4 Conclusion

Our results are lower than the mean in both described tasks. The reasons of the low accuracy may be: (1) PPM is not suitable for this type of text classification; (2) more preprocessing of the texts should be done before classification phase; (3) all terms in text are treated uniformly; they can be weighted in some way while used in calculations. We plan to implement more sophisticated preprocessing and term weighting during next year challenge.

References

Bobicev, V. 2007 Comparison of Word-based and Letter-based Text Classification. *RANLP V, Bulgaria*, pp. 76–80.

Bratko A., Cormack G. V., Filipic B., Lynam T. R., Zupan B. 2006. Spam filtering using statistical data compression models, Journal of Machine Learning Research 7:2673–2698.

Quinlan, J.R. 1993. C4.5: Programs for Machine Learning. *Morgan Kaufmann, San Mateo, CA*.

Shannon, C. E. 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656.

Teahan, W. 1998. Modelling English text, *PhD Thesis, University of Waikato, New Zealand*.

Weissenbacher, Davy, Abeed Sarker, Michael Paul, Graciela Gonzalez-Hernandez. 2018. Overview of the Third Social Media Mining for Health (SMM4H) Shared Tasks at EMNLP 2018, *in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018*.