We explored the PPM (Prediction by Partial Matching) model for automatic text language detection. 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). It obtains all information from the original data, without feature engineering, it is easy to implement and relatively fast. PPM produces a language model and can be used in a probabilistic text classifier. Treating a text as a string of characters, a character-based PPM avoids defining word boundaries; it deals with different types of documents in a uniform way. It can work with texts in any language and be applied to diverse types of classification.

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:

$$P(x) = \sum_{i=1}^{m} \lambda_i p_i(x),$$

where λ_i and p_i are weights and probabilities assigned to each order i $(i=1...m)$.

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 (i = 1…5) is the normalization weight;
5 is the maximal length of the context;
P('esc') is so called 'escape' probability, the probability of an unknown character.

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 m, where m is the maximum model. There are several versions of the PPM algorithm depending on the way the escape probability is estimated. In our implementation, we used the escape method C, named PPMC; 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). In all our experiments with character-based PPM model we used maximal length of a context equal to 5; thus our method is PPMC5.

Our utility function for text classification was cross-entropy of the test document:

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

where

n is the number of symbols in a text d,
$H_d{}^m$ – 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.

Usually, the cross-entropy is greater than the entropy, because the probabilities of symbols in diverse texts are different. 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 will indicate the class of the unknown text. In this way cross-entropy is used for text classification.

On the training step, we created PPMC5 models for each class of documents; on the testing step, we evaluated cross-entropy of previously unseen texts 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 text.

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

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