Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008)

An Effective and Robust Method for Short Text Classification

Victoria Bobicev
Technical University of Moldova
Studentilor, 7, Chisinau, Moldova
vika@rol.md

Marina Sokolova *
CHEO Research Institute
401 Smyth Road, Ottawa, Ontario, Canada,
msokolova@ehealthinformation.ca

Abstract

Classification of texts potentially containing a complex and specific terminology requires the use of learning methods that do not rely on extensive feature engineering. In this work we use prediction by partial matching (*PPM*), a method that compresses texts to capture text features and creates a language model adapted to a particular text. We show that the method achieves a high accuracy of text classification and can be used as an alternative to state-of-art learning algorithms.

Motivation

We focus on classification of texts with a high concentration of a specific terminology and complex grammatical structures. Those characteristics inevitably complicate standard feature engineering, which is done by language pre-processing (e.g., lemmatization, parsing) that is further complicated when the texts are short. Our goal is to avoid complex and, perhaps, error-prone feature construction by using a learning method that can perform reasonably well without preliminary feature engineering. We use *prediction by partial matching* (*PPM*), 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 original data, without feature engineering, is easy to implement and relatively fast. *PPM* produces a language model and can be used in a probabilistic text classifier.

The character-based *PPM* models were used for spam detection, source-based text classification and classification of multi-modal data streams that included texts. We opted to use the compression models for classification of terminology-intense data, e.g., medical texts. We applied *PPM*-based classifiers to the topic and non-topic classification of short texts, including classification of medical diagnosis. We built two versions of *PPM*-based classifiers, one calculating the probability of the next word and other calculating the probability of the next character. Our empirical results show that the *PPM*-based classifiers achieve a competitive accuracy of the short text classification.

*A part of the work was done at RALI, University of Montreal.

PPM Classifier

PPM is based on conditional probabilities of the upcoming symbol given several previous symbols (Cleary and Witten 1984). The *PPM* technique uses character context models to build an overall probability distribution for predicting upcoming characters in the text. 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: $p(\phi) = \sum_{i=-1}^{m} q_i p_i(\phi)$, where q_i and p_i are weights and probabilities assigned to each order i. *PPM* is a special case of the general strategy. The *PPM* models use an escape mechanism to combine the predictions of all character contexts of length $\leq m$, where m is the maximum model order; the order 0 model predicts symbols based on their unconditioned probabilities, the default order -1 model ensures that a finite probability (however small) is assigned to all possible symbols. The *PPM* escape mechanism is more practical to implement than weighted blending. 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 (Bell, Witten, and Cleary 1989).

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; more details can be found in (Bobicev 2007). We, however, built both word-based and letter-based *PPM* classifiers to compare their performance. Our utility function was: $H_m^d = -\sum_{i=1}^{n} p^m(x_i) \log p^m(x_i)$, where n is the number of symbols in a text d, H_m^d – 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. H_m^d was estimated by the modelling part of the compression algorithm. On the training step, we created PPM models for each class of documents; on the testing step, we evaluated cross-entropy of previously unseen texts using models for each class. The lowest value of cross-entropy indicates the class of the unknown text.

Empirical results

We applied our method on Newsgroups, clinical texts, and Reuters-21578. We tested the *PPM models*: word-based with orders $0, 1, 2$ and letter-based with order 5. The results