Applying the Perceptron Rule for Extraction of Keywords from Abstracts

Yaakov HaCohen-Kerner
Department of Computer Science
Jerusalem College of Technology (Machon Lev),
21 Havaad Haleumi St., P.O.B. 16031, 91160 Jerusalem, Israel

kerner@jct.ac.il

Abstract

In this ongoing work, we investigate a model that applies the perceptron training rule for extraction of keywords from abstracts and titles. We have tested our model on a set of abstracts of Academic papers containing keywords composed by the authors. Current results have not indicated any success of our learning approach.

Keywords: abstracts, keywords extraction, learning, perceptron, text summarization, titles

For extracting and learning keywords from abstracts and titles, we use three basic features: (1) frequency of terms, (2) importance of sentences and (3) length of the terms in words.

Our first hypothesis, is that the importance of a sentence is determined by its location according to the following list in descending order: (1) the title of the paper, (2) the first sentence of the abstract, (3) the last sentence of the abstract, and (4) any other sentence of the abstract.

Our second hypothesis - based on the statistical analysis of distribution of the length in words of authors’ keywords - is that most of the keywords should be 2-grams, then unigrams and then 3-grams. Therefore, their weights should be set respectively. The learning algorithm, presented in the next paragraph, was designed to achieve the best weights and to check these hypotheses.

The learning of the weights has been done by the perceptron training rule, described in Fig. 1. Where the left \( w_i \) is the weight of feature \( #i \) after the learning, the right \( w_i \) is the weight before the learning the weight \( #i \), \( \varepsilon \) represents a small constant, (e.g.: 0.01, in order to proceed in small and stable changes), \( t \) represents the training value, \( o \) represents the output value, and \( x_i \) represents the actual value of feature \( #i \). The learning algorithm is described in Fig. 2.

\[
 w_i = w_i + \varepsilon \cdot (t - o) \cdot x_i
\]

Fig. 1. The perceptron training rule

For each abstract from the data base
1. Create word weight matrices for all unigrams, 2-grams and 3-grams. High frequency close class words (e.g.: we, this,) are excluded.
2. Compute the weights of all unigrams, 2-grams, 3-grams by counting full and partial appearances.
3. Select the \( n \) highest weighted groups of words as the proposed keywords by the system.
4. For each value of \( i \)
   for each value of \( k \)
   \( w[i][k] = w[i][k] + \varepsilon \cdot \text{errors_count}[i][k] \)

Fig. 2. The learning algorithm

Where \( w[i][k] \) represents the sentence’s weight, \( i \) is the kind of the sentence, \( k \) is the number of words in a keyword, and \( \varepsilon \) equals 0.08.

We have tested our model on 80 Academic documents. Each document includes a title of a paper, its abstract, and a list of keywords composed by the author.

Before learning full matches were found at a rate of 21.99% and partial matches and up were found at a rate of 61.14%. After learning, full matches were found at a rate of 22.59% and partial matches and up were found at a rate of 61.75%.

The hypotheses we assumed have been carried out almost fully. However, the learning results present only slight improvements. Possible explanations to this findings might be: (1) the set of the initial weights is not optimal, (2) the perceptron training rule is not the right learning rule to use in this task, (3) extracting and learning of keywords should be done from entire articles rather than from titles and abstracts.

Future directions for research are: (1) selecting an optimal set of initial weights, (2) testing different learning techniques, and (3) elaborating the model for extracting and learning keywords from entire articles.

Acknowledgements Thanks to Ari Cirota and two anonymous referees for valuable comments on earlier versions of this document.