MI	CT-TEC003A-20140625	**Pages:**	1/3
Development:		**Module:**	KorAutoWS
Product Name:	Korean Automatic Wordspacer	**Version:**	1.0.1
Delivery:	**??/??/2014**	**Author(s):**	Minsu Ko; Woonho Choi
License:	**Term Limited-** 31.JUL.2014.	**Customer(s):**	

1 Product Introduction

Korean automatic wordspacer (KorAutoWS) is an algorithm designed for the task of automatically tokenizing the spaces among the words in a document collection. The algorithm employs an idea that defining a conditional probability distribution over label sequences given a particular observation sequence in accordance with the conditional random fields (CRFs) among the characters, rather than a joint distribution over both label and observation sequences..

1.1 Algorithm

The probability of a particular label sequence y given observation sequence x to be a normalized product of potential functions is defined as in each of the form below [LAFFERTY01],

$$exp(\sum_j \lambda_j t_j(y_{i-1}, y_i, x, i) + \sum_k \mu_k s_k(y_i, x, i)), \tag{1}$$

where $t_j(y_{i-1,y,x,i})$ is a transition feature function of the entire observation sequence and the labels at positions i and $i-1$ in the label sequence; $s_k(y_i, x, i)$ is a state featrue function of the label at position i and the observation sequence; and λ_j and mu_k are parameters to be estimated from training data.
When defining feature functions, we construct a set of real-valued features $b(x, i)$ of the observation to expresses some characteristic of the empirical distribution of the training data that should also hold of the model distribution as in the formula 2.

$$b(x, i) = \begin{cases} 1, & \text{if the observation at position i is the word} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Each feature function takes on the value of one of these real-valued observation features $b(x, i)$ if the current state (in the case of a state function) or previous and current states (in the case of a transition function) take on particular values. All feature functions are therefore real-valued. For example, consider the following transition function:

$$t_j(y_{i-1}, y_i, x, i) = \begin{cases} b(x, i), & \text{if } y_{i-1} \rightarrow SPACE \text{ and } y_i \rightarrow SPACE \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The notation can be simplified by writing

$$S(y_i, x, i) = s(y_{i-1}, y_i, x, i) \tag{4}$$

and

$$F_j(y, x) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, x, i) \tag{5}$$

where each $f_i(y_{i-1}, y_i, x, i)$ is either a state function $s(y_{i-1}, y_i, x, i)$ or a transition function $t(y_{i-1}, y_i, x, i)$. This allows the probability of a label sequence y given an observation sequence x to be written as

$$p(y|x, \lambda) = \frac{1}{Z(x)} exp(\sum_j \lambda_j F_j(y, x)) \tag{6}$$

1.2 Characteristics

The characteristics of KorAutoWS enables to process the Korean text document files. The character encoding is UTF8 (LF). This program needs the burning time for loading the trained model at the initial stage.

1.3 Time Complexity

The brute-force procedure for the solution of this problem is the generation of all possible N^Tstate sequences and calculating the joint probability of each state sequence with the observed series of events. This approach has time complexity $O(N^2T)$, where T is the length of sequences and N is the number of symbols in the state alphabet.

2 Functional Specification

2.1 Dependencies

The following depedencies are required for the installation of the KorAutoWS in a Linux(Debian/Redhat/CentO
MacOS, Windows 32/64bit environment: JDK 6 or higher.

2.2 Methods

Library : libKorAutoWS_Limited-1.0.jar

- AutomaticWordSpacing.runAWS(String inputPath) :

3 Technical Support

3.1 Policy

KorAutoWS must be installed by a technician from Owlnest on site/via remote. Policies on the source code is not included in the Short-Term Limited license. This software is protected by copyright law of South Korea. If you want to use this software after the limited term, please contact us.

3.2 Contact

- Person-in-Charge :
- Email : contact@owl-nest.com
- Phone No. : +82-02-742-3021

Referências

[1] J. Lafferty, A. McCallum, F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. International Conference on Machine Learning, 2001.