Automatic Word Spacing

Sachin Sai Krishna M Sakthivel S Karthikeyan M
College of Engineering, Guindy
Anna University
Chennai-600025
sachinstar2010@gmail.com

ABSTRACT

To design an efficient system to automatically insert spaces in a chunk of text without spaces. Many manuscripts and classical literature are now being converted to digital format using Optical Character Recognition. This system is not completely reliable as, during conversion, plenty of spacing errors arise. This project is designed to correct all these spacing errors using unsupervised learning.

1. INTRODUCTION

This project aims to build a tool that automatically spaces words in a body of text. We have adopted a model-free approach for this. Our motivation for this project is the fact that text messages occupy 25 spaces for 160 characters. If we could remove and reinsert these spaces, we could fit in more words in a 160-character message.

The novelty of this tool is that this is one of the first tools to be developed for English language, the only other language being Korean, for which there are many such tools. This can be extensively used in mobile messaging systems.

2. RELATED WORK

Previous approaches for automatic word spacing can be classified into two groups: rule based approach and statistical approach. The rule based approach uses lexical information and heuristic rules (Choi, 1997; Kim et al., 1998b; Kang, 1998; Kang, 2000). Lexical information consists of postposition information. Heuristic rules are composed of longest match or shortest match rule, morphological rules, and error patterns. This approach has disadvantage requiring higher computational complexity than the statistical approach. It also costs too much in constructing and maintaining lexical information. Most of rule-based systems use a morphological analyzer to recognize word boundaries. Another disadvantage of rule-based approach is resulted from using morphological analyzer.

First, if ambiguous analyses are possible, frequent backtracking may be caused and many errors are propagated by an erroneous analysis. Second, results of automatic word spacing are highly dependent on the morphological analyzer; false word boundary recognition occurs if morphological analysis fails due to unknown words. In addition, if an erroneous word is successfully analyzed through overgeneration, the error cannot even be detected.

The statistical approach uses syllable statistics extracted from large amount of corpora to decide whether two adjacent syllables should be spaced or not (Shim, 1996; Shin and Park, 1997; Chung and Lee, 1999; Jeon and Park, 2000; Kang and Woo, 2001). In contrast to the rule-based approach, it does not require many costs to

construct and to maintain statistics because they can be acquired automatically. It is more robust against unknown words than rule-based approach that uses a morphological analyzer.

3. MODULE DESCRIPTION

3.1 TOKENIZATION

Tokenizing is nothing but converting a sequence of characters to a sequence of tokens. Takes raw text, splits words by their morphological aspects. Punctuation and whitespace may or may not be included in the resulting list of tokens. All contiguous strings of alphabetic characters are part of one token; likewise with numbers. Tokens are separated, by whitespace characters, such as a space or line break, or by punctuation characters.

3.2 FREQUENCY DATA DICTIONARY

Since this approach spaces words based on the previously occurred words in the corpus, a frequency data dictionary is generated in this module. This dictionary stores the frequency of each word that has occurred in the last but other chapters. In case of ambiguity among words which are selected by our algorithm, the word with the higher frequency gets selected.

3.3 WORD-SPACING TOOL

This tool, which was built on our own, uses KMP algorithm to find the match at the 0-th position and from there proceeds to find the longest match and in case of ambiguity, goes on to apply the same for the immediately following text to resolve the ambiguity. Even if this fails, the word with the higher frequency in the frequency data dictionary becomes the candidate. Once the candidate is found, a space is inserted and this proceeds till the end of the document. Any new word not in the frequency data dictionary is resolved using WordNet and is automatically added to it.

4. PROPOSED SYSTEM

The system considers a book in an electronic format. We take the last chapter in that book and remove all the spaces. We then use the tool to insert all the spaces back. We can easily calculate the accuracy of the newly generated text since we have the original text in hand.

The basic idea is using machine learning to learn a model of English based on all the other chapters of the book as a training corpus, using only unsupervised learning. After constructing a frequency data dictionary from the last but other chapters, we use a modified KMP-algorithm and a forward search to resolve the spaces. The match is found by starting at the 0-th position and then proceeding till the longest match and the selected word is compared against the frequency data dictionary to check the correctness.

5. EVALUATION METRICS

Once the spaces are resolved, they are compared against the original text to calculate the efficiency of the algorithm.

Efficiency = (RW) / (OW)*100

RW – Resolved Text
OW – Original Text
We have been able to achieve an efficiency of
…...

6. FUTURE WORK

This model-free approach's computation time is quite large, mainly due to the forward search procedure implemented. For an unsupervised learning approach, depth-k search can be used to improve this computation

time. As for as efficiency is concerned, in unsupervised learning, using semantic analysis along with frequency data dictionary and WordNet could greatly improve the efficiency, which is left as future work.

7. REFERENCES

1. Do-Gil Lee and Sang-Zoo Lee and Hae-Chang Rim. Automatic Word Spacing Using Hidden Markov Model for Refining Korean Text Corpora

2. Un Yong Nahm. Text mining with Information Extraction: Mining Prediction Rules from Unstructured Text.

3. Kang. 2000. Eojeol-block bidirectional algorithm for automatic word spacing of Hangul sentences. Journal of the Korea Information Science Society.

4. Jeon and H.-R. Park. 2000. Automatic word-spacing of syllable bi-gram information for Korean OCR post processing. In Proceedings of the 12th Conference on Hangul and Korean Information Processing