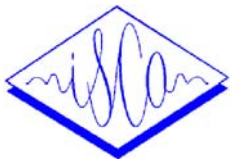
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2004 - ICSLP
8th International Conference on Spoken Language Processing
ICC Jeju, Jeju Island, Korea
October 4-8, 2004

Chinese Text Word-Segmentation Considering Semantic Links among Sentences

Leonardo Badino

Loquendo, Vocal Technology and Services
Turin, Italy
leonardo.badino@loquendo.com

10.21437/Interspeech.2004-246

Abstract

Tokenization of Chinese input text into words is a necessary step to realize a Mandarin Chinese text-to-speech. Several word-segmentation algorithms were developed in which linguistic information are combined with statistical ones or with heuristic rules. In this paper we investigate in the advantages that can arise when semantic relation among sentences is taken into account during the word segmentation process. The algorithm we propose shows how this kind of semantic information could improve the performances of a word segmentation algorithm.

1. Introduction

Text-to-speech conversion needs an initial step for tokenizing the input text into words. Such step was not critical in the development of European languages for the Loquendo multilingual TTS the tokenization module is considered quite simple with the exceptions of some critical points. Developing a text-to-speech system for Mandarin Chinese we encountered the largely studied problem of segmenting Chinese text into dictionary words (see the definition by [1]). This problem arises from the absence of explicit word delimiters (equivalent to the blank space in written English) between the Chinese ideograms. One may be erroneously led into believing that this problem could be easily circumvented, simply by transcribing a character (i.e. an ideogram) at a time without concerns to where a certain word ends and a new one begins. In actual fact, in order to achieve an acceptable quality in speech synthesis, it is necessary that the text be decomposed into single words. This need is dictated by a number of factors:

- each single ideogram may have different forms of pronunciation depending on the words it belongs to;
- certain phonologic and phonetic rules depend on correct word separation: for instance a so-called tonal sandhi phonologic rule provides that in the presence of two syllables (i.e. two ideograms) each conveying a third tone, the former will change its tone if the two syllables belong to the same word;
- the information relating to each word is necessary in order to permit a correct grammatical and syntactic-prosodic analysis.

Fortunately, punctuation symbols that delimit sentences are adopted in written Chinese, so the problem of segmenting the whole input text can be traced back to the problem of word-segmentation of sentences.

The difficulty of Chinese word identification arises from the very common phenomenon for written Chinese, of word boundary ambiguity that causes different tokenizations a Chinese sentence can be normally segmented in. Not all the possible segmentations are always plausible. As an example the sentence (written in simplified Chinese and taken from [2]):

1. 日文章鱼怎么说

lacking of word delimiters, has the two segmentations:

1a. 日文(Japanese) 章鱼(octopus) 怎么(how) 说(say)

and

1b. 日(Japan) 文章(essay) 鱼(fish) 怎么(how) 说(say)

but the first only (meaning “How do you say octopus in Japanese?”) has a plausible meanings.

On the other hand, the sentence:

2. 我喜欢新 西 兰花

can be segmented as

2a. 我 (I) 喜欢(like) 新西兰(New Zeland)花(flowers)

or

2b. 我 (I) 喜欢(like) 新 (fresh) 西 兰花(broccoli)

and both segmentations are plausible sentences .

There is a large number of word identification algorithms based on approaches ranging from pattern matching to statistical method. The most used pattern-matching methods are based on the Maximum Matching heuristics. Among these the most popular is the MM method (one example is [3]) also known as forward maximum tokenization (see [1] for a mathematical definition): starting at the beginning of the sentence, the longest word starting at that point is found (looking up at Chinese dictionary), then, starting at the ideogram following the found word, the process is repeated until the end of the sentence is reached. Other methods following the Maximum Matching heuristics are backward maximum tokenization, shortest tokenization and critical tokenization[1]. These kinds of approach give good results when all the sentence words are in the dictionary but perform poorly when unknown words have to be identified.

Statistical methods give generally better results and several statistical techniques were implemented. In these approaches usage frequencies of words, co-occurrence frequency of ideograms, and probability of tagging are used and constraint-satisfaction models are often included. For example, in the relaxation approach ([4]) all possible words in a sentence are identified and assigned an initial probability based on their

usage frequency. In the next steps these probabilities are updated iteratively by using the adjacency constraint among words, so impossible words are filtered out, leading to the most likely sentence word segmentation.
In some cases, syntactic and semantic analysis of the sentence to be processed, are included in both pattern-matching and statistical method (see [5]).

2. Our Proposal

In all works to our knowledge, each sentence is processed without employing any information about the previous sentences, so every sentence is considered separately from the text containing it. Our purpose was to investigate on the advantages arising from taking into account the semantic context of sentences. We supposed that, in every segmentation method in which probabilities or costs are associated to words, and the highest probability (or the lowest cost) word segmentation is selected, the segmentation of a sentence would be improved by incrementing probability (or decrementing cost) of content words that have been identified in previous semantically related sentences.
The aim of our work was to develop a method able to update such costs on the basis of the "strength" of the semantic links between each sentence to be segmented and the previous segmented sentences.

2.1. Evaluating semantic relation among sentences and updating probabilities

The input text to the Mandarin Chinese LoquendoTTS could be either mono-topic or multi-topic. We report the intuitive definition of topic taken from [6]:
"The notion of 'topic' is clearly an intuitively satisfactory way of describing the unifying principle which makes one stretch of discourse 'about' something and the next stretch 'about' something else, for it is appealed to very frequently in the discourse analysis literature"(pp.39).
In the hypothesis of a multi-topic text, there is a sequence of topics and the shifts between two contiguous topics can be of different kinds: for instance, in a text where two completely different news are reported, one talking about a football match and the other talking about a rock concert, the topic shift is "stronger" than a shift between two subtopics of the same piece of news.
Our aim is to force the word segmentation module to choose (where possible) the segmentation including words identified in previous topic-related sentences (reference words). For instance, if the sentence of the example 2 occurs in a text talking about New Zeland, then the algorithm might be able to choose 2a.
This target can be obtained employing a Dynamic Vocabulary, in which words identified during every stage of word segmentation, are inserted with an assigned cost.
The cost is decremented every time the word is encountered in a new sentence. If the word is not found any more for a number of consecutive sequences it will be discarded from the Dynamic Vocabulary. In this way lexical items of "old" topics aren't take into account for new segmentations.
The Dynamic Vocabulary alone doesn't suffice to take into account the semantic context, and a further device is necessary to avoid that word segmentation of a sentence be influenced by "the past" also when there has been a topic shift. An explicit value, expressing the strength of the semantic relation between sentences must be defined. For this last task, we have been inspired by works concerning the problem of topic identification (and subtopic) boundaries and their strength evaluation. The more the boundaries are strong the less the word tokenization of the current sentence will be influenced by the previous word tokenizations.

2.1.1. Evaluating strength of topic boundaries

Topic boundaries identification is a complex problem, that we won't discuss here, and that we have treated being inspired by Hearst ([6]). According to him and his TextTiling approach, we make two important assumptions:

- we considered the sequence of topics in a text as a linear one so topics and subtopics can be seen as adjacent units.
- we evaluate the strength of topics (subtopics) change by lexical co-occurrence patterns.

The second point means that we make the following assumption: "a set of lexical items is in use during the course of a given subtopic discussion, and when the subtopic changes, a significant proportion of the vocabulary changes as well" (taken from [6], p. 40). Making this assumption, a problem immediately arises: How can we detect topic changes on the basis of lexical co-occurrence patterns if word delimiters don't exist in a Chinese text? Our solution consists in taking into account all that words identified in the text, so words belonging to implausible word segmentations are considered too. As an example the word 鱼 (fish) was identified in the sentence 1 but it didn't belong to the right segmentation 1a. Yet, the topic boundaries identification approach so applied will be disturbed by the noise due to words that are identified in the sentences but not belong to the right words segmentations. So this approach will be less efficient than a topic boundaries identification approach employed for texts of languages where word delimiters exist.
Instead of identifying topic boundaries in the whole input text and then segmenting all sentences taking into account the strength of topic boundaries where they occur, we adopted a slightly different approach that is less expensive in terms of computational time and memory request. For each sentence to be segmented our algorithm evaluates the relation of the sentence with the already word segmented text in terms of lexical co-occurrence assigning a score and considering lexical items as before, both for already segmented text and current sentence. The score we assign is the ratio of the number of all content words identified in the current sentence to the number of words of the current sentence that are found in the Dynamic Vocabulary (where there are also words identified but not selected in the best segmentations). Naturally, all the words inserted in the Dynamic Vocabulary have to be content words: not all the identified words can be inserted in the Dynamic Vocabulary because not all words are topic indicators: for instance, conjunctions don't provide much information about topic divisions because they are uniformly distributed along all the topics. Nevertheless, we know content words are not always "informative", so further constraints, based on vocabulary frequency and heuristic rules, will be employed in the future.

3. Flow of Process

The characters accepted by Mandarin Chinese LoquendoTTS are ideograms and Latin characters with Unicode encoding.
Now we'll describe step by step the Chinese input text word segmentation and the subsequent transliteration from ideograms to pinyin. Pinyin is a sort of phonemic transcription based on Latin characters showing how Chinese words are pronounced.
Given the input text, the first syntagm or sentence is identified by heuristic rules and segmented into words. Then, this process is iterated until the end of the text. In this paper the term syntagm stands for a string of character ending with a "weak" punctuation (as comma, colon, etc…) followed by a blank.
The word segmentation of each sentence needs two fundamental steps. In the first step, a sort of lattice or matrix is built, in which all the elements are words identified by looking up in a static vocabulary *LEX*, or special sequences (such as dates, hours, and so on) identified by heuristic rules. When these two kinds of search fail, unknown words are inserted into the lattice. To all words of the lattice a cost is associated. These costs are computed by employing a dynamic vocabulary *DLEX*. In the second step, the lattice is processed by a Dynamic Programming algorithm that draws out the word segmentation with the minimum cost.
The next sessions will explain in detail the two mentioned steps focusing on updating of word costs based on the semantic links among sentences.

3.1. Lattice construction

The lattice is created having as many columns as the characters in the syntagm (or sentence), whereby a character can be associated to each column. The number of rows varies depending on the columns and corresponds to the number of words located in the static lexicon *LEX* or identified by heuristic rules, having the character corresponding to the column as the first character. Each entry of the *LEX* vocabulary is made up of two fields: a word written with Chinese ideograms (in the Simplified form) and a tag indicating if the word is a content word or not. The heuristic rules describing "special words" (such as dates, hours, sequences of Latin characters and so on) are represented by Finite State Automata.
Starting from the first character (on the left) in the syntagm/sentence, if a fragment of the sentence is accepted by one of the Finite State Automata embodying the heuristic rules, then it will be identified as a word. Also the longest word of *LEX* starting with that ideogram is searched, then the second longest one, and so on by ending up with the ideogram itself.
When none word is found, starting from the current ideogram, then at most a fixed number (for instance five) of unknown words are created: the first word is the current ideogram, the second is made up of the current ideogram and the following ideogram (if it exists) and so on.
For all words identified (known and unknown) a maximum cost is allotted. The maximum costs are necessary to compute the costs of the lattice words and, when a word is identified for the first time, its lattice cost will be its maximum cost, so the maximum cost can be seen as a starting cost. Those words that are found in the static vocabulary *LEX* all have the same cost C_{Lex}, higher than the cost allotted to special words: C_{rule}. For unknown words, a higher cost C_{ukn} is assigned with respect to the costs considered in the foregoing: the cost C_{ukn} is as higher as longer is the unknown word. This metric we employed avoids a long sequence of ideograms will be recognized as an unknown word even if it is made up of *LEX* words; on the other side, if sequences of ideograms are really made up of only unknown words, the maximum costs we assign to unknown words, avoid these sequence will be always segmented in mono-ideogram words only.
The metric we used to assign maximum costs is very simple because we supposed its simplicity could best underline the semantic context influence on the word segmentation of a sentence, with respect to more complex metrics.
After this first search, all the identified words (known and unknown) are searched in the dynamic vocabulary *DLEX* that is empty at the beginning of the process and is filled during it. If an identified content word is absent in the *DLEX*, it is recorded in *DLEX* with an assigned cost C_{dlex} that is equal to the corresponding maximum cost (C_{Lex}, C_{rule} or C_{ukn}) decremented of a constant K_{dec1}. If the word is already recorded in the dynamic lexicon then its C_{dlex} is updated. Not all the words found in *DLEX* have the same kind of updating because we supposed that repeated adjacency of more ideograms should receive a lower cost than a repeated single ideogram. So the updating function depends on the number of ideograms of a Chinese word: if a Chinese monosyllabic word (that is, a word made up of a single ideogram) is already recorded in the dynamic lexicon, the C_{dlex} is equal to the cost assigned to at the recording in *DLEX*, while in all the other cases the C_{dlex} cost is decremented of a constant K_{dec2} (lower than K_{dec1}). In this way polysyllabic words can reach values lower than the minimum threshold cost of monosyllabic words.

3.2. Computing the minimum cost segmentation

Whenever the lattice is completed a Dynamic Programming algorithm ([7]) is employed to compute the best-cost segmentation.
Nevertheless, this algorithm doesn't have to work directly with C_{dlex} costs, because, as we explain in section 2.1, these costs have to be modulated on the basis of the topic relation between current sentence and previous segmented sentences. According to the considerations made in the section 2.1.1, we associate a score to the semantic relation that is the ratio of the number of all content words identified in the current sentence to the number of words of the current sentence that are found in *DLEX*. The words inserted in *DLEX* are all the words identified before choosing the best-cost segmentation, so some of these couldn't belong to the best-cost segmentation; in this way we avoid error propagation, i.e. we avoid that a wrong word segmentation could influence the word segmentations of next sentences. On the other side, we have to accept the "noise" due to those words that are identified but not belong to the right segmentation.
After this step, the Dynamic Programming algorithm can begin: starting from the last position in the sentence/syntagm the sequence with the lowest cost is searched for each word in the column. Given a word identified by the line j and the column i of the lattice (hereinafter referred to simply as $W_{i,j}$) the lowest cost sequence starting from it is given by the following formula:

(1) $MinCostW_{i,j}=Min_{(over\ k)}\{CostW_{i,j}+MinCostW_{(i+lengthWi,j),k}\}$

In the equation (1), if $W_{i,j}$ was absent in the *DLEX* before the word segmentation of the current sentence:

(2) $CostW_{i,j} = C_{fs}$

Otherwise:

(3) $CostW_{i,j}=C_{dlex}+(C_{fs} - C_{dlex}) * (1-Nol/Nw) / k$

Where C_{fs} represents a minimum cost (i.e. one among C_{Lex} , C_{rule} or C_{ukn}), k is a constant, *Nw* is the number of all content words identified in the sentence (or in the sentence of the syntagm) and *Nol* is the number of sentence words found in *DLEX*. The ratio *Nol/Nw* is the indicator of the strength of the semantic links between the current sentence and the already word segmented text. If *Nol/Nw* is equal to zero (for instance, this is the case of the first sentence of the text), then the word cost will be equal to the static cost of the word, while if *Nol/Nw* is equal to 1, then the word cost will be equal to the *DLEX* cost of the word.

The process of words searching (in static and dynamic lexicons), updating, and Dynamic Programming employing for the best cost segmentation computing, is repeated for each sintagm/sentence and after every sentence (not sintagm) word segmentation, the cost of each word present in *DLEX* is updated adding a constant K_{inc}; then, *DLEX* words having a cost higher than their maximum costs (i.e. C_{fs}) are removed from *DLEX.*

Whenever the segmentation process is completed, all Chinese words belonging to best segmentations are transliterated into pinyin form: consequently, the output of our Chinese tokenization module is a sequence of pinyin words, numbers, and other words, like foreign words, that aren't in a pinyin form. This output becomes the input of a language-independent tokenization module that identifies by heuristic rules other "special" token like internet and e-mail addresses, abbreviation written by Latin character and so on.

4. Results and Discussion

In our tests we tried to investigate on the advantages of tacking into account the semantic context of sentences. We have supposed that the quality of our methods could depend on the constants we used for word-cost updating and sentence semantic relation, i.e. the constants we employed for decrementing or augmenting the costs of words in the DLEX, and the k constant used in (3). For instance, high values for constants decrementing costs could let our algorithm select a high number of reference words (high recall), nevertheless the occurrence of new words could be ignored and reference words could be erroneously selected (low precision).

After a series of segmentations of different kinds of test text we fixed the values of constants choosing those values that gave a good compromise between precision and recall value (a higher value of precision was preferred with respect to the recall value). Then, in the test session, we compared results obtained using a fixed word-costs, where no semantic context is considered, algorithm (named A0) with results obtained employing word-costs updated as described above. The goodness of each segmentation was evaluated by two human judges. We compared the segmentations of two different text set: one set with one text only, made up of 310 news randomly taken from the Xinhua corpus (news from January 1990 and March 1991) in which every piece of news was two or three sentences long so several topic boundaries occurred, and another text set with a very lower change of topic made up of five long articles (30-40 sentences per article) taken from "People's Daily". Using the first set, we found 30 different segmentations: among these, 20 segmentations were due to a better segmentation of our algorithm, while the others 10 segmentations were due to a better segmentation of algorithm A0. With the second set we found 18 differences, among these 16 were due to a better segmentation of our algorithm and 2 to a better segmentation of algorithm A0.

Comparison between Chinese word segmentation systems isn't a simple problem because of the disagreement of human judges; however, in our tests, this problem doesn't arise because the differences we found were almost always (with the exception of four cases) between an implausible segmentation and a plausible one.

In some cases our algorithm can identify unknown words that more than once occur in topic related documents but this capability wasn't underlined in our test because of the inability of A0 of recognizing unknown words. Although we didn't focus on unknown words identifications, our approach seems to demonstrate that repeated occurrences of ideograms sequences in topic related sentences could be useful information for the unknown words identification task.

5. Conclusions

In this paper we proposed a method for segmenting Chinese sentences into words, taking into account the semantic links between the sentence to be segmented and the previous segmented sentences. Empirical results demonstrated that such semantic information can improve word segmentation and help identifying unknown words.

6. References

[1] Guo, Jin. "Critical Tokenization and its Properties".1997. In *Computational Linguistics*, 23(4): 569-596.

[2] Sproat, R., W. Gale, C. Shih and N. Chang. "A Stocastic Finite-State Word-Segmentation Algorithm for Chinese". 1996. *Computational Linguistics,* 22(3): 377-404.

[3]Tsai, Chih-Hao. 2000. "MMSEG: A Word Identification System for Mandarin Chinese Text Based on Two Variants of Maximum Matching Algorithm". Avaible on internet at http://www.geocities.com/hao510/mmseg.

[4] Fan, Charng-Kang and Wen-Hsiang Tsai. 1998. "Automatic word identification in Chinese sentences by the relaxation technique". *Computer Processing of Chinese and Oriental Languages*, 4(1): 33-56.

[5] Gan, K., K. Lua, M. Palmer. " A Statistically Emergent Approach for Language Processing: Application to Modeling Context Effects in Ambiguous Chinese Word Boundary Perception". 1996. In *Computational Linguistics*, 22(4):531-533.

[6] Hearst, Marti A.."TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages". *Computational Linguistics*, 23(1): 33-64.

[7] Bellman, R. E.. 1957. *Dynamic Programming*. University of Princeton Press.