Digits and Numeral Expressions Analysis in Machine Translation

Bei Li[1, a], Ling Zhang[2,b] and Ying Wang[1,c]

[1] *Computer Information Centre, Beijing Institute of Fashion Technology, Beijing, China*

[2]*Beijing Information Technology College, Beijing, China*

[a]*jsjlb@bift.edu.cn,* [b]*jzhangling-n@bitc.edu.cn,* [c]*jsjwy@bift.edu.cn*

Abstract.

In order to improve the morphology analysis of expressions containing numerals for machine translation, this paper analyzes corresponding corpus and sums up characteristics of such expressions and proposes a handling strategy based on dynamic template matching and knowledge base. The workings and procedures of the morphology analysis are also described in this study.

Keywords: Template matching; Digit; Numeral; Morphology Analysis; Machine Translation

Introduction

A machine translation system works mainly by performing morphology analysis and semantic analysis over text entered. Morphology analysis includes segmentation and word level handling. Besides handling words in the general sense (e.g. nouns, verbs, adjectives), the system has to deal with words and phrases representing numerals as well as expressions that have specific meanings and are related to digits and numerals in the word handling process. For example, July 28, 1998 (an English expression of a certain date) is not an expression referring to the number of something but an expression related to digits and numerals. All expressions similar to this are in the realm of this study. Their meanings are expressed in the form of a whole language unit. During the

morphology analysis, the handling of such terms is significantly different from that of words in the general sense.

It is an important task for a successful machine translation system and even the entire natural language comprehension at large to correctly, effectively identify and handle digits and numeral expressions. This paper analyzes corresponding corpus, proposes a handling method based on dynamic template matching and knowledge base and explains its workings with examples.

Corpus Analysis

It's highly essential to analyze the problem and learn about its unique features before solving it. There are numerous language forms composed of digits and numerals which roughly fall into three categories, namely, digits, dates and formula.

The first category includes such digits as 23800, 23, 800, 23 8000, 10.25, -5. The numerals like one hundred and one (101), half a million (500,000), one million, two hundred and thirty-four thousand, seven hundred and fifty-three (1,234,753) are counted as digits as well.

There are different expressions for dates, e.g. 1972-04-20, 1972/04/20, 04/20/1972 and July 20, 1971.

A typical form of a formula is 2*3/5.

There are also many other unclassified forms such as 50%, ¥1000, 4th, 202.20, 196.2, etc.

There are a variety of expression units that have specific meaning with digits and numerals as their main elements. According to a large amount of analysis of the corpus extracted from the Internet, this study has found more than 570 kinds of expressions related to digits or numerals. According to the analysis, these language forms related to digits or numerals show the following characteristics:

Diversity of Expression Forms and Complexity in Handling. In a natural language, digits and numerals are used to express dates, time, fraction, currency, formulas and percentage more than figures. There is a big difference among these

forms of expressions. Moreover, a meaning often can be expressed in different forms. For example, April 20th, 1971 can be written as 1971-04-20, 1971/04/20 or 04/20/1971. The diversity of expressions brings about the complexity and difficult maintenance of the translation system.

Correlation among Language Elements. There is a strong interconnection between language elements in an expression related to digits or numerals. It can be said that the entire expression can be seen as an organic whole. In most cases, such an expression can be seen as an indivisible word. At the same time, the correlation form among language elements is somewhat fixed.

Lacking of Corresponding Words. The thorough analysis of the corpus has revealed a fact that however diverse and complex the expressions are, the words and phrases used are only a small collection in a language. Such a collection is mainly composed of cardinal numerals, ordinal numerals, words representing months, some symbols (e.g. period, comma, ¥, double quotation marks, etc.) and some special words (e.g. the, half, an etc. in English). Therefore, a clue has been found for solving the problem.

Handling Strategies

Based on the thoughts above, this study considers using the method of template matching and knowledge analysis. In this manner, a model based on dynamic template and knowledge base is built for identifying and handling digits and numeral expressions. See Fig 1 for the structure of the model.

Identifiable Lexicon. By analyzing the corpus, a conclusion has been drawn that the expressions to be processed are composed of a specific group of words including cardinal numerals, ordinal numerals, words representing months and weeks, some symbols, etc. First, a special lexicon containing these words (including their derivatives) is constructed to identify words in character streams imported. A template is generated in this manner. Such a lexicon contains not only words and their derivatives but also their properties such as template identifiers, word class and translation.

Template and Its Handling Function Library. A template is a description of the structure of an expression entered. The template is formed by a group of capital letters and punctuation marks where each capital letter represents a category of identifiable words. For example, there may be the following rules for a template:

D: Arabic numerals, e.g. 0, 1, 2, 3, etc.

N: cardinal numeral, e.g. one, two, three, etc.

X: ordinal numeral, e.g. first, second, etc.

M: month, e.g. May, July, etc.

W: week, e.g. Monday, Tuesday, etc.

Punctuation marks only represent themselves. For example, colon is shown as ":" in a template.

It should be noted that each capital letter represents a single word from the expression entered except D and N which represent a group of neighboring words of a kind. Here are some examples. The template for 120 is "D". The template for 1200.34 is "D.D". The template for one thousand two hundred is "N". The template for July 28, 1998 is "MD, D".

Structure and Content of the Knowledge Base. With the generation and matching of templates, language elements in an expression are reshaped into a specific form. This has provided a sound basis for further analysis. Since the relevance among language elements in an expression is different, it's highly necessary to analyze such relevance so that the system can process the expression correctly. This system has adopted the basic ideas of SC grammar conditional lookup function, expanded them to build the expression forms and rules in the knowledge base and applied them in identifying and handling digits and numeral expressions. The basic form of a rule is shown as follows:

<head mode> → *<condition mode>*|*<handling function>*

<head mode> is the current template and its element form including some intermediate template symbols, also known as nonterminal symbols. *<condition mode>* is the condition of a rule and represents the context lookup condition and

value condition of the handled element. <*handling function*> refers to the function call to be used.

For example, N represents a template composed of a numeral while TWON represents a template containing two numerals, a nonterminal symbol. The rules for the two templates are as follows:

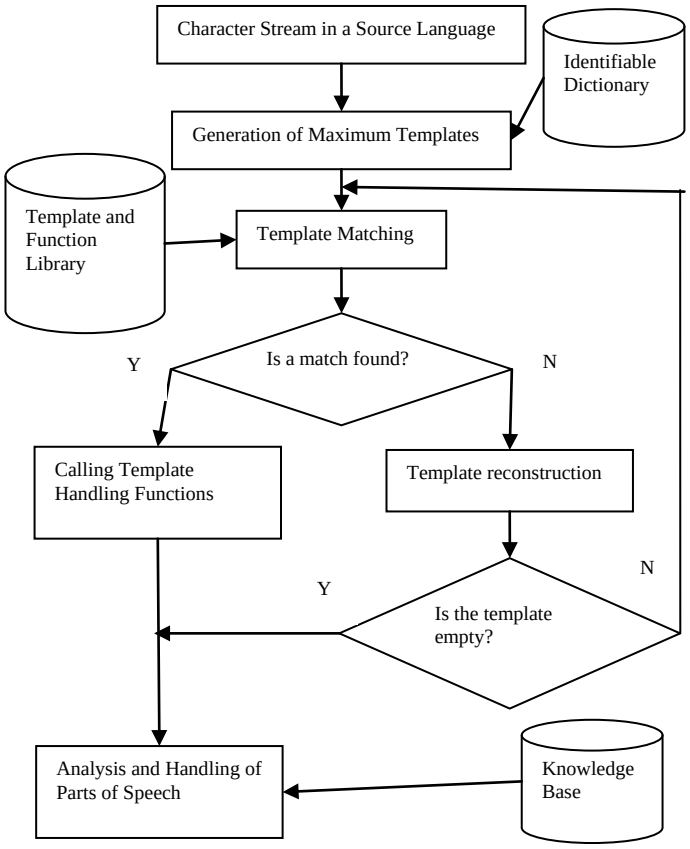

Fig. 1 Workings of Template Matching

RULE1： N→Count (BOF,EOF)=2|TWON
RULE2:TWON→Search (L,(1,BOF),in(20-90))|Is Num(TWON)
RULE3:TWON→Search (L,(1,BOF),in(1-24))|Is Time(TWON)

BOF and EOF represent the head and tail of an input string, respectively. Handled according to these rules, forty seven is identified as a numeral expression

while five three is identified as an expression of time. But twenty five may be a numeral expression or an expression of time. So, syntactic analysis of the context is required to rule out such an ambiguity.

Input Character Stream in a Source Language. The caller in the upper level is used to identify, segment and separate text. This is implemented by the caller scanning the source language. When the system is about to scan and identify a new word, the caller calls this identification function to first identify digits and numeral expressions.

Generation of the Maximum Template. The maximum template is the one generated for the first time. When the caller scans the beginning character of a new word, it transmits the current character pointer to this function. Then, the function scans the character stream in a proper sequence and attempts to identify each word according to the identifiable lexicon. A template is generated at the same time. The caller stops scanning once its encounters an unidentifiable word. The template generated at this moment is the maximum template.

Template Matching. The template library stores all identifiable templates for digits and numeral expressions as well as implementation functions of corresponding templates. The template generated during the scanning can be compared with those in the template library. If a match is found, the template matching is successful. Then, a proper handing function is called to deal with the language elements corresponding to this template. The handling process may be different for different expression forms. Translation is required in some cases. Numeral phrases can be directly converted into Arabic numerals.

Reconstruction of Templates. If no match is found during template matching, the existence of identifiable language elements is still uncertain. The reason why a match is not found may be that the generated template is out of the realm of the identifiable templates. In this case, reconstruction of templates is required. As the objects to be identified are the language elements of the word from the character stream that are pointed by the character pointer and transmitted by the caller, the tail truncation method can be used to reduce the length of the template, making it

closer to the pointed word after every truncation. In each truncation, a template element at the tail is abandoned. In this way, a new template is generated. Then, another template matching can be conducted according to this template. If no match is found, more tail truncation operations are performed until a match is found or the template becomes empty which marks the end of template matching.

Analysis and Handling of Parts of Speech. The general composition of the expression can be obtained from the template matching results. By analyzing the properties of words, the meaning of the expression as a whole may be known as well. Analysis of parts of speech is based on the knowledge based. By executing a series of rules, the information of corresponding parts of speech is obtained (including ambiguous information). Moreover, forms of certain expressions can be converted or pretreated for translation (for example, numeral expressions can be directly translated into Arabic numerals). Thus, the morphology analysis of digits and numeral expressions is completed.

Conclusions

Machine translation is a complicated system engineering project that involves theories and technologies is multiple disciplines such as computational linguistics, artificial intelligence and computer application. There are many questions yet to be solved in the development and commercialization of machine translation products. This paper first analyzes the characteristics of digits and numeral expressions as well as requirements for a machine translation system and then proposes a method for handling such expressions based on dynamic template matching and knowledge base. This method has successfully used in multilingual machine translation websites. Facts have proved the ideal performance of this method. However, there are still some aspects that need to be improved. The flaws include ambiguity in form, uncertainty in knowledge expression, etc. which may provide some clues for future discussion.

References

[1] Tong R, Huang H Y, Chen Z X and Song J P. An approach to processing format information streams of networking machine translation systems. Journal of computer research & development. 2000, 37(10): 1271-1275.

[2] Chen Z X. A new context-sensitive subcategory (SC) grammar for machine translation. Chinese Journal of Computers. 1992, 15(11): 801-808.

[3] Zhang B Q, Liu Z D, Huang H Y. The Morphology analysis technology of expressions including digits and numerals for machine translation. Computer Engineering and Applications. 2002, 18: 80-82.