Text Mining by Pseudo-Natural Language Understanding

Ruqian Lu
Chinese Academy of Sciences, China

INTRODUCTION

Text mining by pseudo natural language understanding (TM by PNLU for short) is a technique developed by the AST group of Chinese Academy of Sciences, as part of the project automatic knowledge acquisition by PNLU, which introduces a partial parse technique to avoid the difficulty of full NLU. It consists of three parts: PNL design, PNL parser implementation and PNLU based automatic knowledge acquisition. Its essence is twofold: a trade-off between information gain and feasibility of parsing, and a rational work division between human and computer,

BACKGROUND

Experts in knowledge engineering have been agreeing on the basic point of view that the most challenging problem in the construction of knowledge based systems is how to acquire enough and high quality domain knowledge. People usually use natural language understanding techniques to acquire knowledge from technical literature. As it was shown by practice, natural language understanding has always been a very difficult problem.

Experiences have shown that many practical applications do not need a complete and perfect understanding of the natural language texts. By lowering down the requirement of NLP a bit and providing the computer with human help a bit we can let the computer process and understand "natural language" texts, including acquiring knowledge from it, massively, automatically and efficiently. This was the goal of developing the technique of TM by PNLU.

MAIN FOCUS

Definition of PNL

Let's use the notation PNL for pseudo natural language and PNLU for PNL understanding. The former denotes a class of languages, while the latter denotes a kind of technique for processing PNL. Generally speaking, PNL looks very similar to natural language, but can be understood, analyzed and compiled by computer to an extent by which it can meet the need of some application, for example compiling a text book in a knowledge base for expert consultation.

Design of a PNL

The process of designing a PNL is as follows:

1. Determine a set of semantic constructs of the application domain. For example, if the domain is mathematics, then the semantic constructs are case frames providing sentence semantics frequently used in mathematics textbooks, like concept definition, theorem proving, exercise presentation, etc.
2. Select a natural language, e.g. English, as background language;
3. Look for sentence patterns in this language, whose meaning corresponds to the semantic constructs selected in the first step. These sentence patterns may look like: **if * then * is called ***; **since * is true and * is not true we can infer from * that * is true**.
4. Organize the set of selected sentence patterns in grammar form and call it the key structure of the language, which is now called pseudo-natural. That means: not every combination of sentence patterns is a legal key structure. All parts marked with stars will be skipped by any PNL parser. We call these parts "don't care".

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.

PNL Grammar

As an example, the following tiny grammar implies the key structure of a general classification statement:

⟨Classification Sentence⟩::=[⟨Leading word⟩<***Don't care***>,]⟨classification leading sentence⟩
⟨Leading word⟩::=**According to | Based on**
⟨classification leading sentence⟩::=⟨***Don't care***⟩⟨classification word⟩[⟨number⟩[**main**]⟨type word⟩[,⟨sequence of ***Don't care***⟩]] | There⟨be word⟩⟨number⟩[**main**]⟨type word⟩of⟨***Don't care***⟩. They are⟨sequence of ***Don't care***⟩.
⟨classification word⟩::=⟨be word⟩**classified into** | ⟨mood word⟩**be classified into**
⟨be word⟩⟩::= **is | are**
⟨type word⟩::= **classes | types | sorts | kinds** ……

This grammar may recognize sentences like: *Blood cells* **are classified into two types,** *red blood cells*, *white blood cells***.**

Parsing a PNL Text

While parsing a PNL text, the computer tries to understand the text, based (and only based) on the semantics of the underlying key structure. This understanding is necessary superficial in the sense that no information other than that implied by the key structure will be gained by the computer. It abstracts away all unnecessary details regarding the current application and thus makes the language understanding much easier. For example: consider the sentence:

If the color of the blood cell is red **than** the blood cell **is called** erythrocyte.

This is a (shallow) definition of red blood cell. But here we can already see the abstraction principle of PNL. With this knowledge, a computer can answer questions like "what is erythrocyte?," "How do we call a blood cell when the color of the blood cell is red?" etc. even without knowing the meaning of "red" or "cell."

Mechanism of TM by PNLU

The mechanism of TM by PNLU can be roughly described as follows:

1. Design a PNL;
2. Implement a compiler, which can parse PNL texts, acquiring knowledge from it and organizing it in a domain knowledge base;
3. Each time when knowledge is to be acquired, use an OCR device to scan the documents into the computer. Modify the scanned texts slightly to turn them in their PNL form;
4. Let the computer parse and analyze the PNL texts and produce a knowledge base (, which may need to be integrated with an existing knowledge base).

A series of PNL in different domains have been developed for automatic knowledge acquisition and system prototyping, including BKDL for expert systems, EBKDL and SELD for ICAI systems, DODL/ BIDL for Management Information Systems and KUML/WKPL for knowware engineering.

Layers of PNL

For the same application domain, one can divide the key structure of a PNL in several layers. There are three basic layers: the core layer, which contains sentence patterns used in all domains; the domain layer, which contains technical expressions used in a particular domain; and the jargon layer, which contains professional expressions used by a particular group of users. Each time when one designs a new PNL, the core layer, which occupies the major part of the key structure, does not have to be modified. Only part of the domain layer should be renewed. The jargon layer is usually very small and does not play an important role. Of course it is also possible to define intermediate layers between the basic layers.

Spectrum of PNL

It is easy to see that if we enlarge the key structure of PNL, then the computer will acquire more detailed knowledge from a PNL text. For example, if we add the sentence pattern "**color of * is ***" to the key structure, then the computer may additionally know that color is an attribute, which can be used to describe physical objects. On the other hand, if we reduce the key structure, then the knowledge acquired by the computer will have a larger granule. In this way, PNL defined with different key structure form a spectrum, which is a partial order. The upper limit of this spectrum is the natural

3 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/text-mining-pseudo-natural-language/11085

Related Content

Scientific Web Intelligence
Mike Thelwall (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 1714-1719).*
www.irma-international.org/chapter/scientific-web-intelligence/11049

Web Design Based on User Browsing Patterns
Yinghui Yang (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 2074-2079).*
www.irma-international.org/chapter/web-design-based-user-browsing/11105

Data Mining in the Telecommunications Industry
Gary Weiss (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 486-491).*
www.irma-international.org/chapter/data-mining-telecommunications-industry/10864

Data Mining with Cubegrades
Amin A. Abdulghani (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 519-525).*
www.irma-international.org/chapter/data-mining-cubegrades/10869

Data Mining in Protein Identification by Tandem Mass Spectrometry
Haipeng Wang (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 472-478).*
www.irma-international.org/chapter/data-mining-protein-identification-tandem/10862