Advanced Engineering Informatics xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

A two-level parser for patent claim parsing ☆

Jingjing Wang [a,b,*], Wen Feng Lu [c], Han Tong Loh [d]

[a] School of Statistics, Jiangxi University of Finance and Economics, Nanchang 330013, China
[b] Research Center of Applied Statistics, Jiangxi University of Finance and Economics, Nanchang 330013, China
[c] Department of Mechanical Engineering, National University of Singapore, Singapore
[d] Singapore Institute of Technology, Singapore

ARTICLE INFO

Article history:
Received 8 April 2014
Received in revised form 25 January 2015
Accepted 28 January 2015
Available online xxxx

Keywords:
Patent search
Product design
Patent claim
Parsing
Dependency syntax
Parser

ABSTRACT

Patent claim parsing can contribute in many patent-related applications, such as patent search, information extraction, machine translation and summarization. However, patent claim parsing is difficult due to the special structure of patent claims. To overcome this difficulty, the challenges facing the patent claim parsing were first investigated and the peculiarities of claim syntax that obstruct dependency parsing were highlighted. To handle these peculiarities, this study proposes a new two-level parser, in which a conventional parser is imbedded. A patent claim is pre-processed in order to remove peculiarities before passed to the conventional parser. The process is based on a new dependency-based syntax called Independent Claim Segment Dependency Syntax (ICSDS). This two-lever parser has demonstrated promising improvement for patent claim parsing on both effectiveness and efficiency over the conventional parser.

1. Introduction

Patent analysis approaches are generally classified as patent mining or visualization [1]. The rapid growth of published patents has made a call for more sophisticated patent analysis tools. Patent parsing is an essential operation in many patent mining approaches such as function-behaviour-state information extraction [2], concept-based patent search [3] and conceptual graph extraction [4]. However, patent claim parsing is considered very difficult [5]. The claim section of a patent defines the scope of Intellectual Property (IP) protection granted by the patent. In a patent, the claim section is the only part that are examined and conferred for IP protection. In contrast, other parts like the description section or drawings are used for understanding and interpreting the claims, but do not provide any IP protection themselves. Semantic patent claim analysis can examine patents for possible infringements and identify which needs to be manually perused [6]. Moreover, patent claims are important to the value of the patent [7], especially claims in essential patent i.e. those patents that are indispensable for designing and manufacturing products [8]. Besides, the number of claims a patent makes has significant effects on the duration that a patent is under consideration [9]. It is expected that the improvement on patent claim parsing can promote more sophisticated patent analysis and therefore tackle the challenges of the rapid growth of published patents.

The patent claim syntax follows exactly common English grammar but is peculiar [10]. These peculiarities are usually not considered when designing a conventional natural language parser. Therefore, conventional natural language parsers may fail to correctly parse patent claims [5].

To design a completely new data-driven parser, it is necessary to prepare both the training data and a data-driven model that can handle the peculiarities of patent claim syntax. In this way, existent natural language resources are discarded. In contrast, a smarter way is to utilize existent natural language resources by improving the adaptability of a conventional parser. This study applies the latter approach and proposes a two-level parser for patent claim parsing. At the top level, patent claims are pre-processed so that a conventional parser e.g., Stanford parser [11] can be more adaptable to them; while at the bottom level, the conventional parser is evoked to parse the pre-processed claims.

To build such a two-level parser, the peculiarities of the claim syntax that lead to challenges of claim parsing were investigated. A new dependency-based syntax, called Independent Claim Segment Dependency Syntax (ICSDS), was then proposed in order to address these challenges. The two-level parser was finally build based on the proposed dependency-based syntax.

☆ Handled by C.-H. Chen.

* Corresponding author at: School of Statistics, Jiangxi University of Finance and Economics, Nanchang 330013, China. Tel.: +86 0791 83816428.
E-mail addresses: wang_jingjing@jxufe.edu.cn (J. Wang), mpelwf@nus.edu.sg (W.F. Lu), HanTong.Loh@SingaporeTech.edu.sg (H.T. Loh).

http://dx.doi.org/10.1016/j.aei.2015.01.013
1474-0346/

The rest of this paper is organized as the following. The related works are reviewed in Section 2. The Section 3 highlights the challenges facing the patent claim parsing. The Section 4 introduces the proposed approach, including the new dependency-based syntax and the parser system. The Section 5 evaluates both the effectiveness and efficiency of the proposed parser. Lastly, the Section 6 addresses conclusions and gives recommendations for future work.

2. Related works

A complex knowledge-based natural language analysis approach was proposed [12] to capture both the structure and content of a claim text. The knowledge includes both shallow lexicon and predicate lexicon. The shallow lexicon is a word list which was automatically acquired from a corpus of five million words in US patents. The predicate lexicon for claims on apparatuses was manually acquired from a corpus of 1000 US patent claims. It was expected that the proposed claim parsing can be used in machine translation and improvement on the readability of patent claims. However, with regard to performance, this approach was not compared with other approaches.

Since not only structure but also content is useful in patent claim analysis, claim parsing in this study focuses on dependency parsing. Compared with phrase structure (or constituency) parsing, dependency parsing offers an easier way to extract the content of a claim. This is because dependency grammar [13] can explicitly express word-to-word relations. Further, the result of dependency parsing can be converted from that of phrase structure parsing [14]. The phrase structure grammars have a high proportion in formal grammatical systems. Thus, many existent natural language resources can be reused in dependency parsing.

Dependency parsing methods are generally classified into two categories: grammar-based parsing or data-driven parsing. The grammar-based parsing requires grammar or rules, e.g., context-free dependency grammar. In contrast, data-driven parsing does not need grammar or rules; it relies on models, which are learned from training data, to make decisions. The learned models can be classified into graph-based models [15,16], transition-based models [17,18], or hybrid models [19–21].

A simple way to realize domain adaptation is by correcting and enriching the training set of a data-driven parser with domain-specific data. In this way, previous work [22] had retrained a Bohnet's parser. However, the improvement on performance is not very promising. This is probably because the parser lacks a mechanism to handle the peculiarities of patent claim. Besides, effectiveness improvement should consider natural language processing. Most works focus on the detection of a restricted number of prominent verbal relations, including in particular is-a, has-part and cause, while deriving a large number of content relations relies on deep syntactic structures [23].

In parsing, the length of a sentence is the number of tokens, which are words and punctuations in the sentence. Similarly, the length of a claim can be defined as the number of tokens in a claim. The length of an independent claim is usually too long so that the claim cannot be parsed [10]. To improve the efficiency of patent claim parsing, a common strategy is segmentation. An approach [10] was proposed for reducing the length and the complexity of patent claim via claim decomposition. The decomposition can obviously improve the success rate of parsing with a Stanford parser. However, no evaluation results were given to show the accuracy of parsing after decomposition. A finite-state machine [4] was implemented to split a patent claim into a set of sub-sentences before passed to a Stanford parser. This finite-state machine was designed for handling two claim's forms. However, these two claim's forms can not cover all claims. Two segmentation tasks [24] were carried out. The first task segments a claim into three components: preamble, transition and body text; the second task further segment the claim into subordinate and coordinate clauses. The evaluation of these two tasks only focuses on claim segmentation rather than the effectiveness of claim parsing.

Briefly, little previous works had focused on patent claim parsing. It should be noted that the efficiency of patent claim parsing can be improved with claim segmentation. However, it is still a question whether claim segmentation will depress the effectiveness of patent claim parsing. Moreover, simply enriching training set of a data-driven parser does not make very promising improvement on performance. A mechanism is needed to handle the peculiarities of patent claim.

3. Peculiarities of claim syntax

Claim syntax follows exactly English grammar. However, compared to daily English usage, claim syntax is peculiar [10]. This study focuses on those peculiarities that may increase the difficulty of dependency parsing. These peculiarities are highlighted in the following sub-sections.

3.1. Claim template

There are some formal templates for starting a claim. For example, "file folder" is the patented product found in US Patent 7,954,694. The first independent claim starts with "*We claim*:" and a dependent claim starts with "*The file folder of claim 3, wherein*". It is necessary to use these templates for organizing multiple claims.

These templates do not offer any information pertaining to the patented product. In other words, removing these templates does not lead to useful information loss. However, the existence of these templates does increase the difficulty of dependency parsing.

3.2. Post attribute past participle

For regular verbs, the verb form of their past forms is the same as that of their past participles. In claims, the past forms are rarely used since the basic tense is present tense rather than past tense. On the other hand, it is frequent to use a complex noun phrase with post attributive present participle phrase or post attributive past participle phrase. However, given such a complex noun phrase with post attributive past participle phrase, a conventional parser is prone to mark the past participle as a past form. It is because a conventional parser usually treats the input text as a sentence rather than a noun phrase.

3.3. Parenthetical sentence

Parenthesis means that additional word, phrase or sentence is inserted into a passage which would be complete without it. Insertion of an independent sentence is rare in daily English usage, but it is frequent in claims. This increases the difficulty of dependency parsing because a conventional parser usually treats the input text as a single sentence.

3.4. Complex noun phrase as sentence

In claim, it is frequent that an independent sentence consists of only a complex noun phrase rather than has a subject–verb–object structure. Moreover, it is frequent that such a complex noun phrase is inserted into another sentence, i.e., parenthesis. Theoretically, noun phrase as an independent sentence is allowed by dependency grammar. A parser can correctly parse a noun phrase as a sentence

Please cite this article in press as: J. Wang et al., A two-level parser for patent claim parsing, Adv. Eng. Informat. (2015), http://dx.doi.org/10.1016/j.aei.2015.01.013