UNIVERSITY OF WATERLOO
Faculty of Mathematics

INFORMATION EXTRACTION FROM TABLE-FORMAT DOCUMENTATION

Zenefits
San Francisco, California

Xi Zhu
ID: 20453988
4A Computer Science
April 18, 2016

MEMORANDUM

To: Usman Ghani

From: Xi Zhu

Date: April 18, 2016

Re: Work Report: Information Extraction From Table-Format Documentation

---

I have prepared the enclosed report, “Information Extraction From Table-Format Documentation”, for my 2B work report. This is my third of four work reports that are required by Co-operative Education Program.

As a developer at Zenefits, I have been working on developing the program that extracts target information from legal documents. This report describes and analyzes different methods that are used in solving this problem.

The Faculty of Mathematics requests that you evaluate this report for technical content and analysis. This report and your evaluation will be submitted to the Math Undergrad Office for evaluation on campus by qualified work report markers.

Sincerely,

Xi Zhu

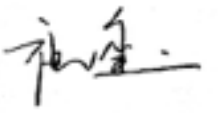

TABLE OF CONTENT

LIST OF FIGURES

Executive Summary

Zenefits has business in insurance industry. A large number of legal documents need to be processed. The goal is to automate this process. This report introduces and analyzes different methods that extract information from table-like documents.

Hardcode solution is straightforward and template-specific. It returns promising result as long as the rules are already defined on input template. Heuristic solution focuses on analyzing table attributes, hence apply to more general situations. Two heuristic solutions, N-gram solution and fuzzy mapping solution, are discussed in second part.

N-gram solution firstly converts data into sequence of terms, then extract information from it with natural language processing method. By doing so, it ignores confusions from the document template and tries to understand context meaning about the terms. Fuzzy mapping solution maps between keys and values based on their spatial relationship, then decide the most possible mapping from candidates by some measurement.

Hardcode solution and heuristic solutions both have pros and cons. Both solutions are worthy to be implemented. Same results from both solutions indicate a reliable answer. Otherwise, at least one solution fails, so user should be notified for a manually check.

1. Introduction

The task is to extract a few required information from a legal document. Information in a document can be generalized as key-value pair, where the key defines what the pair is and value defines how it is. Key is recognized as keywords, and value is recognized as numerical value. However, these information is presented in a semi-structured format: it is presented in a table-like format document, but the table templates vary. Figure 1 shows two example templates. The deductible data is displayed in similar table format, but in different templates. One challenge is there are technically infinity templates. Therefore, the ideal solution should be able to handle as many templates as possible.

Important Questions	Answers	
	In-Network	Out-of-Network
What is the overall deductible?	**$500** Individual / **$1,000** Family	**$1,000** Individual / **$2,000** Family

Benefit Plan Features	Your Cost In-Network	Your Cost Out-Of-Network[1]
Deductible		
Individual/Family	$2,000 / $4,000	$4,000 / $8,000
Out-of-Pocket Maximum (includes copays, coinsurance, and deductibles)		
Individual/Family	$5,000 / $10,000	$15,000 / $30,000

FIGURE 1. TWO SAMPLES OF TABLE IN DIFFERENT FORMAT

Two assumptions are made in solving the problem: first, target information must exist in the document; second, the document can be converted into program readable format while keeping the table-like format.

2. Hardcode Solution

Hardcode solution is the easiest way to find a describable and value information pair. The basic idea is by defining a bunch of rules that detect the values of decide which of them are accepted. These rules are defined based on documents' features observed from practical problem. Here are a few rules in general case: First, if a keyword appears on the left of a value on the same row, they are likely to be related. Second, if keywords of the same type appears on one row without any value, they are likely to be table header and apply to all rows below. [Figure 1]

In program, rules are implemented as nested or parallel condition statements. Hardcode solution is easy to implement and debug, and it works for most cases. However, the difficulty is that defining the rules requires awareness of the feature on observed documents, so the amount of rules can be huge in order to cover all cases and new rules must be added if there is new template. Besides, pdf conversion is not perfect, hardcode solution doesn't have the ability to handle any extraordinary case because of lack of fault tolerance. Therefore, some heuristic solutions are considered. Instead of defining cumbersome rules for each single format, heuristic solutions are built upon some assumptions/truths of general table-like documents. Hence, it tries to understand documents' attributes in a general way, which requires less dependency on specific template. In the following section, two heuristic solutions are discussed.

3. Heuristic Solution

3.1 N-gram Solution and Natural Language Processing

N-gram is widely used in natural language processing, and N-gram solution is trying to understand the document content in a natural language way. It is established on two facts: information is displayed in human reading order (top-to-bottom, left-to-right) across cells, and mapping order between two cells will be conserved [Figure 1(b)].

Following these facts, N-gram solution reads a document as human reads, collects keys and values and stores them in order. Following shows the result on Figure I:

[in-network, out-of-network, deductible, $, individual, $, family, $, individual, family]

[in-network, out-of-network, deductible, individual, family, $, $, $, $]

In the example, even though information is displayed differently in two tables, the orders are consistent. N-gram solution screens out the noise from document templates but keeps the essence of how information is displayed.

The rest work is information extraction[2] from the previous result. Natural language processing techniques work here, such as tokenizing, chucking and tagging[3]. These require defining grammar with regular expression rule. This is similar to hardcode solution, but the content is much simplified by focusing on information itself, hence a lot of hardcoding work on processing document template is saved.

In practice, N-gram method can be used to decide which rule should apply, and eliminate unrelated keyword. (see Figure 3 as described in next section)

3.2 Fuzzy Mapping Solution

Fuzzy mapping solution analyzes the spatial relationship between entities across the document. As the name suggested, the mapping is conducted fuzzily: there are possibly multiple candidate mappings for one value during the process, but the optimal mapping is selected at last based on some measurement on them.

The strategy is to detect every appearance of target describable information and all value information, then calculate the scores for them to be the right match. The calculation of scores relies on the truth that related entities are close or aligned in a table. In particular, consideration is given to following factors on a candidate key-value pair:

- First, horizontal distance reflects the fact that key and value are aligned by column;
- Second, horizontal distance reflects that key and value are most likely to be on same row. If not, they are likely to be very close. In practice, an exponential function $f(x) = e^{-x}$ on the vertical distance is utilized to better fit this property.
- Third, bag-of-word[1] is worthy to be considered to avoid mismatch from above factor when there are multiple values on one row. Example in Figure II.

These factors should be assigned different weight contributing into the final score. This can be represented as a function $f(x) = w*x$. The weights depend on practical experience, hence are much difficult to decide.

In practical, there are two ways of mapping: first is mapping values to the describable information that applies to it. This answers the question "what is the value for this key". However, describable information is recognized by keyword, but keyword can have different meaning depends on context. Figure 2 is an example where keyword "deductible" is not a

describable information. In contrast, value information is in numerical form and has static meaning. Therefore, mapping reversely, which answer the question "which key applies to this value", shows better performance.

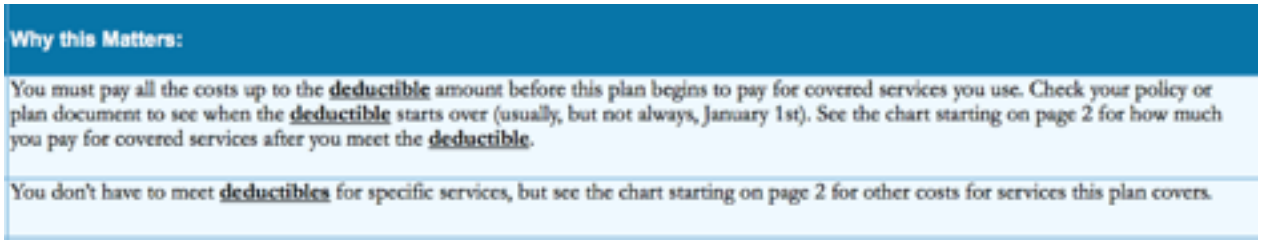

FIGURE 2. EXAMPLE WHERE KEYWORD IS UNRELATED

Compared with N-gram solution, fuzzy mapping solution totally removes hardcode, and hence can be applied to more general situations. But without hardcode, this also means the process is not strictly conducted, and possibly leads to wrong result.

4. Conclusions

Hardcode solution is straightforward, but also demands extensive efforts on pre-defining rules for each single template. It produces promising result as long as the situation is defined. On the other side, it is not robust enough to handle undefined templates and extraordinary situations. Heuristic solutions cover the shortage of hardcode solution by focusing on table's general attributes.

Two heuristic solutions are discussed in this article. N-gram solution firstly converts semi-structured data into unstructured data, then extracts structured data by hardcode rules of grammar. It aims at seizing the content by ignoring the disturb from template. Fuzzy mapping solution

applies to more general situations. The development relies on mathematical models that requires careful design. Since it lacks hardcode, the accuracy of results is not perfectly promised.

To sum up, hardcode and heuristic solutions have their own strengths. Hardcode solution is more reliable while fuzzy solution is more general. It is unfair to exaggerate the advantage of either one over another. Both solutions should be considered in solving this problem.

5. Recommendations

Hardcode solution requires pre-defined rules on different table formats. Modularization should be considered in order to make the program extendable and scalable. For example, develop one base parser, and define rules for each format independently as separate package. All packages follow the same function interface, and are imported during run time. By doing so, it is easy to modify existing package or add new package without affecting others.

The performance of the heuristic solutions heavily depends on the parameters, like the weights assigned to the factors. The choices of these parameters need experience from practical problem. Machine learning techniques, like neural network, are helpful in finding the appropriate parameters.

Hardcode and heuristic solutions have their own strengths. It is worthwhile to implement both algorithms and collect output from both. If two results are different, then it means at least one algorithm fails, so the user should get notified and manually check. Otherwise, it is more likely to believe two same results are correct since there is little chance for two algorithms giving same wrong results at the same time.

References

[1] Steven Bird, 2009. *Natural Language Processing with Python*. 1 Edition. O'Reilly Media.

[2] Yoshikawa, Yuya, Tomoharu Iwata, and Hiroshi Sawada (2014). “Latent Support Measure Machines for Bag-of-Words Data Classification”. In: *Advances in Neural Information Processing Systems*, pages 1961–1969 (page 40).