# Table Extraction from Web Pages Using Conditional Random Fields to Extract Toponym Related Data

**Hayyu' Luthfi Hanifah**[1,2], **Saiful Akbar**[1,3]

[1]School of Electrical Engineering and Informatics, Bandung Institute of Technology

[2]hayyuhanifah52@gmail.com
[3]saiful@informatika.org

**Abstract**. Table is one of the ways to visualize information on web pages. The abundant number of web pages that compose the World Wide Web has been the motivation of information extraction and information retrieval research, including the research for table extraction. Besides, there is a need for a system which is designed to specifically handle location-related information. Based on this background, this research is conducted to provide a way to extract location-related data from web tables so that it can be used in the development of Geographic Information Retrieval (GIR) system. The location-related data will be identified by the toponym (location name). In this research, a rule-based approach with gazetteer is used to recognize toponym from web table. Meanwhile, to extract data from a table, a combination of rule-based approach and statistical-based approach is used. On the statistical-based approach, Conditional Random Fields (CRF) model is used to understand the schema of the table. The result of table extraction is presented on JSON format. If a web table contains toponym, a field will be added on the JSON document to store the toponym values. This field can be used to index the table data in accordance to the toponym, which then can be used in the development of GIR system.

**Keywords -** table extraction, toponym, GIR, CRF, gazetteer

## 1. Introduction

Nowadays, many people fulfill their needs of information by browsing on the internet. There are billions web pages composing the World Wide Web [1]. On these web pages, a lot of information is presented by some visualization techniques, such as text, image, table, audio, and video. The aim of certain visualization technique used on a web page is to ensure people can understand the information easily without considering computer's ability to understand the information. With abundant number of web pages, there is a need for an automatic system (computer-aided) to help people find the needed information effectively and efficiently.

Among several visualization techniques mentioned above, table has attracted researchers to find methods for automatically extracting data from it. Tables use visual structure besides textual form to deliver information. This combination differs data extraction from web tables from data extraction from articles (textual form only). Some researches on table extractions are [2], [3], [4], [5], and [6].

One type of information on the internet is information about certain place. This information is usually identified by the name of the place (toponym). On [7] and [8], it is stated that approximately

12%-15% of search engine queries contain toponym. This becomes a background for research on Geographic Information Retrieval (GIR) [9] focusing on extracting and indexing documents (i.e. web pages) so that people can easily get geo-referenced information from them. In indexing process, toponym recognition is done to identify location-related information in documents. Some researches about toponym recognition are [10] and [11].

In information extraction field, it is common to use some statistical models. These models can be used for some tasks, such as sequence labeling. Some examples of statistical models are Hidden Markov Model (HMM), Maximum Entropy Markov Model (MEMM), and Conditional Random Fields (CRF). Among those models, CRF outperforms the others. So, in this research, CRF will be used on the process of table extraction from web pages.

This paper is organized as follows. Section 1 introduces the underlying problems of this research. Section 2 summarizes some works related to this research. Section 3 explains the proposed method to detect toponym from web table. Section 4 describes the experiment and evaluation for the proposed method. And section 5 states the conclusion of this research.

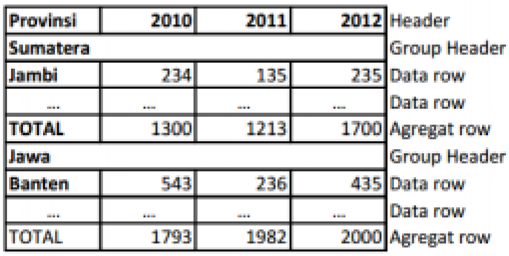
## 2. Related Works

### *2.1. Table Extraction from Web Pages*

There are some researches on web table extraction so far. Generally, those researches use two approaches: rule-based approaches and machine learning approaches.

[2] and [3] attempt to recognize HTML tables using rule-based approach. They construct a rule set to detect tables on web pages containing worth extracted information (genuine tables). This rule set will eliminate HTML tables that are used for formatting/layouting purposes, not for displaying relational data. There are rules about number of rows, number of columns, and number of some HTML elements, such as images, forms, and links in this rule set. In [3], Penn, et al add a rule which specifies that genuine HTML table must not contain other HTML table.

[5] and [6] try to extract data from web tables with different approach. They use CRF model to recognize table schema before extracting data from those tables. Table schema can be understood by knowing the role of each row on a table. [6] categorizes rows on a table as title rows, header rows, group-header rows, data rows, aggregate rows, non-relational rows, and blank rows as showed on Figure 1. To build a CRF model, some features are extracted from each row on the table. After that, each row is labeled coresponding to its role. Then, this feature set and labels are used for training CRF model.



| Provinsi | 2010 | 2011 | 2012 | Header |
|---|---|---|---|---|
| Sumatera | | | | Group Header |
| Jambi | 234 | 135 | 235 | Data row |
| ... | ... | ... | ... | Data row |
| TOTAL | 1300 | 1213 | 1700 | Agregat row |
| Jawa | | | | Group Header |
| Banten | 543 | 236 | 435 | Data row |
| ... | ... | ... | ... | Data row |
| TOTAL | 1793 | 1982 | 2000 | Agregat row |

**Fig. 1.** Example of table with labeled rows

### *2.2. Toponym Recognition*

In toponym recognition field, many researches are conducted in attempt to recognize toponym from textual corpus (set of sentences). [10] uses a rule-based approach to detect toponym on Italian text while [11] combines rule-based and statistical approach to recognize toponym in social media.

The rules which are used on [10] take advantage of some prepositions (e.g. ‘from’, ‘to’) and adjectives (e.g. ‘near’) to detect toponym in a sentence. Then, a dictionary is used to ensure whether the word/phrase after those prpositions/adjectives is a toponym.

In [11], toponym recognition is done by the help of CRF model. Although the use of CRF model is considered as statistical approach, Sagcan & Kagaroz use a rule (i.e. regex) as a feature to build the model. So, it can be said that this is a hybrid approach.

### *2.3. Conditional Random Fields*

Conditional Random Fields (CRF) is one of probabilistic model which is often used on sequence labeling tasks (e.g. part-of-speech tagging, named entity recognition, etc). Compared to other probabilistic models, CRF is better than HMM because of its ability to take more complex feature set on model building [12] and it can handle the label bias problem which may occur on MEMM.

There are some terms related to CRF model: observation, state, and transition. Observation is something that can be observed from the corpus, something that can be used to formulate the feature set. State is the label and transition is the change from a state to another state.

CRF model tries to find the most probable label sequence for a given observation. To do so, CRF model calculates the probabilities for each lable sequence. There are two components affecting this calculation: feature functions and weights of feature functions. The feature function has four parameters, i.e. current state, previous state, observation sequence, and current position. The weight of feature function will be updated on each iteration to get the best model (model which can predict the label sequence with high accuracy).

Mathematically, CRF model building complies a formula as shown in Equation (1).

$$p(y_{1:N}|x_{1:N}) = \frac{1}{Y} exp(\sum_{n=1}^{N} \sum_{i=1}^{F} \lambda_i f_i(y_{n-1}, y_n, x_{1:N}, n)) \quad (1)$$

In that equation, $f$ denotes the feature functions, $\lambda$ is the weight of the feature function, $y$ is the state, $x$ is the observation, and $n$ is the sequence number.

## 3. Toponym Recognition from Web Table

The most common way to create a HTML table is by using table tag (`<table>`). Nevertheless, not all HTML elements with table tag are genuine tables. Table tag is often used for formatting/layouting purposes on web pages (non-genuine table). As stated in the previous section, some rules can be applied to distinguish genuine tables from non-genuine tables. Nevertheless, HTML element that complies those rules is not necessarily a valid relational table. A table will be considered as a valid relational table if it consists of at least a header row and some data rows. Therefore, the schema of a table must be recognized before we extract data from the table.

Meanwhile, the problem we face on toponym recognition from web tables is that table does not usually use full sentence for cell contents. In the other research, toponym recognition is usually done by using the features of adjacent words on a sentence. For web tables, the role of adjacent words on toponym recognition can be replaced by column names, i.e. words/phrases on the header rows.

The proposed method to recognize toponym from HTML table is described in Figure 2 and the following subsections.

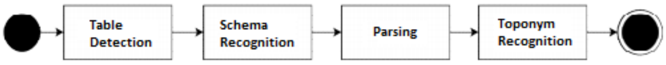


**Fig. 2.** Diagram of the proposed method

### *3.1. Table Detection*

The first step in table recognition is to find HTML element with table tag and complies these rules (adapted from [2] and [3]):

- It does not contain other HTML table. In other words, it must be the innermost HTML table element.
- It has at least 3 non-empty rows and 2 non-empty columns, according to Drasden Web Table Corpus (DWTC) framework. To ensure the majority of rows do not only consist of a cell, a rule regarding the ratio of cell number and row number is added. If the ratio is more than 1.5, the table will be considered as genuine table.

- It does not contain too many images, forms, and links. The maximum limit for images and forms is less than the number of non-empty rows on the table. And the maximum limit for links is less than 50% from the table content.

The limit values in those rules are determined after doing some observations on web table in some websites.

### *3.2. Schema Recognition*

Table schema recognition in this research is adapted from the proposed method in [6]. In brief, the steps of schema recognition are:

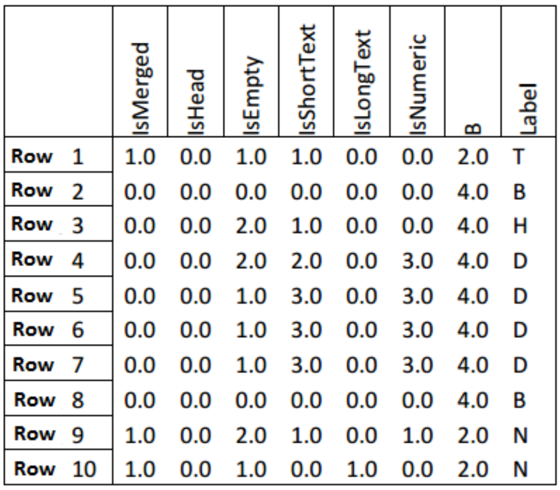
- Extract the attributes of each cell on a row of the table.
- Combine those attributes to be the features of the row.
- Assign a label for the row corresponding to its role on the table (title row, header row, data row, group header row, aggregate row, non-relational row, or blank row).

The cell attributes used in this research involve attributes related to cell layout and cell content. Here is the list of those attributes:

- IsMerged, whether the cell has rowspan or colspan attribute.
- IsHead, whether the cell is an HTML element with head tag (<th>) .
- IsEmpty, whether the cell is empty.
- IsShortText, whether the cell content is a short text (less than 80 characters long).
- IsLongText, whether the cell content is a long text (more than or equal to 80 characters long).
- IsNumeric, whether the cell content is a numeric string.

After extracting these boolean-valued attributes, row features are constructed by applying logarithmic binning [6] to those attributes.

To build a CRF model, all rows in some HTML table elements from table detection module are labeled manually. Figure 3 shows an example of row features and labels for a table that is used as data train. This CRF model will be used to predict the label of each row on a table before the table is parsed to a more structured form.



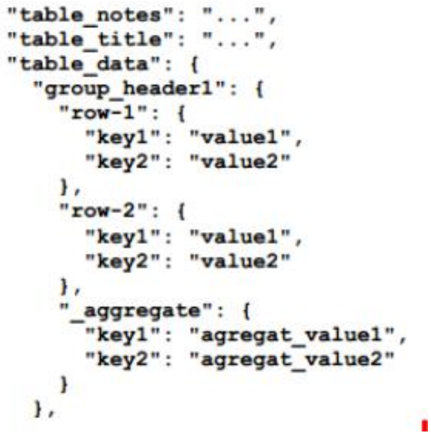
| | IsMerged | IsHead | IsEmpty | IsShortText | IsLongText | IsNumeric | B | Label |
|---|---|---|---|---|---|---|---|---|
| Row 1 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 2.0 | T |
| Row 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | B |
| Row 3 | 0.0 | 0.0 | 2.0 | 1.0 | 0.0 | 0.0 | 4.0 | H |
| Row 4 | 0.0 | 0.0 | 2.0 | 2.0 | 0.0 | 3.0 | 4.0 | D |
| Row 5 | 0.0 | 0.0 | 1.0 | 3.0 | 0.0 | 3.0 | 4.0 | D |
| Row 6 | 0.0 | 0.0 | 1.0 | 3.0 | 0.0 | 3.0 | 4.0 | D |
| Row 7 | 0.0 | 0.0 | 1.0 | 3.0 | 0.0 | 3.0 | 4.0 | D |
| Row 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | B |
| Row 9 | 1.0 | 0.0 | 2.0 | 1.0 | 0.0 | 1.0 | 2.0 | N |
| Row 10 | 1.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 2.0 | N |

**Fig. 3.** Example of row features and row labels

### *3.3. Parsing*

As table extraction method proposed on [2], before parsing, a web table is transformed to matrix first. This can be done by duplicating cells which have `rowspan` or `colspan` attributes. After that, each row is parsed corresponding to its label.

In this research, JSON is chosen as the format of data structure to represent the result of table extraction. The reason behind this choice lies in the flexibility of JSON usage for information retrieval system. Figure 4 shows the format of table extraction result.



```
"table_notes": "...",
"table_title": "...",
"table_data": {
  "group_header1": {
    "row-1": {
      "key1": "value1",
      "key2": "value2"
    },
    "row-2": {
      "key1": "value1",
      "key2": "value2"
    },
    "_aggregate": {
      "key1": "agregat_value1",
      "key2": "agregat_value2"
    }
  },
```

**Fig. 4.** JSON format of table extraction result

There are three main parts in the JSON: table_notes, table_title, and table_data. Value for table_notes is taken from the content of non-relational rows on a table. Value for table_title is the content of title rows on a table. And values for table_data are composed from the content of group header, header, data, and aggregate rows. Contents of header rows will be the keys on each row (“key1” and “key2” in Figure 4) while contents of data rows will be the value of the corresponding key.

### *3.4. Toponym Recognition*

Toponym recognition from the result of table extraction begins with an examination on table header. This aims to find table column presuably containing toponyms. Such column can be identified by some toponym indicator words, e.g. ‘province’, ‘city’, ‘country’, etc.

If the table has column which presumably contains toponym, all contents of that column will be checked whether they meet the following rules:

- The word/phrase must be an alphabet string (not alphanumeric or numeric string)
- The length of the word/phrase is less than 200 characters.

If those rules are met, each word/phrase in the column will be checked whether it is registered in the gazetteer or not. In this research, the gazetteer is constructed from location names registered in GeoNames. Because of the abundant number of toponyms in GeoNames, a tool having feature for text searching is used. We choose Elasticsearh over other tools because its text search feature supports fuzzy matching. The requirement for fuzzy matching is caused by the possibility of different spellings between toponym on a web page and toponym on the gazetteer (although they refer to the same location).

If the table does not have column which presumably contains toponym, then each cell content on the header rows which is an alphabet string and less than 200 characters in length will be checked on the gazetteer. This step is to handle a table which has toponym on its header.

## 4. Experiments and Evaluations

There are two experiments in this research: experiment of CRF model building and experiment of gazetteer usage to recognize toponym on table. The building of CRF model is done several times with different combinations of parameters to obtain the best model. Meanwhile, the experiment of gazetteer usage is done by trying some settings on Elasticsearch to obtain the best configuration which can recognize toponym with high accuracy.

In addition to those experiments, this section will explain the tests performed on table detection module and parsing module.

### *4.1. CRF Model Building*

The scenario for CRF model building experiment is described as follows:

- This experiment uses two types of dataset. The first dataset contains both valid tables and invalid tables. The second dataset only contains valid tables (tables which have at least header and data rows).
- The maximum numbers of iterations in the model building are 100, 200, 250, 300, 350, 400, and 500.
- In the model building process, it can be set whether it will take unseen transitions into account or not. Unseen transition is the transition between states/labels that does not exist in the dataset.

CRF models from this experiment will be evaluated by per label accuracy (especially H and D label accuray) and by full table accuracy. Full table accuracy is associated with the ability of CRF model to predict the label of each row on a table precisely.

Table I describes the statistics of dataset for this experiment.

TABLE I. STATISTICS OF THE DATASET

| **Dataset** | **Table** | **Labels** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **T** | **H** | **G** | **D** | **A** | **N** | **B** |
| Valid + invalid tables (mixed) | 160 | 24 | 80 | 9 | 1243 | 27 | 101 | 120 |
| Valid tables | 145 | 18 | 56 | 16 | 781 | 13 | 69 | 96 |

From this experiment, it can be concluded that the best CRF model is obtained from the model building process with 200-300 iterations and without taking unseen transitions into account. Per label F1-score for this CRF model reaches 0.88. In terms of dataset usage, CRF model which is built by a mixed dataset will yield a better full table accuracy when it is used to predict invalid tables.

### *4.2. Toponym Recognition using Gazetteer*

There are three fields on GeoNames which are used to store the names of a location, i.e. name, asciiname, and alternatenames. Name and asciiname fields contain a toponym. Meanwhile alternatenames field may contain more than one toponym.

To check whether a word/phrase is registered in the gazetteer, the text search feature in Elasticsearch is used. In this feature, there are some configurations which can be set to obtain the optimum look up results. Generally, these configurations are related to mapping the data (index structure) and query structures used for searching toponym on Elasticsearch.

There are three kinds of mapping scenarios in this experiment:

- Mapping-1 indexes the three toponym fields from GeoNames into three different fields on Elasticsearch. For name and asciiname fields, not_analyzed index type is used. This type will index the words on a field without analyzing (e.g. do tokenization) those words.
- Mapping-2 indexes the three toponym fields into three different fields and uses `analyzed` index type for all fields.
- Mapping-3 indexes the three toponym fields into one field and uses analyzed index type for the field.

Meanwhile the query structures used in this experiment are the combination of match query, bool query, and fuzziness parameter provided by Elasticsearch. Generally, there are two types of queries, i.e. match (or multi_match) queries and bool queries. The difference lays in the score calculation. Bool query calculates the score in accordance with the type of its subqueries (`must`, `should`, or `must_not`).

From this experiment, the best configuration for toponym recognition using gazetteer is to use bool query for searching on Mapping-1. This configuration can recognize the toponym on the web tables with decent accuracy. The bool query has `must` subquery and `should` subquery. Combination of these subqueries can be used to do fuzzy text searching but it will give a higher score for toponym which exactly matched to the query.

### *4.3. Table Detector Testing*

The rules which are mentioned in section III.A are effective enough to detect genuine web tables. From 3684 HTML elements with table tag on BPS web pages (http://bps.go.id), there are 942 elements that comply those rules. The remainder 2742 elements are not detected as genuine tables and will not be processed any further. These non-genuine tables are HTML elements with tabel tag which are used for form layouting or to display a list of images and links.

### *4.4. Parser Testing*

Before parsing a web table to JSON format, the validity of the table is checked first. A table must have at least header and data rows. The validity of a table is checked using a Finite State Machine (FSM) as shown in Figure 5.

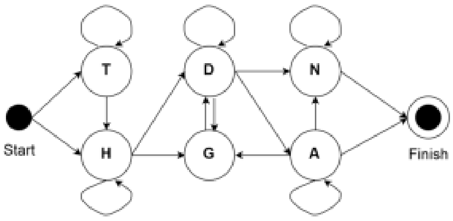


**Fig. 5.** FSM to check table validity

From 160 manually labeled web tables, there are 20 tables which cannot reach the finish state of this FSM (e.g. tables which only contain non-relational rows, tables that do not have header rows).

From this testing, it is also known that there is a small amount of valid tables which are not parsed to JSON perfectly (corresponding to the designed format described in section III.C). This mostly because of the vagueness of data group boundaries (on a table that has group header or aggregate rows).

## 5. Conclusion

Table extraction from web pages can be done by combining the rule-based approach and statistical-based approach. Rule-based approach is used to distinguish genuine tables from non-genuine tables. Meanwhile the statistical-based approach is used to recognize the schema of a table before that table is parsed to a more structured format. Table schema recognition can be done by using CRF model to predict the role of each row on a table. In this experiment, the best CRF model yields 0.88 F1-score.

Toponym recognition on the extracted data from a table can be done by implementing a rule-based approach and using a gazetteer. The first step to recognize toponym on a table is to check the table header. If there is a column presumably containing toponym, the values of that column will be checked in the gazetteer. If such column cannot be detected from the table header, then an examination is conducted to detect toponym in the header itself. After the toponym on a table can be recognized, the data of that table can be indexed by this toponym and can be used in future development of GIR system.

This research only tries to handle web tables which are made by using HTML table tag. It is possible that a web table is not made by using HTML table tag, even though it is rare. Hence, it will be better if the rules and features, which are used on table extraction, are defined more generally in the future works so that they can detect and extract all kinds of web tables. Besides, an experiment that involves many more toponyms is needed to test the toponym recognition module (this research uses location names on Indonesia only).

## References

[1] D. Yadav. (2010). Users Search Trends on WWW and Their Analysis. Proceeding IITM '10 Proc. First Int. Conf. Intell. Interact. Technol. Multimed. , pp. 59-66.

[2] H. Chen, S. Tsai, J. Tsai. (2000). Mining Tables from Large Scale HTML Texts. COLING'00 Proceedings of the 18th conference on Computational linguistics – Volume 1, page 166-172.

[3] G. Penn, J. Hu, H. Luo, R. McDonald. (2001). Flexible Web Document Analysis for Delivery to

Narrow Bandwidth Devices. *Sixth International Conference on Document Analysis and Recognition Proceedings.*

[4] Y. Wang, J. Hu. (2002). A Machine Learning Based Approach for Table Detection on The Web. *WWW '02 Proceedings of the 11 th international conference on World Wide Web.*

[5] D. Pinto, A. McCallum, X. Wei, W.B. Croft. (2003). Table Extraction Using Conditional Random Fields. SIGIR '03 Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval.

[6] M.D. Adelfio, & H. Samet. (2013). Schema Extraction for Tabular Data on the Web. *Proceedings of the VLDB Endowment, vol. 6 issue 6, page 421-432.*

[7] M. Himmelstein, "Local Search : The Internet Is the Yellow Pages," Computer (Long. Beach. Calif). , vol. 38, no. 2, pp. 26–34, 2005.

[8] M. Sanderson, W. Bank, and J. Kohler, "Analyzing geographic queries," in SIGIR Workshop on Geographic Information Retrieval., 2004, vol. 2, pp. 8–10.

[9] A. Dewandaru, I.S. Suwardi, and S. Akbar. (2015). Evaluation on Geospatial Information Extraction, Retrieval Mining Thematic Maps from Web Source. *Proceedings International Conference on Information and Communication Technology (ICoICT),* 283-288. Nusa Dua.

[10] D. Caruso, R. Giunta, D. Messina, G. Pappalardo, E. Tramontana. (2015). Rule-based Location Extraction from Italian Unstructured Text. *WOA, volume 1382 of CEUR Workshop Proceedings, page 59-64.*

[11] M. Sagcan, P. Karagoz. (2015). Toponym Recognition in Social Media for Estimating the Locations of Events. *IEEE International Conference on Data Mining Workshop*.

[12] X. Zhu. (2007). CS838-1 Advanced NLP: Conditional Random Fields. *Technical report, The University of Wisconsin Madison*.