Automatic Named Entity Set Expansion Using Semantic Rules And Wrappers 
For Unary Relations

Mai-Vu Tran, Tien-Tung Nguyen
Information Systems Dept., University of Engineer and Technology, VNU Hanoi, Vietnam
Email: {vutanmai, tungnt52}@gmail.com

Thanh-Son Nguyen, Hoang-Quynh Le
Information Systems Dept., University of Engineer and Technology, VNU Hanoi, Vietnam
Email: {ant.sonnt, lhquynh}@gmail.com

Abstract—Named Entity Set Expansion is the task from a set given a small number of named entities in same type finds a more complete Named Entity set. Two famous Named Entity expansion systems using the web are Google Sets\(^1\) and Boo!Wa\(^2\). In this paper, we proposed an approach using Vietnamese semantic rule combined with wrappers for unary relations to expand a given set of Vietnamese Named Entities. Our model using the results which are provided by Google search engine in Vietnam, consists of six main phases. We did the experiments for many times to choose the rank method that gave the best results is Page Rank, and mixing coefficient is 0.6. Experimental results showed that the feasible approach with average precision is 96.4% with 5 first entities and 92.6% with 10 first entities.

Keywords—Vietnamese Named Entity, Set Expansion, Wrapper, Semantic Rule.

I. INTRODUCTION

This research area is quite new and receiving more and more attention of researchers. To be aware of this potential research area, in this paper, we propose the model which combines two methods for automatic expanding Named Entity set. One is Boo!Wa!, which used and gave good result. The other, using Vietnamese special characteristics, is based on knowledge of Vietnamese Language Processing [HH09]. Then we use a mixing coefficient to combine the results of two methods to receive the best results. Moreover, Internet is a vast and precious resource; many search engines have been built and have better and better performance. To take advantage of this resource, we use the famous search engine “Google” to support our Named Entity set expansion process.

Both of two methods mentioned above build queries and put them into search engine to retrieve results that contain candidates, then extracting the candidates using some methods. However, the process of building queries is different. The semantic rules-based method utilizes semantic relation on Vietnamese text. By studying semantics relation and Vietnamese grammar rules, we built a set of rules in pattern form then using them to make queries put into search engine. The other method Boo!Wa which used is wrappers for unary relations. It uses the characteristics of semi-structured documents such as list, table, etc. This method does not require any parsing; it finds all contextual patterns in the whole document that maximally matching at least one instance of every seed.

By combining two methods, our model not only takes advantage of the characteristics of semi-structured text, but also utilizes the hidden information in the text with semantic rules.

After having the candidates, we rank them then pick out the best candidate to be new entities. To increase the accuracy, the experiments were repeated several times, but using different ranking methods to choose the method gives the best results, it is Page Rank. Also by repeating experiments, changing mixing coefficient each time, we choose that the best mixing coefficient is 0.6.

The main content of this paper is organized as following. Firstly, we present some related works in section II. Then, in section III and IV, we explain how to build and use semantic relation patterns and wrapper rules to extract candidates. Next, section V presents the proposed models and section VI describes the experiments and results. Finally, section VII is conclusion.

II. RELATED WORK

Following RC.Wang and W.Cohen [WC07], Google Sets uses a proprietary method that has not been published in details. It uses an HTML parser for identifying sub-trees of a parsed web page. For each selected sub-tree, it finds one contextual pattern that maximally matches all of the seeds.

In 2007, RC.Wang and W.Cohen have been proposed Set Expander for Any Language (SEAL) model [WC07]. SEAL is worked by automatically finding semi-structured web pages that contain “lists” of items and the aggregating these “lists” so that the “most promising” items are ranked higher. SEAL is a language-independent system has shown good performance in previously published results. By using only three seeds and the top one hundred documents returned by Google, SEAL achieved 93% in average precision, averaged over 36 datasets from three languages: English, Chinese, and Japanese. They also proposed iterative SEAL (iSEAL) model [WC08] which improves SEAL’s performance by handling unlimited number of supervised seeds. In each iteration, it expands a couple of randomly selected seeds while accumulating statistics from one iteration to another.

The KnowItAll system [EC05] contains a List Extractor (LE) component that is functionally similar to Google Sets. The authors described a number of possible variants of the LE component, but it is not clear which

\(^1\)Automatically create sets of items from a few examples: http://labs.google.com/sets/

\(^2\) Boo!Wa! - A List Extractor for Many Languages: http://boowa.com/
system achieved precisions of 23~79% on four sample problems.

Several researchers propose a set expansion method using free text rather than semi-structured Web documents; for instance, Talukdar [TB06] presents a method for automatically selecting trigger words to mark the beginning of a pattern, which is then used for bootstrapping from free text.

Ghahramani [GH05] illustrates a Bayesian Sets algorithm that solves a particular sub-problem of set expansion, in which candidate sets are given, rather than a corpus of web documents. We intend to compare their ranking method with graph-walks in future experiments.

III. SEMANTIC RULES

To extract information from the non-structural data (normally the main content of website), we use semantic rules to identify similar entities. Currently, we concentrate on some simple relations that can be identified by rules in pattern form, include:

- **Hyponymy (IS-A) relation.** It is a relation between two words in which the meaning of one of the words includes the meaning of the other word.
- **Synonymy-name relation.** It is a relation in which various words have different forms but have the same or nearly the same type of meaning.

An example of these semantic relations shown in Fig.1:

![Figure 1. (a) An example of hyponymy and synonymy-name relation in Vietnamese. (b) This example corresponding in English.](image)

We built a set of Vietnamese semantic rules which based on Vietnamese grammar to identify the new entity. These semantic rules are formed in patterns. They are combined with the seed entities in turn to make queries putting into Google search engine. By this way, we receive new “candidate” to rank in next phrase.

This method obviously depends on language. Because of this dependence, we can take advantage of the hidden relationship among entities in the Internet resources that the method based on wrapper will be described in next section could not do. Therefore, this method will have many significant advantages when applied to Vietnamese data.

After making and putting query into a search engine using seed entity and semantic rule patterns, we receive the results provided by search engine which contain candidate. To extract these candidates, the system applies some simple string-processing algorithms.

For example, with the seed set {sư tê, ngựa vân, nai} (lion, zebra, deer), using the relationship patterns built, the system can make queries put into search engine to crawl data, such as: “Một số * như sư tê” (It means “Some * such as lion”). Assumption that data collected had following sentences: “Một số đông vật như sư tê, nai, ngựa vân” (“Some animals such as lion, deer, zebra”), after string processing, we have new candidates “nai” (“deer”) and “ngựa vân” (“zebra”).

IV. WRAPPER RULES FOR UNARY RELATIONS

This method uses the characteristics of semi-structured documents such as list, table, etc.

Following RC. Wang and W. Cohen [WC07], every wrapper is defined by two character strings, which specify entity to be extracted from a page. These strings are chosen to be maximally-long contexts that bracket at least one occurrence of every seed string on a page. The use of character-level wrapper definitions means that this method is completely language-independent (both written language and the markup language used to annotate the semi-structured text). It does not require any parsing and not even necessary to tokenize the text retrieved by the search engine.

When extracting candidate entities use left and right contexts L and R, we only consider the substrings between L and R which do not contain both L and R. The detailed algorithm for automatic construction of wrappers is described in [WC07]. For example, suppose car makes “ford”, “toyota”, and “nissan” were provided as seeds, wrappers can be automatically constructed for each document by using the proposed wrapper construction algorithm. Table 1 shows the contexts that the algorithm selected for constructing wrappers from ‘curryauto.com’ with the symbol “[…]” representing the placeholder for an extracted entity.

| Wrapper #1: | Extractions: | n-li class="[...]"><a href="http://www. ford, honda, acura, kia, toyota, scion, nissan, buick, pontiac |
| Wrapper #2: | Extractions: | src="/common/logos/[...]/logo-horiz-rgb-lg. dkb.png" alt="" chevrolet, ford, kia, toyota, scion, nissan, pontiac, cadillac, hyundai |
| Wrapper #3: | Extractions: | <span class="dName">Curry [...]</span> chevrolet, ford, honda atlanta, honda, honda yorktown, acura, subaru, chiopepe, subaru, kia, toyota, scion, nissan, buick, pontiac, cadillac |

V. PROPOSED MODEL

_A. Mixing coefficient of semantic rules and wrapper rules_

The system uses a combination of two methods semantic rules and wrapper to extract candidate entities, but how to combine this two methods most effective?
Through research and experiments, we propose using a mixing coefficient $\alpha$ to combine the results provided by two methods. Its usage is described in formula (5.1)

$$S(c_i) = \alpha S_w(c_i) + (1 - \alpha) S_s(c_i)$$ (5.1)

*With:*
- $S(c_i)$: collective weight of candidate $i$.
- $S_w(c_i)$: weight of candidate $i$, extracted by using wrapper rules, calculated by certain ranking method.
- $S_s(c_i)$: weight of candidate $i$, extracted by using semantic rules, calculated by certain ranking method.
- $\alpha$: mixing coefficient. It expresses the contribution of each component results in the final result. It is determined through a process of several experiments.

Through experiments, we choose the mixing coefficient $\alpha = 0.6$

**B. Proposed Model**

The proposed model shows in Figure 2. It has six main phases, phase 1 and 2 are based on wrapper method, phase 3 and 4 use semantic rule-based method, phase 5 and 6 pick up new entities.

1) **Phase 1 - Fetcher 1:**
- Input: Seed Entity set, including two to four seed entities.
- Output: Websites contain initial Seed Entity set.
- Processing: Make query contain the seed set and put into search engine, then select $m$ first link returned by search engine (select $m = 100$) to download.

2) **Phase 2 - Extractor 1:**
- Input: Websites downloaded from Fetcher 1 and Seed Entity set.
- Output: Candidate entities.
- Processing: Using website downloaded and the initial seed entity set, applying method for extracting wrapper described in section 4 to find out wrapper rules. After that, using these rules to extract candidate entities (the method was described in section 4).

3) **Phase 3 - Fetcher 2:**
- Input: Seed Entity set and semantic rules in pattern form.
- Output: Websites contain initial Seed Entity set.
- Processing: Using seed set and semantic patterns to make queries and putting into search engine, then selecting $m$ first link returned by search engine (select $m = 100$) to download.

4) **Phase 4 - Extractor 2:**
- Input: Websites downloaded from Fetcher 2, Seed Entity set and semantic patterns.
- Output: Candidates.
- Processing: Pre-processing used JvnTextPro tool (Remove noise and html code, extract main content and sentence segment). After that, extracting candidate entities using semantic patterns and string processing.

5) **Phase 5 - Ranker:**
- Input: The candidate entities extracted by phases Extractor 1 and Extractor 2.
- Output: List of candidate entities has been ranked.
- Processing: Using one of these ranking method:
  - Extracted Frequency: Based on the number of entity that appears in the extractor process.
  - Wrapper length: Just applying to entities extracted by wrapper rules. RC.Wang and W.Cohen [WC08] observed that longer strings are generally better:

$$\log \text{score}(x) = \sum_{j \in \text{extracts} x} \log (\text{length}(w_j))$$ (5.2)

Where $w_j$ is the $j^{th}$'s wrapper composed of a pair of left and right contextual strings, and the function length returns the sum of the character lengths of those pair of string in $w_j$.

---

Page Rank: The class of entities will be calculated which are based on the class of websites. It was extracted. Class of these sites were calculated, according to the Page Rank algorithm (formula (5.3)):

\[ R(u) = c \sum_{v \in \mathcal{B}_u} \frac{R(v)}{N_v} \]  

(5.3)

Where \( u \) is a vertex of the graph (a graph showing the relationship between documents, wrappers and candidates have been extracted); \( \mathcal{B}_u \) is the set of vertices that point to \( u \); \( \mathcal{B}_u \) is the set of vertices that point to \( u \). \( N_u = | \mathcal{F}_u | \) is the number of links from \( u \) and \( c \) is a coefficient used to standardize.

6) Phase 6- Entities Choosing:
- Input: Ranked candidates.
- Output: New entities set similar to intitial seed entity set.
- Processing: Selecting \( k \) first entities in list of ranked candidates (\( k \) is selected in the experimental process).

VI. EXPERIMENTAL RESULTS

Based on 100 Entity sets in English David Nadeau [DN07] proposed, we built 100 Vietnamese seed Entity sets for experiments.

Table 2. Some experimental data

<table>
<thead>
<tr>
<th>No.</th>
<th>Type in English</th>
<th>Type in Vietnamese</th>
<th>Seeds in English</th>
<th>Seeds in Vietnamese</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>colour</td>
<td>màu xanh</td>
<td>90.2%</td>
<td>90.2%</td>
</tr>
<tr>
<td>1</td>
<td>colour</td>
<td>màu đỏ</td>
<td>90.2%</td>
<td>82.7%</td>
</tr>
<tr>
<td>1</td>
<td>colour</td>
<td>màu trắng</td>
<td>90.2%</td>
<td>74.3%</td>
</tr>
<tr>
<td>2</td>
<td>city</td>
<td>thành phố</td>
<td>90.2%</td>
<td>90.2%</td>
</tr>
<tr>
<td>2</td>
<td>city</td>
<td>hải nội</td>
<td>82.7%</td>
<td>82.7%</td>
</tr>
<tr>
<td>2</td>
<td>city</td>
<td>hải Phòng</td>
<td>74.3%</td>
<td>74.3%</td>
</tr>
<tr>
<td>3</td>
<td>hospital</td>
<td>bệnh viện</td>
<td>90.2%</td>
<td>90.2%</td>
</tr>
<tr>
<td>3</td>
<td>hospital</td>
<td>bệnh mát, viện đa khoa</td>
<td>82.7%</td>
<td>82.7%</td>
</tr>
<tr>
<td>3</td>
<td>hospital</td>
<td>bệnh mát, viện đa khoa</td>
<td>74.3%</td>
<td>74.3%</td>
</tr>
</tbody>
</table>

A. Experimental result of ranking algorithms for semantic rules

Table 3. The result using semantic rules

<table>
<thead>
<tr>
<th></th>
<th>Top 5 cands</th>
<th>Top 10 cands</th>
<th>Top 15 cands</th>
<th>Top 20 cands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extracted Frequency</td>
<td>90.2%</td>
<td>82.7%</td>
<td>74.3%</td>
<td></td>
</tr>
</tbody>
</table>

The accuracy is 90.2% for candidate entities of highest weight, and 82.7% for candidate entities of highest weight. This result is quite good.

B. Experimental evaluation of ranking algorithms for wrapper rules

Table 4. Comparing the accuracy of ranking algorithms

<table>
<thead>
<tr>
<th></th>
<th>Top 5 cands</th>
<th>Top 10 cands</th>
<th>Top 15 cands</th>
<th>Top 20 cands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrapper Length</td>
<td>89.8%</td>
<td>81.6%</td>
<td>64.9%</td>
<td>48.7%</td>
</tr>
<tr>
<td>Extracted Frequency</td>
<td>92.6%</td>
<td>86.5%</td>
<td>66.3%</td>
<td>53.6%</td>
</tr>
<tr>
<td>PageRank</td>
<td>94.2%</td>
<td>89.7%</td>
<td>69.3%</td>
<td>48.7%</td>
</tr>
</tbody>
</table>

The table above showed that experiments use Page Rank method has the best results with accuracy is 94.2% for first 5 entities, and 89.7% for first 10 entities. Extracted Frequency algorithm has lower accuracy and Wrapper Length algorithm has the lowest accuracy.

C. Experimental evaluation of result combined wrapper and semantic rules

Table 5. The result applied in combination wrapper and semantic rules

<table>
<thead>
<tr>
<th></th>
<th>Top 5 cands</th>
<th>Top 10 cands</th>
<th>Top 15 cands</th>
<th>Top 20 cands</th>
</tr>
</thead>
<tbody>
<tr>
<td>90.4%</td>
<td>92.0%</td>
<td>90.4%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Applying the formula of combination (5.1) with the mixing coefficient \( \alpha = 0.6 \), the accuracy when combining two methods is better than the accuracy when using each method: 96.4% for the first 5 entities found and 92.6% for the first 10 entities found.

The good results obtained confirm that utilizing both characteristics of the semi-structured text and semantic characteristics would provide result better. This is a positive outcome for the problem Set Expansion for Vietnamese.

VII. CONCLUSIONS

Vietnamese is a complex language, especially its specific characteristics like marks and grammar. In this paper, we have chosen two methods using semantic and wrapper rules then combining them with using a mixing coefficient to build a model for automatic expanding Named Entity set. This approach utilizes both characteristics of the semi-structured text and semantic characteristics.

The highest result of experiment is 96.4% for the first 5 new entities and shows some specific advantages for Vietnamese. With these positive results, the system can be used to support the resolution of other problems such as Question Answering system, Ontology construction, Recommendation system, Entity Search engine, etc.

ACKNOWLEDGMENT

This work was partly supported by the research project VNU QG.10.38.

REFERENCES


