

# LEARNING EFFECTIVE SURFACE TEXT PATTERNS FOR INFORMATION EXTRACTION

Gijs Geleijnse      Jan Korst

*Philips Research Laboratories*  
*High Tech Campus 34, 5656 AE Eindhoven, the Netherlands*  
{gijs.geleijnse,jan.korst}@philips.com

## Abstract

When we extract information from the Web using a search engine, we want to formulate effective queries to generate usable search results. We query a combination of a pattern expressing some relation (e.g. *was born in*) with some instance (e.g. *Albert II*) and identify related instances (*Brussels* or *Belgium*) in the excerpts found by the search engine. We present a novel method to identify effective patterns [1]. A pattern is effective, if it links many different instance-pairs in the excerpts found with a search engine. Precision is only one of the criteria to identify the most effective patterns among the candidates found. The learned surface text patterns are applied in an ontology population algorithm.

## 1 Selecting Effective Patterns

We present an algorithm to learn text patterns expressing relations [1]. We query Google<sup>TM</sup> with a training set  $T_{\mathcal{R}}$  of instance pairs that are  $\mathcal{R}$ -related. The training set  $T_{\mathcal{R}}$  should be chosen such that the instance-pairs are typical for relation  $\mathcal{R}$ . We first discover how relation  $\mathcal{R}$  is expressed in natural language texts on the web. For example, one of the patterns that relates instances of *person* and *city* shows to be the pattern ‘*was born in*’. By querying pairs in  $T_{\mathcal{R}}$  we thus obtain a list of patterns that link the related instances. From this list we select the most effective ones.

Earlier work [2], describes the identification of *precise* patterns. However, precision is not the only criterion for a pattern to be effective in information extraction from the web.

We have formulated three criteria for selecting effective relation patterns.

1. The patterns should *frequently* occur on the web, to increase the probability of getting any results when querying the pattern in combination with an instance.
2. The pattern should be *precise*. When we query a pattern in combination with an instance, we want to have many search results containing  $\mathcal{R}$ -related instances.
3. If relation  $\mathcal{R}$  is not functional, the pattern should be *wide-spread*, i.e. among the search results when querying a combination of the pattern and an instance there must be as many distinct  $\mathcal{R}$ -related instances as possible.

We formulate scoring functions based on these criteria. In order to reduce the amount of queries, we use the frequency of patterns in the list found as an indication for their occurrence on the web. We therefore only evaluate the most frequently found patterns. We do so, since the evaluation of precision and wide-spreadness requires additional queries per pattern.

## 2 Using Patterns in Information Extraction from the Web

Having a method to identify effective patterns, we now focus on utilizing these patterns in information extraction from texts found by a search engine. We use an ontology to represent the information extracted.

Suppose we have an ontology  $O$  with classes  $(c_1, c_2, \dots)$  (e.g. *Actor*) and corresponding instance sets  $(I_1, I_2, \dots)$  (e.g. *Jean-Claude van Damme* is an instance of *Actor*). On these classes, relations  $\mathcal{R}_{(i,j)}$  are defined, with  $i$  and  $j$  the index number of the classes. The non-empty sets  $T_{(i,j)}$  contain the training set of instance-pairs of the relations  $\mathcal{R}_{(i,j)}$ . For example the instance pair (*Tom Cruise*, *Top Gun*) can be part of the training set to learn patterns for the relation *acts in*.

We combine instances with learned patterns into queries. From the retrieved excerpts, we extract related instances. In this way, we simultaneously extract instances of some class and instance-pairs of some relation.

- **Step 1:** Select a relation  $\mathcal{R}_{(i,j)}$ , and an instance  $v$  from either  $I_i$  or  $I_j$  such that there exists at least one pattern expressing  $\mathcal{R}_{(i,j)}$  we have not yet queried in combination with  $v$ .
- **Step 2:** Construct queries using the patterns with  $v$  and send these queries to Google.
- **Step 3:** Extract instances from the excerpts.
- **Step 4:** Add the newly found instances to the corresponding instance set and add the instance-pairs found (thus with  $v$ ) to  $T_{(i,j)}$ .
- **Step 5:** If there exists an instance that we can use to formulate new queries, then repeat the procedure.  
Else, learn new patterns using the extracted instance-pairs and then repeat the procedure.

Note that instances of class  $c_x$  learned using the algorithm applied on relation  $\mathcal{R}_{(x,y)}$  can be used as input for the algorithm applied to some relation  $\mathcal{R}_{(x,z)}$  to populate the sets  $I_z$  and  $T_{(x,z)}$ .

We recognize instances in the Google excerpts using regular expressions. For example, two or three capitalized words describe person names. We check the extracted terms by querying the term in combination with a class-instance relation pattern. For example, if we are interested in movies and we have extracted *The Godfather*, we check whether the number of hits to the query “*The movie The Godfather is*” exceeds a given threshold.

### 3 Conclusion

Our main contributions are the following.

1. We have developed a method to effectively access relevant web data using a search engine by querying combinations of learned patterns and instances.
2. We use a double bootstrapping mechanism to extract information from the web. On the one hand we use newly identified instances to retrieve other instances, on the other hand we learn new relation patterns by adding learned instance pairs to the training set.

A first experiment, the identification of hyponym patterns, showed that the patterns identified indeed intuitively reflect the relation considered. Moreover, we have generated a ranked list of hyponym patterns. An experiment with an ontology describing restaurants and their locations (countries) illustrated that a small training set suffices to learn effective patterns and populate an ontology with good precision (80%) and recall (85%) providing the countries where Burger King is located. The algorithm performs well with respect to recall of the instances found. The identification of the instances however is open to improvement, since the additional check does not filter out all falsely identified candidate instances.

### References

- [1] Gijs Geleijnse and Jan Korst. Learning effective surface text patterns for information extraction. In *Proceedings of the EACL 2006 workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 1–8, Trento, Italy, April 2006. <http://acl.ldc.upenn.edu/W/W06/W06-2201.pdf>.
- [2] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 41–47, Philadelphia, PA, 2002.