Exploiting Redundancy for Pattern-based Relation Instantiation using tOKo

Viktor de Boer
Department of Computer Science, Web & Media,
Vrije Universiteit Amsterdam, the Netherlands
v.de.boer@cs.vu.nl

Maarten W. van Someren
Informatics Institute, Universiteit van
Amsterdam, the Netherlands
m.w.vansomeren@uva.nl

Bob J. Wielinga
Department of Computer Science, Web & Media,
Vrije Universiteit Amsterdam, the Netherlands
bj.wielinga@few.vu.nl

Anjo A. Anjewierden
Behavioural Science IST, University of Twente,
the Netherlands
a.a.anjewierden@gw.utwente.nl

1. INTRODUCTION

The Semantic Web calls for semi-automatic methods to learn, populate and enrich ontologies. In this work, we present a method for the extraction of domain-specific relations between instances (*relation instantiation*). The method uses hand-crafted extraction patterns which are executed on a text corpus using the tOKo text analysis tool[Anjewierden, 2006]. Additionaly, the extracted candidate relation instances can be filtered in a post-processing phase by using domain- and task-specific background knowledge.

The tOKo pattern language allows for patterns that include references to semantic classes. This allows for a wider variety of generality of the patterns (cf. [Califf and Mooney, 2003]). When very specific patterns are used, we can expect a high precision but a relatively low recall. If more general patterns are used, recall is expected to go up. This will negatively affect precision, but if we exploit the redundancy of the relation instances in the corpus by putting a threshold on the frequency of pattern matches, we can compensate for this loss in precision. Especially for extraction tasks where the expected recall is very low, boosting the recall is very beneficial to increase the overall performance when measured in terms of the F-measure. In this paper, we show how exploiting the redundancy in this way improves performance of the method. An extended version of this work can be found in [de Boer, 2010].

2. TASK AND METHOD

We define the task of relation instantiation from a corpus as follows: Given two classes C_i and C_j in a partly populated ontology, with sets of instances I_i and I_j and given a relation $R : C_i \times C_j$, identify for an instance $i \in I_i$ all instances $j \in I_j$ such that the relation $R(i, j)$ holds given the information in the corpus. In this work we will discuss both the situation where all elements of I_i or I_j are known as well as the situation where we discover new instances of the class C_i or C_j.

The tOKo tool and its pattern language The open source tool tOKo [Anjewierden, 2006] has a large number of interactive text analysis and ontology engineering functionalities that can be accessed through a user-interface or through a Prolog API. The tool also provides a powerful pattern search functionality. The pattern language includes 'standard' syntactic abstractions such as matches on exact words, lemma's, word classes, numbers, punctuations, special characters, etc. TOKo also allows the use of populated ontology concepts in these patterns (denoted by square brackets) where all term instances of that class are matched in a text corpus. For example, the pattern *I ate an* $[fruit]$ matches the phrases "'I ate an apple"' and "'I ate an orange"', assuming that the class *fruit* is populated with these instances.

Relation Instantiation using patterns. The input for the method is a specific relation R and the related concepts C_i and C_j from the ontology and any instances I_i and I_j from the knowledge base. In the first step, we create a corpus for the task using the labels from the concepts and the relation. These are presented to the Google search engine. The first N pages are retrieved to form the corpus. On this corpus, a manually constructed tOKo extraction pattern is executed. A pattern query consists of three sub-patterns corresponding to the concept C_i, the relation R and the concept C_j respectively. The sub-patterns for C_i and C_j are constructed using tOKo's sub-concept retrieval feature. If the task also includes populating one of the classes, the expected word class can be used to match potential candidate instances. The generality of a relation instantiation pattern can be adjusted by choosing more general pattern constructs for the subpattern for R (*I verb an* $[fruit]$ is more general than *I eat an* $[fruit]$.

Next, the specific phrases that are the result of the Information Extraction phase are converted to RDF triples by mapping the three different sub-phrases to the corresponding instances of C_i, R and C_j respectively using the tOKo API. Synonyms, misspellings and abbreviations are mapped to single instances. The output is a list of *candidate relation instances* ordered by their associated frequencies in the corpus. In our experiments, we evaluate the performance of the method for various experiments by putting a threshold

EKAW 2010 11-15 October 2010 - Lisbon, Portugal
.

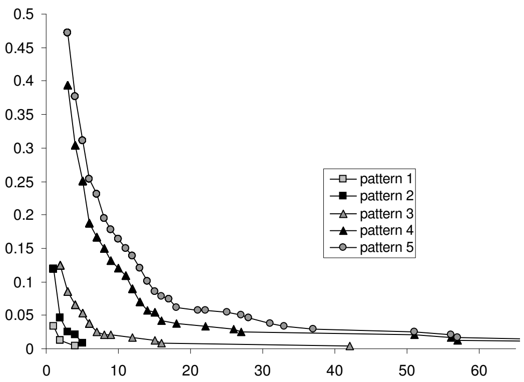

Figure 1: F-measure values for the five patterns for Experiment 1.

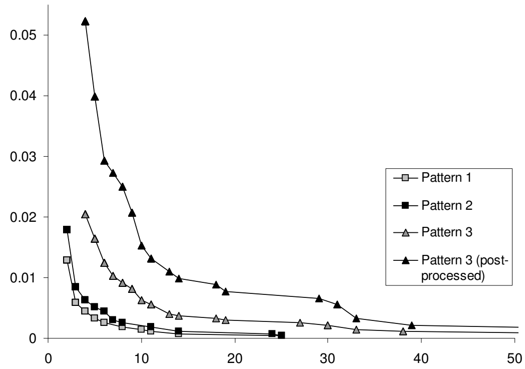

Figure 2: F-measure values threshold values for Experiment 2, including post-processed results

on the frequency of the candidate relations.

Background knowledge about the classes C_i and C_j and the relation R can be used to improve the performance of the method and to reduce unwanted redundancy in the candidate relation instances. In Section 4 we give an example.

3. EXP. 1: ROMAN GODS

For this experiment, we constructed an extremely simple 'ontology' consisting of two classes: GODS:ROMAN GOD populated with 259 instances and GODS:DOMAIN (unpopulated), with the relation GODS:IS_GOD_OF between the two. We constructed a corpus by extracting from the web the first 1000 pages resulting from the google query *'Roman +God +Goddess'*. We constructed the following 5 patterns of varying generality:

1: $[Roman\_god]$ *is the* $\{god|\}$ *of* $\langle noun\rangle$
2: $[Roman\_god]$ $*$ *the* $\{god|goddess\}$ *of* $\langle noun\rangle$
3: $[Roman\_god]\{|\_\}$ ...10 *the god|goddess of* $\langle noun\rangle$
4: $[Roman\_god]\{|\_\}$ ...10 *god|goddess of* $\langle noun\rangle$
5: $[Roman\_god]\{|\_\}$ ...10 *god|goddess* ...10 $\langle noun\rangle$

The results show the expected tradeoff between precision and recall depending on the generality of the pattern. To show the combined performance, we plotted the harmonic mean of both precision and recall, the F-measure against the threshold value in Figure 3 for all patterns. This figure shows that using a general pattern and a threshold on the frequency is preferable to using specific patterns. This is the case when a large number of relation instances are to be found and recall is the main contributor to the F-measure.

4. EXP. 2: ARTISTS' BIRTH PLACES

To test the performance of the method in a second domain and to show the post-processing step, we attempt a second relation instantiation task where the goal is to extract instances of the relation PAINTER BORN_IN BIRTHPLACE the subject and object classes were populated with 1808 European painters and 47.000 European birthplaces. Three patterns of varying generality were constructed:

1: $[painter]$ *was* $(born)$ *in* $[place]$
2: $[painter]$ *was* $(born)$ $\{in|at\}$ ...10 $[place]$
3: $[painter]$ ...10 $(born)$ ...20 $[place]$

We manually evaluated the results. Again, more general patterns lead to higher recall, while more specific patterns lead to higher precision. In Figure 2 we plot the values for the F-measure for different threshold values. We here also observe that the value of the F-measure for more general patterns is higher than that of more specific patterns for all threshold values that are evaluated. Thus we can conclude that if the harmonic mean is used as an evaluation criterion, using more general patterns results in a better performance.

We also performed a postprocessing step on this data where we exploit the hierarchical structure of the geographic places in the TGN[1]. Candidate relation instances that are hierachically equivalent are mapped to a single relation, where occurrence frequencies are summed. Figure 2 also shows the results of the evaluation this postprocessed candidate relation instance set, which shows a significantly higher F-measure value.

5. CONCLUSIONS

We have shown the working of the various steps of the extraction method and the performance-boosting effect of the post-processing step. In both experiments, the values of the F1-measures are largely determined by the relatively low recall values. If the corpus is finite and the list of instances to be found is large enough this data sparseness will occur for all patterns. In that case, using more general patterns in combination with a threshold, thereby exploiting the redundancy will have a beneficial influence on the performance. For relation instantiation tasks, where semi-automatic methods are most needed due to the large number of target relation instances, using redundancy will be beneficial.

6. REFERENCES

[Anjewierden, 2006] Anjewierden, A. (2006). toko and sigmund: text analysis support for ontology development and social research. http://www.toko-sigmund.org.

[Califf and Mooney, 2003] Califf, M. E. and Mooney, R. J. (2003). Bottom-up relational learning of pattern matching rules for information extraction. *J. Mach. Learn. Res.*, 4:177–210.

[de Boer, 2010] de Boer, V. (2010). *Ontology Enrichment from Heterogeneous Sources on the Web*. PhD thesis, Universiteit van Amsterdam.

[1] Getty's Thesaurus of Geographic Names