Acquiring Relational Patterns from Wikipedia: A Case Study

Rahmad Mahendra[1,2], Lilian Wanzare[1,2],
Bernardo Magnini[1], Raffaella Bernardi[3], Alberto Lavelli[1]

[1]Human Language Technology, Fondazione Bruno Kessler
via Sommarive, 18, I-38050 Povo (Trento), Italy
[2]Faculty of Computer Science, Free University of Bozen - Bolzano
Universitätsplatz 1 - piazza Università, 1-39100 Bozen-Bolzano, Italy
[3]Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento
via Sommarive, 14, I-38123 Povo (Trento), Italy
{mahendra,magnini,lavelli}@fbk.eu, liliwanzie@gmail.com, bernardi@disi.unitn.it

Abstract

This paper proposes the automatic acquisition of binary relational patterns (i.e. portions of text expressing a relation between two entities) from Wikipedia. There are a few advantages behind the use of Wikipedia: (i) relations are represented in the DBpedia ontology, which provides a repository of concepts to be used as semantic variables within patterns; (ii) most of the DBpedia relations appear in Wikipedia infoboxes, and are likely to be expressed in the corresponding pages, which increases the effectiveness of the extraction process; (iii) finally, as Wikipedia has pages in several languages, this opens the possibility for the acquisition of multilingual relational patterns. We show that a relatively simple methodology performs quite well on a set of selected DBpedia relations, for which a benchmark has been realized.

Keywords: Relational patterns, Wikipedia, DBpedia, Ontology Population, Question Answering

1. Introduction

Relational patterns represent linguistic variations through which a certain relation among entities is expressed in a certain language. As an example, the pattern:

[<Country>, whose capital is <City>]

might be used to express the relation Has_Capital[1] between an instance of a Country, like Italy and an instance of a City, like Rome, in English.

Relational patterns are of utmost importance for a number of applications, including Information Extraction from text, Question Answering and Knowledge Base Population, as they facilitate the search in textual material, allowing to retrieve relevant information. However, the ability to automatically collect relational patterns on a large scale for a high number of relations and with a good quality is currently limited by two main factors: (i) as state-of-the-art Information Extraction techniques are based on automatic learning from training data, currently available for very few relations, the potential coverage for such approaches is quite limited; (ii) on the other hand, non supervised approaches suffer for lack of precision, as potential relational patterns still contain too much noise to ensure good performance.

The working hypothesis of this paper is that relational patterns can be automatically acquired from Wikipedia, on a significant scale (i.e. 1.200 relations), and with high quality. In fact, Wikipedia constitutes a notable exception with respect to other sources, as it contains both structured information (i.e. the so called "infoboxes") and non-structured information (i.e. the textual descriptions).

In this paper we take advantage of the redundancy of information in Wikipedia, where the same relation about an entity is expressed both in the infobox and in the textual description. This redundancy, together with the fact that entities are marked in the text, makes it easier the task of collecting relational patterns. More specifically, there are three advantages behind the use of Wikipedia: (i) relations are represented in the DBpedia ontology, which provides a repository of concepts to be used as semantic variables within patterns; (ii) most of the relations appears in Wikipedia infoboxes, and are expressed in the corresponding pages with high probability, which increases the effectiveness of the extraction process; (iii) finally, as Wikipedia has pages in several languages, this opens the possibility for the acquisition of multilingual relational patterns.

We aim at collecting textual patterns that express relations which are both formally represented in DBpedia (there are about 1,200 such relations) and that are used in Wikipedia infoboxes. Patterns are supposed to contain both variables (i.e. semantic concepts for the domain and for the range of the relation) and textual material (see the example above).

We present the results obtained experimenting pattern acquisition for a set of pilot relations, selected according to a number of criteria. We show a three-phase procedure: first a Wikipedia dump has been pre-processed; second, sentences containing a target relation are extracted, and finally the relational patterns from the sentences are extracted. Precision and recall of the last two phases are computed against a benchmark of manually annotated relations, which is available as an independent resource.

[1] Throughout the paper we will use courier fonts in order to identify concepts and relations from an ontology.

The result analysis shows that two parameters affect the results of the pattern extraction process: (i) relations whose range is a datatype according to DBPedia (e.g. `birthDate`) need specific processing in order to produce acceptable patterns; (ii) the frequency of the relations in the infoboxes is proved to affect quite significantly the performance, as low frequency relations produce a low number of patterns.

The paper is structured as follows. Section 2 reports some of the related work, both addressing relational pattern acquisition and the use of patterns in Ontology Population and Question Answering. Section 3 describes the acquisition methodology from Wikipedia and DBpedia. Section 4 presents our experimental setting, including the evaluation of the results against a benchmark, which we have realized annotating about one thousand Wikipedia pages.

2. Related Work

There are several notable works on relational patterns acquisition and their use in Information Extraction and Question Answering.

Ravichandran and Hovy (2002) leveraged the usage of surface text patterns for open-domain question answering system. A tagged corpus was built from the web in a bootstrapping process by providing a few hand-tagged examples of each question type. Instances of each question type consist of the question term and the answer term, where the question term usually refers to the subject of the relation (primary entity) and the answer term is the object of the relation. For example, given the `BIRTHYEAR` question type, they select the instance pair "Mozart" and "1756". They submit both terms as queries to the Altavista search engine[2] and consider only the sentences in the top 1,000 documents matching with queries that contain both the question and the answer terms. The longest common matching substrings (computed using suffix trees) for all sentences for each type were considered as patterns. In their experiments, Ravichandran and Hovy tested 6 question types: `BIRTHYEAR`, `INVENTOR`, `DISCOVERER`, `DEFINITION`, `WHY-FAMOUS`, and `LOCATION` on two different input sources, the TREC collection and the web. For evaluation, they measured Mean Reciprocal Rank (MRR) score[3]. The result indicated that the method worked better on the web as input source.

TextRunner (Banko et. al., 2007) and WOE (Wu and Weld, 2010) are two examples of Open Information Extraction (Open IE) systems. Open IE systems usually make a single data-driven pass over their corpus and extract an unbound number of relational tuples without any manual intervention and using an unsupervised learning method to grab a large number of relations from the corpus. While TextRunner runs over millions of web data, WOE was applied on Wikipedia data. WOE runs into two distinguished modes. The former is the same as Text Runner's, i.e. it learns patterns using limited shallow features, like part of speech tags. The latter explores the possibility of exploiting the output of a dependency parser.

WRPA [Vila et al., 2010] extracts paraphrasing patterns of relations from Wikipedia pages. A possible paraphrasing patterns would be

{text}[X]{text}[Y]{text}[Z]{text}

where *X* is the source of relation (author and person in this case) and *Y* is the target (work, birth or death information). In total, four kinds of paraphrasing patterns were examined in WRPA: `DateOfBirth`, `DateOfDeath`, `PlaceOfBirth`, and `Authorship`. WRPA was tested on Spanish and English Wikipedia corpus. Villa et al. (2010) evaluated the top *N* patterns for each paraphrasing and got quite promising (but need to be improved) F-measure score.

3. Methodology

Our basic idea is to exploit the availability of both Wikipedia and DBpedia as seeds to harvest a set of relational patterns. The method includes three phases: (i) pre-processing Wikipedia and DBpedia, in order to extract a text corpus (which we called Wiki-corpus) from Wikipedia and to select the DBpedia relations; (ii) extracting the sentences in the Wikipedia corpus which contain the selected relations; (iii) finally, learning the patterns from the set of sentences.

3.1. Pre-processing Wikipedia/DBpedia

Wikipedia[4] is a free online encyclopedia driven by a collaborative effort of a community of volunteers, covering various multi-domain topics in different languages, whose characteristics make it suitable as a rich lexical semantic resource [Zesch, Gurevych, Mühlhäuser, 2007]. Besides providing unstructured textual descriptions about specific topics, most article pages have structured information, which is presented in the so-called infoboxes. An infobox is a fixed-format table put in the right-hand corner of the articles giving a fact summary about information mentioned in the text. For example, the infobox in the English Wikipedia page "Bolzano" contains geographical information, i.e country, region, province (in which country, region, and province Bolzano is), total area, total population, etc.

The DBpedia[5] knowledge base contains structured information extracted from the Wikipedia infoboxes, providing about one million instances of relations and concepts, including various classes, i.e. `persons`, `places`, `organizations`, `works`, etc. [Bizer, et.al., 2009].

The Wiki-corpus used for our experiments has been created from a Wikipedia dump (January, 15, 2011 version) and consists of the textual part for each Wikipedia page. The names of the relations and the instances for each relation were derived from DBpedia. A relation instance is represented as the triple <*D*, *Rel*, *R*>, where *D* (domain) is the primary entity which is the source of the relation; *R* (range) is the target of the relation and can be either an entity or a datatype attribute; *Rel* is the relationship between the domain and the range of the relation.

[2] http://www.altavista.com

[3] http://en.wikipedia.org/wiki/Mean_reciprocal_rank

[4] http://www.wikipedia.org

[5] http://dbpedia.org/about

3.2. Sentence Extraction

Given a relation instance, the sentence extraction task aims at collecting the sentences in the Wiki-corpus expressing such relation instance. The system tries to find both the domain and range of the relation instance in the sentences. For example, given the following relation instance:

```
<http://dbpedia.org/resource/Forrest_Gump>
<http://dbpedia.org/ontology/writer>
<http://dbpedia.org/resource/Winston_Groom>
```

the sentence extractor module would return the following sentence (1), where both the domain and the range of the `writer` relation are highlighted.

1. ***Forrest Gump** is a 1994 American comedy-drama film based on the 1986 novel of the same name by **Winston Groom.***

However, the system would not return the two following sentences (2 and 3), because either the domain or the range are not explicitly mentioned:

2. *The film was directed by Robert Zemeckis, starring Tom Hanks, Robin Wright, and Gary Sinise.*
3. *The story depicts several decades in the life of **Forrest Gump**, a simple Alabama man who travels across the world, sometimes meeting historical figures, influencing popular culture, and experiencing firsthand historic events of the late 20th century.*

In order to increase the recall of our system, we apply different string matching algorithms, ranging from exact matching to approximate matching (e.g. we try to match not the entire domain or range in the relation instance, but one or more tokens which are parts of the domain or the range).

As for exact string matching, we search precisely the occurrence of the domain or the range name in the sentence. For instance, if we have the domain name *Forrest Gump*, we check whether a sentence contains the string "Forrest Gump" or not. If the sentence cannot be matched by applying exact matching, we relax the matching process. In some cases, the domain or range name include category information. For example, for the page "Gone with the Wind (Film)" and "John Kay (Musician)" we remove the substrings "(Film)" and "(Musician)" and try to match "Gone with the Wind" and "John Kay" within the sentence. Finally, we apply token matching. For example, to check whether a sentence contains the name "George Washington", we try to search the tokens "George" and "Washington" in the sentence independently.

3.3. Pattern Extraction

The next step after sentences extraction is to automatically learn the patterns for a certain relation. Our method is inspired by Ravichandran and Hovy (2002)'s work, as we find the longest common substrings between the extracted sentences for a certain relation using suffix trees. The starting point are sentences where both the domain and the range for a certain relation are identified, as in the following examples for the `BIRTHDATE` relation:

1. **Alex** was born on **July, 17, 1984** in Ottawa, Canada.
2. **Jimmy** (**25-10-1949** - 13 12 1999) was a talented painter in Albania.
3. **Marina** (**Dec, 18, 1889** - Nov, 19 1940) is the second child of Brian Smith.
4. **Josephine** was born in Pennsylvania on **3 May 1990**.
5. **Sabrina**, the daughter of Sir Philip, was born on **August, 14, 1989** in small town near Birmingham.

Before processing the suffix tree, the domain and range of the relation in the text are automatically converted into semantic labels, using the concepts defined in the Dbpedia ontology.

In the above examples, the domain of the `BIRTHDATE` relation is defined in DBpedia as `PERSON`, while the range is defined as `DATE`. Given this information, the corresponding sequence of tokens of the sentence representing the domain are substituted with [`PERSON`], while the range is substituted with [`DATE`]. For instances, sentences number 1 and 2 are transformed into:

1'. [`PERSON`] was born on [`DATE`] in Ottawa, Canada.
2'. [`PERSON`] ([`DATE`] - 13 12 1999) was a talented painter in Albania.

One the sentences are generalized using DBpedia semantic labels, the suffix tree in Figure 1 would be used to represent them.

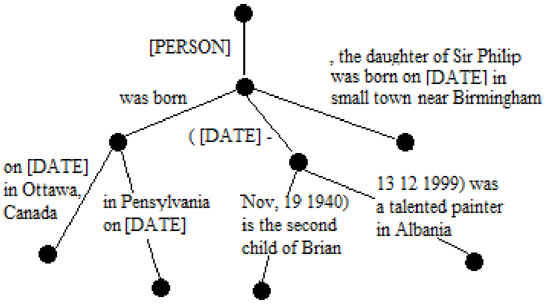
[PERSON]
, the daughter of Sir Philip was born on [DATE] in small town near Birmingham
was born
([DATE] -
on [DATE] in Ottawa, Canada
in Pensylvania on [DATE]
Nov, 19 1940) is the second child of Brian
13 12 1999) was a talented painter in Albania

Fig 1: Pattern Representation using suffix tree

A relational pattern for a certain relation is extracted from the suffix tree when: (i) it contains both the domain and the range for the relation and (ii) it can be derived from the tree by tracing from the root to a specific internal node.

As an example, the most frequent pattern for these examples is

"[`PERSON`] ([`DATE`] - ".

This pattern has size 2 since there are two sentences in the training set matching with it, i.e. sentences 2 and 3. On the other hand, according to our definition, the following pattern.

"[`PERSON`] was born"

is not a valid pattern, as it does not contain the range of the relation.

4. Experiments and Results

For the purposes of the experiments reported in this paper ten DBpedia relations were sampled based on their frequency of mentions (both high and low frequency relations), the level of the domain in DBpedia ontology (both those higher and those lower in the ontology) and whether the range is an entity or a literal property like date, string etc. (both relations with entities and properties). Table 1 shows the relations that were used in the experiments.

Relation	Domain type	Range type	No. of annotated pages	No. of gold-standard sentences
birthDate	Person	Date	290	296
writer	Work	Person	180	135
language	Thing	Language	110	48
NumberOf Students	University	Non negative integer	80	38
color	Thing	String	150	19
Game Engine	Video Game	Thing	80	40
Wine Region	Grape	Wine region	16	88
collection	Museum	String	10	6
isPartOf	Brain	Anatomical Structure	6	10
Launch Date	Space Mission	Date	5	6

Table 1. DBpedia relations used in the experiments.

For each target relation, the corresponding Wikipedia page contains a mention of the relation in the infobox.

4.1. Benchmark

For evaluation purposes a benchmark has been realized annotating about one thousand sentences in the Wiki-corpus containing a mention of the domain and range of the target relations (as given in the infobox or DBpedia resource page)

In total 927 pages were selected and 686 sentences used to build the gold-standard.

The annotation scheme used has been partially adapted from SemEval-2010 task-8[6] annotations. Each target sentence was annotated with the relation name, domain, range and sequence of tokens expressing the relation. The tags used are elaborated below:

- <e1>[DOMAIN]</e1> - Used to tag the entity representing the domain of the relation.
- <e2>[RANGE]</e2> - Used to tag the range value of the relation.
- <rel>[RELATION]</rel> - Used to tag the sequence of tokens expressing the target relation.

As an example, the sentence:

Alex was born on July, 17, 1984 in Ottawa, Canada.

When the `birthDate` relation is considered, is annotated as follows:

<e1>Alex</e1> was born on <e2>July, 17, 1984</e2> in Ottawa, Canada.

Annotation guidelines[7] were built to specify the rules used in the annotation process in handling various issues.

The annotation task was carried out by two annotators and inter-annotator agreement was measured using Dice (Dice, 1945) coefficient given by $2C/(A+B)$, where C is the number of common annotations (i.e. both annotators have identified the same sequence of tokens being tagged or the same sentence); A is the number of entities (or sentences) annotated by the first annotator and B is the number of entities (or sentences) annotated by the second annotator.

Table 2 shows the results of the inter-annotator agreement. The domain tag has the highest average while relation tag has the least because it is challenging to decide the exact sequence of tokens expressing a given relation.

Relation	Dice: Sentence	Dice: Domain	Dice: Range	Dice: Relation
birthDate	0.96	0.93	0.98	0.98
writer	0.84	0.93	0.90	0.89
language	0.78	0.98	0.93	1.00
NumberOfStudents	0.89	0.92	0.96	0.64
color	0.39	0.50	0.50	0.22
gameEngine	0.76	0.78	0.97	0.83
wineRegion	0.63	0.90	0.83	0.61
collection	0.60	0.93	0.36	0.64
isPartOf	0.78	0.67	0.87	0.33
launchDate	0.97	0.93	0.93	0.86
Average	**0.76**	**0.85**	**0.82**	**0.7**

Table 2. Results of inter-annotator agreement.

4.2. Sentence and Pattern Evaluation

Each sentence has a unique ID, so the sentence evaluation simply compares the sentence IDs in the automatically extracted sentences (extracted from the pages used to build the gold-standard) with those in the gold-standard. Both Precision, Recall and F-measure were used to evaluate the automatic sentence extraction algorithm, defined as follows.

For sentences S, α is the number of sentences in the gold-standard, β'' is the total number of sentences extracted by the system and $\beta' = \{S \mid S \in \beta'' \wedge S \in \alpha\}$ i.e. the total number of sentences present both in the automatically extracted corpus and in the gold-standard. Therefore Precision P is calculated as β'/β''. Recall R is calculated as β'/α.

In pattern evaluation, we measured the accuracy of the set of patterns extracted for a given relation R. Let $\beta \in R$, where β has a fixed part, and a variable part representing the place holders for the domain and range of the relation.

[6] http://semeval2.fbk.eu/semeval2.php?location=tasks#T11

[7] The annotation guidelines are available at https://sites.google.com/site/wikipatterns2011/home

Let $\alpha \in \Gamma$, where α is the set of gold-standard sentences belonging to some relation R. Each β is transformed into a regular expression and is matched against all sentences in gold-standard Γ. Let α' be the set of sentences that have been matched by at least one pattern β.

Precision is calculated as the portion of those sentences matched by patterns for some relation R that are also in the gold-standard sentences for the same relation over all sentences extracted by the system i.e. $(\alpha \cap \alpha')/\alpha'$.

Recall is calculated as the portion of those sentences that have been matched by patterns for some relation R that are also in the gold-standard sentences for the same relation over all sentences belonging to the gold-standard sentences α of that relation i.e. $(\alpha \cap \alpha')/\alpha$.

Table 3 shows the evaluation results for each relation. Relation `birthDate` has the highest F1 measure both for sentence and pattern evaluation.

Relation	**Sentence Eval.**			**Pattern Eval.**		
	P	**R**	**F1**	**P**	**R**	**F1**
birthDate	1.00	0.85	0.92	0.73	0.25	0.37
writer	0.37	0.53	0.44	0.49	0.19	0.28
language	0.40	0.40	0.40	0.23	0.46	0.31
NumberOfStudents	0.67	0.21	0.32	0.15	0.11	0.12
color	0.04	0.21	0.06	0.20	0.16	0.18
gameEngine	0.28	0.33	0.30	0.17	0.12	0.14
wineRegion	0.48	0.27	0.35	0.23	0.10	0.14
collection	0.09	0.33	0.14	0.00	0.00	0.00
isPartOf	0.73	0.80	0.76	0.00	0.00	0.00
launchDate	1.00	0.43	0.60	0.00	0.00	0.00
Macro Average	**0.51**	**0.44**	**0.43**	**0.22**	**0.14**	**0.15**

Table 3. Results of sentence and pattern evaluation.

4.3. Discussion

We have applied simple techniques for both sentence extraction and pattern learning: the benefit is that such techniques are almost language independent, allowing a wide application on the several languages maintained by Wikipedia. On the other side, there are a number of linguistic phenomena related to language variability that are not captured by the current algorithms. Sometimes either the domain or the range of the relation can be expressed as completely different terms from those used in the infoboxes, such as synonyms, acronyms, or pronouns, which explains the low recall for some relations in sentence extraction. For the pattern learning task, the current method still cannot capture patterns which involve very long-distance text between the domain and the range. In addition, the 0 results in pattern extraction evaluation reported in Table 3 are for those relations whose extracted patterns did not match any sentence in the gold-standard, due to the low frequency of the relations.

5. Conclusions

We have shown that starting from binary relations defined in DBpedia, we can automatically extract textual patterns for such relations from Wikipedia pages.

As for future work we plan the following extensions of our research: (i) we intend to apply our approach to the full set of Wikipedia relations; (ii) as the current method is language independent, we are in the process to extend it to other languages; (iii) the extracted patterns will be further evaluated and used in shared tasks, like KBP (Knowledge Base Population) and QALD (Question Answering over Linked Data).

Acknowledgements

This work has been partially supported by the LiveMemories (www.livememories.org) project, funded by the Autonomous Province of Trento and by the Erasmus Mundus European Master Program in Language and Communication Technologies Program (http://www.lct-master.org/).

We thank Angelo Basili, student of the University of Trento, for helping developing the annotation guidelines.

References

Banko, M., M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. (2007). Open Information Extraction from the Web. In *Proceeding of the 20th International Joint Conference on Artificial Intelligence* (IJCAI'07), pp. 2670-2676, Hyberabad, India.

Bizer, C., J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and, S. Hellmann. (2009). DBpedia – A Crystallization Point for the Web of Data. in *Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Issue 7, pp 154–165.*

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein. (2001). Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, Chapter 32: String Matching, pp.906–932.

Gusfield, D. (1997). Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Chapter 6: Linear Time construction of Suffix trees, 94–121.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Marco Pennacchiotti, Diarmuid O Seaghdha, Sebastian Pad´o, Lorenza Romano and Stan Szpakowicz. (2010). *SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals.* Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2010).

Ravichandran, D. and E.H. Hovy. (2002). *Learning Surface Text Patterns for a Question Answering System.* Proceedings of the 40th ACL conference, Philadelphia, PA.

Lee R. Dice. (1945). *Measure of the amount of ecological association between species.* Journal of Ecology, 26(3):297-302.

Vila, Marta, Horacio Rodriguez and M. Antonia Marti. (2010). *WRPA: A system for relational paraphrase acquisition from Wikipedia.* Procesamiento del Lenguaje Natural, 45:11-19.

Wu, F. And D. Weld. (2010). Open Information Extraction using Wikipedia. In *Proceeding of the 48th Annual Meeting of the Association for Computational Linguistics* (*ACL'10*), pp 118-127, Uppsala: Sweden

Zesch, T., I. Gurevych, and M. Mühlhäuser, (2007). Analyzing and Accessing Wikipedia as a Lexical Semantic Resource, in *Data Structures for Linguistic Resources and Applications*, pp. 197-205.