Towards Large-scale Non-taxonomic Relation Extraction: Estimating the Precision of Rote Extractors*

Enrique Alfonseca[*†] **Maria Ruiz-Casado**[*†] **Manabu Okumura**[*] **Pablo Castells**[†]
[*]Precision and Intelligence Laboratory
Tokyo Institute of Techonology
enrique@lr.pi.titech.ac.jp
oku@pi.titech.ac.jp
[†]Computer Science Department
Universidad Autonoma de Madrid
enrique.alfonseca@uam.es
maria.ruiz@uam.es
pablo.castells@uam.es

Abstract

In this paper, we describe a rote extractor that learns patterns for finding semantic relations in unrestricted text, with new procedures for pattern generalisation and scoring. An improved method for estimating the precision of the extracted patterns is presented. We show that our method approximates the precision values as evaluated by hand much better than the procedure traditionally used in rote extractors.

1 Introduction

With the large growth of the information stored in the web, it is necessary to have available automatic or semi-automatic tools so as to be able to process all this web content. Therefore, a large effort has been invested in developing automatic or semi-automatic techniques for locating and annotating patterns and implicit information from the web, a task known as Web Mining. In the particular case of web content mining, the aim is automatically mining data from textual web documents that can be represented with machine-readable semantic formalisms such as ontologies and semantic-web languages.

Recently, there is an increasing interest in automatically extracting structured information from large corpora and, in particular, from the Web (Craven et al., 1999). Because of the characteristics of the web, it is necessary to develop efficient algorithms able to learn from unannotated data (Riloff and Schmelzenbach, 1998; Soderland, 1999; Mann and Yarowsky, 2005). New types of web content such as blogs and wikis, are also a source of textual information that contain an underlying structure from which specialist systems can benefit.

Consequently, rote extractors (Brin, 1998; Agichtein and Gravano, 2000; Ravichandran and Hovy, 2002) have been identified as an appropriate method to look for textual contexts that happen to convey a certain relation between two concepts. In this paper, we describe a new procedure for estimating the precision of the patterns learnt by a rote extractor, and how it compares to previous approaches. The solution proposed opens new possibilities for improving the precision of the generated patterns, as described below.

This paper is structured as follows: Section 2 describe related work; Section 3 and 4 describe the proposed procedure and its evaluation, and Section 5 presents the conclusions and future work.

2 Related work

Extracting information using Machine Learning algorithms has received much attention since the nineties, mainly motivated by the Message Understanding Conferences. From the mid-nineties, there are systems that learn extraction patterns from partially annotated and unannotated data (Huffman, 1995; Riloff, 1996; Riloff and Schmelzenbach, 1998; Soderland, 1999).

Generalising textual patterns (both manually and automatically) for the identification of relations has been proposed since the early nineties (Hearst, 1992), and it has been applied to extending ontologies with hyperonymy and holonymy relations (Morin and Jacquemin, 1999; Kietz et al., 2000; Cimiano et al., 2004; Berland and Charniak, 1999). Finkelstein-Landau and Morin (1999) learn patterns for company merging relations with exceedingly good accuracies. Recently, kernel

*This work has been sponsored by MEC, project number TIN-2005-06885.

methods are also becoming widely used for relation extraction (Bunescu and Mooney, 2005; Zhao and Grishman, 2005).

Concerning rote extractors from the web, they have the advantage that the training corpora can be collected easily and automatically, so they are useful in discovering many different relations from text. Several similar approaches have been proposed (Brin, 1998; Agichtein and Gravano, 2000; Ravichandran and Hovy, 2002), with various applications: Question-Answering (Ravichandran and Hovy, 2002), multi-document Named Entity Coreference (Mann and Yarowsky, 2003), and generating biographical information (Mann and Yarowsky, 2005). Szpektor et al. (2004) applies a similar, with no seed lists, to extract automatically entailment relationships between verbs, and Etzioni et al. (2005) report very good results extracting Named Entities and relationships from the web.

2.1 Rote extractors

Rote extractors (Mann and Yarowsky, 2005) estimate the probability of a relation $r(p, q)$ given the surrounding context $A_1pA_2qA_3$. This is calculated, with a training corpus T, as the number of times that two related elements $r(x, y)$ from T appear with that same context $A_1xA_2yA_3$, divided by the total number of times that x appears in that context together with any other word:

$$P(r(p,q)|A_1pA_2qA_3) = \frac{\sum_{x,y \epsilon r} c(A_1xA_2yA_3)}{\sum_{x,z} c(A_1xA_2zA_3)} \quad (1)$$

x is called the *hook*, and y the *target*. In order to train a Rote extractor from the web, this procedure is mostly used (Ravichandran and Hovy, 2002):

1. Select a pair of related elements to be used as seed. For instance, *(Dickens,1812)* for the relation *birth year*.
2. Submit the query *Dickens AND 1812* to a search engine, and download a number of documents to build the training corpus.
3. Keep all the sentences containing both elements.
4. Extract the set of contexts between them and identify repeated patterns. This may just be the m characters to the left or to the right (Brin, 1998), the longest common substring of several contexts (Agichtein and Gravano, 2000), or all substrings obtained with a suffix tree constructor (Ravichandran and Hovy, 2002).
5. Download a separate corpus, called *hook corpus*, containing just the hook (in the example, *Dickens*).
6. Apply the previous patterns to the hook corpus, calculate the precision of each pattern in the following way: the number of times it identifies a target related to the hook divided by the total number of times the pattern appears.
7. Repeat the procedure for other examples of the same relation.

To illustrate this process, let us suppose that we want to learn patterns to identify birth years. We may start with the pair (*Dickens*, *1812*). From the downloaded corpus, we extract sentences such as

Dickens was born in 1812
Dickens (1812 - 1870) was an English writer
Dickens (1812 - 1870) wrote Oliver Twist

The system identifies that the contexts of the last two sentences are very similar and chooses their longest common substring to produce the following patterns:

```
<hook> was born in <target>
<hook> ( <target> - 1870 )
```

The rote extractor needs to estimate automatically the precision of the extracted patterns, in order to keep the best ones. So as to measure these precision values, a hook corpus is now downloaded using the hook *Dickens* as the only query word, and the system looks for appearances of the patterns in this corpus. For every occurrence in which the hook of the relation is Dickens, if the target is 1812 it will be deemed correct, and otherwise it will be deemed incorrect (e.g. in *Dickens was born in Portsmouth*).

3 Our proposal

3.1 Motivation

In a rote extractor as described above, we believe that the procedure for calculating the precision of the patterns may be unreliable in some cases. For example, the following patterns are reported by Ravichandran and Hovy (2002) for identifying the relations Inventor, Discoverer and Location:

Relation	Prec.	Pattern
Inventor	1.0	<target> 's <hook> and
Inventor	1.0	that <target> 's <hook>
Discoverer	0.91	of <target> 's <hook>
Location	1.0	<target> 's <hook>

In the particular application in which they are used (relation extraction for Question Answering), they are useful because there is initially a question to be answered that indicates whether we are

looking for an invention, a discovery or a location. However, if we want to apply them to unrestricted relation extraction, we have the problem that the same pattern, the genitive construction, represents all these relations, apart from the most common use indicating possession.

If patterns like these are so ambiguous, then why do they receive so high a precision estimate? One reason is that the patterns are only evaluated for the same hook for which they were extracted. To illustrate this with an example, let us suppose that we obtain a pattern for the relation *located-at* using the pairs *(New York, Chrysler Building)*. The genitive construction can be extracted from the context *New York's Chrysler Building*. Afterwards, when estimating the precision of this pattern, only sentences containing *<target>'s Chrysler Building* are taken into account. Because of this, most of the pairs extracted by this pattern may extract the target *New York*, apart from a few that extract the name of the architect that built it, *van Allen*. Thus we can expect that the genitive pattern will receive a high precision estimate as a *located-at* pattern.

For our purposes, however, we want to collect patterns for several relations such as *writer-book*, *painter-picture*, *director-film*, *actor-film*, and we want to make sure that the obtained patterns are only applicable to the desired relation. Patterns like *<target> 's <hook>* are very likely to be applicable to all of these relations at the same time, so we would like to be able to discard them automatically by assigning them a low precision.

3.2 Suggested improvements

Therefore, we propose the following three improvements to this procedure:

1. Collecting not only a *hook corpus* but also a *target corpus* should help in calculating the precision. In the example of the *Chrysler building*, we have seen that in most cases that we look for the pattern *'s Chrysler building* the previous words are *New York*, and so the pattern is considered accurate. However, if we look for the pattern *New York's*, we shall surely find it followed by many different terms representing different relations, and the precision estimate will decrease.
2. Testing the patterns obtained for one relation using the hook and target corpora collected for other relations. For instance, if the genitive construction has been extracted as a possible pattern for the *writer-book* relation, and we apply it to a corpus about painters, the rote extractor can detect that it also extracts pairs with painters and paintings, so that particular pattern will not be very precise for that relation.
3. Many of the pairs extracted by the patterns in the hook corpora were not evaluated at all when the hook in the extracted pair was not present in the seed lists. To overcome this, we propose to use the web to check whether the extracted pair might be correct, as shown below.

3.3 Algorithm

In our implementation, the rote extractor starts with a table containing some information about the relations for which we want to learn patterns. This procedure needs a little more information than just the seed list, which is provided as a table in the format displayed in Table 1. The data provided for each relation is the following: (a) The **name of the relation**, used for naming the output files containing the patterns; (b) the name of the file containing the **seed list**; (c) the cardinality of the relation. For instance, given that many people can be born on the same year, but for every person there is just one birth year, the cardinality of the relation *birth year* is n:1; (d) the **restrictions** on the hook and the target. These can be of the following three categories: *unrestricted*, if the pattern can extract any sequence of words as hook or target of the relation, *Entity*, if the pattern can extract as hook or target only things of the same entity type as the words in the seed list (as annotated by the NERC module), or *PoS*, if the pattern can extract as hook or target any sequence of words whose sequence of PoS labels was seen in the training corpus; and (e) a sequence of **queries** that could be used to check, using the web, whether an extracted pair is correct or not.

We assume that the system has used the seed list to extract and generalise a set of patterns for each of the relations using training corpora (Ravichandran and Hovy, 2002; Alfonseca et al., 2006a). Our procedure for calculating the patterns' precisions is as follows:

1. For every relation,
   (a) For every *hook*, collect a *hook corpus* from the web.

Relation name	Seed-list	Cardinality	Hook-type	Target-type	Web queries
birth year	birth-date.txt	n:1	entity	entity	$1 was born in $2
death year	death-date.txt	n:1	entity	entity	$1 died in $2
birth place	birth-place.txt	n:1	entity	entity	$1 was born in $2
country-capital	country-capital.txt	1:1	entity	entity	$2 is the capital of $1
author-book	author-book.txt	n:n	entity	unrestricted	$1 is the author of $2
director-film	director-film.txt	1:n	entity	unrestricted	$1 directed $2, $2 directed by $1

Table 1: Example rows in the input table for the system.

(b) For every *target*, collect a *target corpus* from the web.

2. For every relation r,
   (a) For every pattern P, collected during training, apply it to every hook and target corpora to extract a set of pairs.
   For every pair $p = (p_h, p_t)$,
   - If it appears in the seed list of r, consider it correct.
   - If it appears in the seed list of other relation, consider it incorrect.
   - If the hook p_h appears in the seed list of r with a different target, and the cardinality is 1:1 or n:1, consider it incorrect.
   - If the target p_t appears in r's seed list with a different hook, and the cardinality is 1:1 or 1:n, incorrect.
   - Otherwise, the seed list does not provide enough information to evaluate p, so we perform a test on the web. For every query provided for r, the system replaces $1 with p_h and $2 with p_t, and sends the query to Google. The pair is deemed correct if and only if there is at least one answer.

   The precision of P is estimated as the number of extracted pairs that are supposedly correct divided by the total number of pairs extracted.

In this step, every pattern that did not apply at least twice in the hook and target corpora is also discarded.

3.4 Example

After collecting and generalising patterns for the relation *director-film*, we apply each pattern to the hook and target corpora collected for every relation. Let us suppose that we want to estimate the precision of the pattern

<target> 's <hook>

and we apply it to the hook and the target corpora for this relation and for *author-book*. Possible pairs extracted are *(Woody Allen, Bananas), (Woody Allen, Without Fears), (Charles Dickens, A Christmas Carol)*. Only the first one is correct. The rote extractor proceeds as follows:

- The first pair appears in the seed list, so it is considered correct.
- Although *Woody Allen* appears as hook in the seed list and *Without Fears* does not appear as target, the second pair is still not considered incorrect because the *directed-by* relation has n:n cardinality.
- The third pair appears in the seed list for *writer-book*, so it is directly marked as incorrect.
- Finally, because still the system has not made a decision about the second pair, it queries Google with the sequences

  Woody Allen directed Without Fears
  Without Fears directed by Woody Allen

  Because neither of those queries provide any answer, it is considered incorrect.

In this way, it can be expected that the patterns that are equally applicable to several relations, such as *writer-book*, *director-film* or *painter-picture* will attain a low precision because they will extract many incorrect relations from the corpora corresponding to the other relations.

4 Experiment and results

4.1 Rote extractor settings

The initial steps of the rote extractor follows the general approach: downloading a training corpus using the seed list and extracting patterns. The training corpora are processed with a part-of-speech tagger and a module for Named Entity Recognition and Classification (NERC) that annotates people, organisations, locations, dates, relative temporal expressions and numbers (Alfonseca et al., 2006b), so this information can be included in the patterns. Furthermore, for each of the terms in a pair in the training corpora, the system also

Birth year:

```
BOS/BOS <hook> (/( <target> -/- number/entity )/) EOS/EOS
BOS/BOS <hook> (/( <target> -/- number/entity )/) British/JJ writer/NN
BOS/BOS <hook> was/VBD born/VBN on/IN the/DT first/JJ of/IN time_expr/entity ,/, <target> ,/, at/IN location/entity ,/, of/IN
BOS/BOS <hook> (/( <target> -/- )/) a/DT web/NN guide/NN
```

Birth place:

```
BOS/BOS <hook> was/VBD born/VBN in/IN <target> ,/, in/IN central/JJ location/entity ,/,
BOS/BOS <hook> was/VBD born/VBN in/IN <target> date/entity and/CC moved/VBD to/TO location/entity
BOS/BOS Artist/NN :/, <hook> -/- <target> ,/, location/entity (/( number/entity -/-
BOS/BOS <hook> ,/, born/VBN in/IN <target> on/IN date/entity ,/, worked/VBN as/IN
```

Author-book:

```
BOS/BOS <hook> author/NN of/IN <target> EOS/EOS
BOS/BOS Odysseus/NNP :/, Based/VBN on/IN <target> ,/, <hook> 's/POS epic/NN from/IN Greek/JJ mythology/NN
BOS/BOS Background/NN on/IN <target> by/IN <hook> EOS/EOS
did/VBD the/DT circumstances/NNS in/IN which/WDT <hook> wrote/VBD "/'' <target> "/'' in/IN number/entity ,/, and/CC
```

Capital-country:

```
BOS/BOS <hook> is/VBZ the/DT capital/NN of/IN <target> location/entity ,/, location/entity correct/JJ time/NN
BOS/BOS The/DT harbor/NN in/IN <hook> ,/, the/DT capital/NN of/IN <target> ,/, is/VBZ number/entity of/IN location/entity
BOS/BOS <hook> ,/, <target> EOS/EOS
BOS/BOS <hook> ,/, <target> -/- organization/entity EOS/EOS
```

Figure 1: Example patterns extracted from the training corpus for each several kinds of relations.

stores in a separate file the way in which they are annotated in the training corpus: the sequences of part-of-speech tags of every appearance, and the entity type (if marked as such). So, for instance, typical PoS sequences for names of authors are "NNP"[1] (surname) and "NNP NNP" (first name and surname). A typical entity kind for an author is person.

In the case that a pair from the seed list is found in a sentence, a context around the two words in the pair is extracted, including (a) at most five words to the left of the first word; (b) all the words in between the pair words; (c) at most five words to the right of the second word. The context never jumps over sentence boundaries, which are marked with the symbols BOS (*Beginning of sentence*) and EOS (*End of sentence*). The two related concepts are marked as <hook> and <target>. Figure 1 shows several example contexts extracted for the relations *birth year*, *birth place*, *writer-book* and *country-capital city*.

The approach followed for the generalisation is the one described by (Alfonseca et al., 2006a; Ruiz-Casado et al., in press), which has a few modifications with respect to Ravichandran and Hovy (2002)'s, such as the use of the wildcard * to represent any sequence of words, and the addition of part-of-speech and Named Entity labels to the patterns.

The input table has been built with the following nineteen relations: birth year, death year, birth place, death place, author–book, actor–film, director–film, painter–painting, Employee–organisation, chief of state, soccer player–team, soccer team-city, soccer team-manager, country or region–capital city, country or region–area, country or region–population, country–bordering country, country-name of inhabitant (e.g. Spain-Spaniard), and country-continent. The time required to build the table and the seed lists was less than one person-day, as some of the seed lists were directly collected from web pages.

[1] All the PoS examples in this paper are done with Penn Treebank labels.

Relation	Seeds	Extr.	Gener.	Filt.
Birth year	244	2374	4748	30
Death year	216	2178	4356	14
Birth place	169	764	1528	28
Death place	76	295	590	6
Author-book	198	8297	16594	283
Actor-film	49	739	1478	3
Director-film	85	6933	13866	200
Painter-painting	92	597	1194	15
Employee-organisation	62	1667	3334	6
Chief of state	55	1989	3978	8
Soccer player-team	194	4259	8518	39
Soccer team-city	185	180	360	0
Soccer team-manager	43	994	1988	9
Country/region-capital city	222	4533	9066	107
Country/region-area	226	762	1524	2
Country/region-population	288	318	636	3
Country-bordering country	157	6828	13656	240
Country-inhabitant	228	2711	5422	17
Country-continent	197	1606	3212	21

Table 2: Number of seed pairs for each relation, and number of unique patterns in each step.

For each step, the following settings have been set:

- The size of the training corpus has been set to 50 documents for each pair in the original seed lists. Given that the typical sizes of the lists collected are between 50 and 300 pairs, this means that several thousand documents are downloaded for each relation.
- Before the generalisation step, the rote extractor discards those patterns in which the hook and the target are too far away to each other, because they are usually difficult to generalise. The maximum allowed distance

No.	Pattern	Applied	Prec1	Prec2	Real
1	Biography|Hymns|Infography|Life|Love|POETRY|Poetry|Quotations| Search|Sketch|Woolf|charts|genius|kindness|poets/NN */*	6	1.00	1.00	1.00
2	OF|Of|about|by|for|from|like|of/IN <hook> (/(<target> -/- "/'' <hook> (/(<target> -/-	4	1.00	1.00	1.00
3	[BOS]/[BOS] <hook> was/VBD born/VBN about|around|in/IN <target> B.C.|B.C.E|BC/NNP at|in/IN	3	1.00	1.00	1.00
4	[BOS]/[BOS] <hook> was/VBD born/VBN about|around|in/IN <target> B.C.|B.C.E|BC/NNP at|in/IN location/entity	3	1.00	1.00	1.00
5	[BOS]/[BOS] <hook> was/VBD born/VBN around/IN <target> B.C.E/NNP at/IN location/entity ,/, a/DT	3	1.00	1.00	1.00
6	[BOS]/[BOS] <hook> was/VBD born/VBN around|in/IN <target> B.C.|B.C.E/NNP at|in/IN location/entity ,/,	3	1.00	1.00	1.00
7	[BOS]/[BOS] */* ATTRIBUTION|Artist|Author|Authors|Composer|Details| Email|Extractions|Myth|PAL|Person|Quotes|Title|Topic/NNP :/, <hook> (/(<target> -/-	3	1.00	1.00	1.00
8	classical/JJ playwrights/NNS of/IN organisation/entity ,/, <hook> was/VBD born/VBN near/IN location/entity in/IN <target> BCE/NNP ,/, in/IN the/DT village/NN	3	1.00	1.00	1.00
9	[BOS]/[BOS] <hook> (/(<target> -/-)/)	2	1.00	1.00	1.00
10	[BOS]/[BOS] <hook> (/(<target> -|--/-)/)	2	1.00	1.00	1.00
11	[BOS]/[BOS] <hook> (/(<target> person/entity BC/NNP ;/, Greek/NNP :/,	2	1.00	1.00	1.00
12	ACCESS|AND|Alice|Author|Authors|BY|Biography|CARL|Dame|Don|ELIZABETH| (...)|web|writer|writerMuriel|years/NNP <hook> (/(<target> -|- -/-	8	0.75	1.00	
13	-/- <hook> (/(<target> -/-	3	0.67	1.00	0.67
14	-|--/- <hook> (/(<target> -/-	3	0.67	1.00	0.67
15	[BOS]/[BOS] <hook> (/(<target> -/-	60	0.62	1.00	0.81
16	[BOS]/[BOS] <hook> (/(<target> -/- */*)/)	60	0.62	1.00	0.81
17	[BOS]/[BOS] <hook> (/(<target> -|--/-	60	0.62	1.00	0.81
18	,|:/, <hook> (/(<target> -/-	32	0.41	0.67	0.28
19	[BOS]/[BOS] <hook> ,/, */* (/(<target> -|--/-	15	0.40	1.00	0.67
20	,|:|;/, <hook> (/(<target> -|--/-	34	0.38	0.67	0.29
21	AND|Alice|Authors|Biography|Dame|Don|ELIZABETH|Email|Fiction|Frances| GEORGE|Home|I.|Introduction|Jean|L|Neben|PAL|PAULA|Percy|Playwrights| Poets|Sir|Stanisaw|Stanislaw|W.|WILLIAM|feedback|history|writer/NNP <hook> (/(<target> -/-	3	0.33	n/a	0.67
22	AND|Frances|Percy|Sir/NNP <hook> (/(<target> -/-	3	0.33	n/a	0.67
23	Alice|Authors|Biography|Dame|Don|ELIZABETH|Email|Fiction|Frances| GEORGE|Home|I.|Introduction|Jean|L|Neben|PAL|PAULA|Percy|Playwrights| Poets|Sir|Stanisaw|Stanislaw|W.|WILLIAM|feedback|history|writer/NN <hook> (/(<target> -/-	3	0.33	n/a	0.67
24	[BOS]/[BOS] <hook> ,|:/, */* ,|:/, <target> -/-	7	0.28	0.67	0.43
25	[BOS]/[BOS] <hook> ,|:/, <target> -/-	36	0.19	1.00	0.11
26	[BOS]/[BOS] <hook> ,/, */* (/(<target>)/)	20	0.15	0.33	0.10
27	[BOS]/[BOS] <target> <hook> ,/,	18	0.00	n/a	0.00
28	In|On|on/IN <target> ,/, <hook> grew|was/VBD	17	0.00	0.00	0.00
29	In|On|on/IN <target> ,/, <hook> grew|was|went/VBD	17	0.00	0.00	0.00
30	[BOS]/[BOS] <hook> ,/, */* DE|SARAH|VON|dramatist|novelist| playwright|poet/NNP (/(<target> -/-	3	0.00	n/a	1.0
	TOTAL	436	0.46	0.84	0.54

Table 3: Patterns for the relation *birth year*, results extracted by each, precision estimated with this procedure and with the traditional hook corpus approach, and precision evaluated by hand).

between them has been set to 8 words.

- At each step, the two most similar patterns are generalised, and their generalisation is added to the set of patterns. No pattern is discarded at this step. This process stops when all the patterns resulting from the generalisation of existing ones contain wildcards adjacent to either the hook or the target.
- For the precision estimation, for each pair in the seed lists, 50 documents are collected for the hook and other 50 for the target. Because of time constraints, and given that the total size of the hook and the target corpora exceeds 100,000 documents, for each pattern a sample of 250 documents is randomly chosen and the patterns are applied to it. This sample is built randomly but with the following constraints: there should be an equal amount of documents selected from the corpora from each relationship; and there should be an equal amount of documents from hook corpora and from target corpora.

4.2 Output obtained

Table 2 shows the number of patterns obtained for each relation. Note that the generalisation procedure applied produces new (generalised) patterns to the set of original patterns, but no original pattern is removed, so they all are evaluated; this is why the set of patterns increases after the generalisation. The filtering criterion was to keep the patterns that applied at least twice on the test corpus.

It is interesting to see that for most relations the reduction of the pruning is very drastic. This is because of two reasons: Firstly, most patterns are far too specific, as they include up to 5 words at each side of the hook and the target, and all the words in between. Only those patterns that have generalised very much, substituting large portions with wildcards or disjunctions are likely to apply to the sentences in the hook and target corpora.

Secondly, the samples of the hook and target corpora used are too small for some of the relations to apply, so few patterns apply more than twice.

Note that, for some relations, the output of the generalisation step contains less patterns that the output of the initial extraction step: that is due to the fact that the patterns in which the hook and the target are not nearby were removed in between these two steps.

Concerning the precision estimates, a full evaluation is provided for the *birth-year* relation. Table 3 shows in detail the thirty patterns obtained. It can also be seen that some of the patterns with good precision contain the wildcard *. For instance, the first pattern indicates that the presence of any of the words *biography*, *poetry*, etc. anywhere in a sentence before a person name and a date or number between parenthesis is a strong indication that the target is a birth year.

The last columns in the table indicate the number of times that each rule applied in the hook and target corpora, and the precision of the rule in each of the following cases:

- As estimated by the complete program (Prec1).
- As estimated by the traditional hook corpus approach (Prec2). Here, cardinality is not taken into account, patterns are evaluated only on the hook corpora from the same relation, and those pairs whose hook is not in the seed list are ignored.
- The real precision of the rule (real). In order to obtain this metric, two different annotators evaluated the pairs applied independently, and the precision was estimated from the pairs in which they agreed (there was a 96.29% agreement, Kappa=0.926).

As can be seen, in most of the cases our procedure produces lower precision estimates.

If we calculate the total precision of all the rules altogether, shown in the last row of the table, we can see that, without the modifications, the whole set of rules would be considered to have a total precision of 0.84, while that estimate decreases sharply to 0.46 when they are used. This value is nearer the precision of 0.54 evaluated by hand. Although it may seem surprising that the precision estimated by the new procedure is even lower than the real precision of the patterns, as measured by hand, that is due to the fact that the web queries consider unknown pairs as incorrect unless they appear in the web exactly in the format of the query in the input table. Specially for not very well-known people, we cannot expect that all of them will appear in the web following the pattern *"X was born in date"*, so the web estimates tend to be over-conservative.

Relation	Prec1	Prec2	Real
Birth year	**0.46 [0.41,0.51]**	0.84 [0.81,0.87]	0.54 [0.49,0.59]
Death year	**0.29 [0.24,0.34]**	**0.55 [0.41,0.69]**	0.38 [0.31,0.44]
Birth place	0.65 [0.62,0.69]	0.36 [0.29,0.43]	0.84 [0.79,0.89]
Death place	0.82 [0.73,0.91]	1.00 [1.00,1.00]	0.96 [0.93,0.99]
Author-book	0.07 [0.07,0.07]	0.26 [0.19,0.33]	0.03 [0.00,0.05]
Actor-film	**0.07 [0.01,0.13]**	1.00 [1.00,1.00]	0.02 [0.00,0.03]
Director-film	0.03 [0.03,0.03]	0.26 [0.18,0.34]	0.01 [0.00,0.01]
Painter-painting	0.10 [0.07,0.12]	0.35 [0.23,0.47]	0.17 [0.12,0.22]
Employee-organisation	**0.31 [0.22,0.40]**	1.00 [1.00,1.00]	0.33 [0.26,0.40]
Chief of state	**0.00 [0.00,0.00]**	-	0.00 [0.00,0.00]
Soccer player-team	**0.07 [0.06,0.08]**	1.00 [1.00,1.00]	0.08 [0.04,0.12]
Soccer team-city	-	-	-
Soccer team-manager	0.61 [0.53,0.69]	1.00 [1.00,1.00]	0.83 [0.77,0.88]
Country/region-capital city	**0.12 [0.11,0.13]**	0.23 [0.22,0.24]	0.12 [0.07,0.16]
Country/region-area	**0.09 [0.00,0.19]**	1.00 [1.00,1.00]	0.06 [0.02,0.09]
Country/region-population	**1.00 [1.00,1.00]**	**1.00 [1.00,1.00]**	1.00 [1.00,1.00]
Country-bordering country	**0.17 [0.17,0.17]**	1.00 [1.00,1.00]	0.15 [0.10,0.20]
Country-inhabitant	**0.01 [0.00,0.01]**	0.80 [0.67,0.93]	0.01 [0.00,0.01]
Country-continent	0.16 [0.14,0.18]	0.07 [0.04,0.10]	0.00 [0.00,0.01]

Table 4: Precision estimates for the whole set of extracted pairs by *all* rules and all relations.

Table 4 shows the precision estimates for every pair extracted with all the rules using both procedures, with 0.95 confidence intervals. The real precision has been estimating by sampling randomly 200 pairs and evaluating them by hand, as explained above for the *birth year* relation. As can be observed, out of the 19 relations, the precision estimate of the whole set of rules for 11 of them is not statistically dissimilar to the real precision, while that only holds for two relationships using the previous approach.

Please note as well that the precisions indicated in the table refer to all the pairs extracted by all the rules, some of which are very precise, but some of which are very imprecise. If the rules are to be applied in an annotation system, only those with a high precision estimate would be used, and expectedly much better overall results would be obtained.

5 Conclusions and future work

We have described here a new procedure for estimating the precision of the patterns learnt by a rote extractor that learns from the web. Compared to other similar approaches, it has the following improvements:

- For each pair *(hook,target)* in the seed list, a *target corpora* is also collected (apart from the *hook corpora*), and the evaluation is performed using corpora from several relations.

This has been observed to improve the estimate of the rule's precision, given that the evaluation pairs not only refer to the elements in the seed list.

- The cardinality of the relations is taken into consideration in the estimation process using the seed list. This is important, for instance, to be able to estimate the precision in n:n relations like *author-work*, given that we cannot assume that the only books written by someone are those in the seed list.
- For those pairs that cannot be evaluated using the seed list, a simple query to the Google search engine is employed.

The precisions estimated with this procedure are significantly lower than the precisions obtained with the usual hook corpus approach, specially for ambiguous patterns, and much near the precision estimate when evaluated by hand.

Concerning future work, we plan to estimate the precision of the patterns using the whole hook and target corpora, rather than using a random sample. A second objective we have in mind is not to throw away the ambiguous patterns with low precision (e.g. the possessive construction), but to train a model so that we can disambiguate which is the relation they are conveying in each context (Girju et al., 2003).

References

E. Agichtein and L. Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of ICDL*, pages 85–94.

E. Alfonseca, P. Castells, M. Okumura, and M. Ruiz-Casado. 2006a. A rote extractor with edit distance-based generalisation and multi-corpora precision calculation. In *Poster session of ACL-2006*.

E. Alfonseca, A. Moreno-Sandoval, J. M. Guirao, and M. Ruiz-Casado. 2006b. The wraetlic NLP suite. In *Proceedings of LREC-2006*.

M. Berland and E. Charniak. 1999. Finding parts in very large corpora. In *Proceedings of ACL-99*.

S. Brin. 1998. Extracting patterns and relations from the World Wide Web. In *Proceedings of the WebDB Workshop at EDBT'98*.

R. Bunescu and R. J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the HLT Conference and EMNLP*.

P. Cimiano, S. Handschuh, and S. Staab. 2004. Towards the self-annotating web. In *Proceedings of the 13th World Wide Web Conference*, pages 462–471.

M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. 1999. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118(1–2):69–113.

O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2005. Unsupervised named entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

M. Finkelstein-Landau and E. Morin. 1999. Extracting semantic relationships between terms: supervised vs. unsupervised methods. In *Workshop on Ontological Engineering on the Global Info. Infrastructure*.

R. Girju, A. Badulescu, and D. Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *HLT-NAACL-03*.

M. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING-92*.

S. Huffman. 1995. Learning information extraction patterns from examples. In *IJCAI-95 Workshop on New Approaches to Learning for NLP*.

J. Kietz, A. Maedche, and R. Volz. 2000. A method for semi-automatic ontology acquisition from a corporate intranet. In *Workshop "Ontologies and text"*.

G. S. Mann and D. Yarowsky. 2003. Unsupervised personal name disambiguation. In *CoNLL-2003*.

G. S. Mann and D. Yarowsky. 2005. Multi-field information extraction and cross-document fusion. In *Proceedings of ACL 2005*.

E. Morin and C. Jacquemin. 1999. Projecting corpus-based semantic links on a thesaurus. In *ACL-99*.

D. Ravichandran and E. Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL-2002*, pages 41–47.

E. Riloff and M. Schmelzenbach. 1998. An empirical approach to conceptual case frame acquisition. In *Proceedings of WVLC*, pages 49–56.

E. Riloff. 1996. Automatically generating extraction patterns from untagged text. In *AAAI*.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. in press. Automatising the learning of lexical patterns: an application to the enrichment of WordNet by extracting semantic relationships from the Wikipedia. *Data and Knowledge Engineering*, in press.

S. Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1–3):233–272.

I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP 2004*.

S. Zhao and R. Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of ACL-2005*.