Weakly Supervised, Data-Driven Acquisition of Rules for Open Information Extraction

Fabrizio Gotti(✉) and Philippe Langlais(✉)

RALI, Université de Montréal, CP 6128 Succ. Centre-Ville,
Montreal H3C 3J7, Canada
{gottif,langlais}@iro.umontreal.ca

Abstract. We propose a way to acquire rules for Open Information Extraction, based on lemma sequence patterns (including potential typographical symbols) linking two named entities in a sentence. Rule acquisition is data-driven and requires little supervision. Given an arbitrary relation, we identify, in a large corpus, pairs of entities that are linked by the relation and then gather, score and rank other phrases that link the same entity pairs. We experimented with 81 relations and acquired 20 extraction rules for each by mining ClueWeb12. We devised a semi-automatic evaluation protocol to measure recall and precision and found them to be at most 79.9% and 62.4% respectively. Verbal patterns are of better quality than non-verbal ones, although the latter achieve a maximum recall of 76.5%. The strategy proposed does not necessitate expensive resources or time-consuming handcrafted resources, but does require a large amount of text.

Keywords: Open Information Extraction · Weak supervision · Natural Language Processing

1 Introduction

Open Information Extraction (OIE) is a research area in Natural Language Processing that seeks to acquire shallow semantic representation elements from unstructured texts [6]. Typically, this extracted knowledge takes the form of relational tuples like (*Einstein*, *was born in*, *Germany*), composed of arguments flanking a central relation. Contrarily to plain Information Extraction, and importantly, OIE is not bound by a closed list of pre-defined relations or patterns guiding extraction. On the contrary, an ideal OIE system can handle previously unseen ways of expressing information and cast it in tuples, whatever the domain and without supervision.

OIE has been used recently for question-answering [8], and for building domain-targeted knowledge bases [13], among various applications.

Most current OIE systems rely on sets of rules or patterns to perform extraction. These rules can be manually crafted and/or acquired automatically.

M.-J. Meurs and F. Rudzicz (Eds.): Canadian AI 2019, LNAI 11489, pp. 16–28, 2019.
https://doi.org/10.1007/978-3-030-18305-9_2

For the most part, they home in on verbs in the original text to produce an extraction. These heuristics are usually limited to producing extractions built from the tokens originally present in the source text. They rarely "conjure up" new tokens to produce an extraction, instead clinging to the tokens in the source text. Some systems overcome these limitations successfully for a limited number of cases (e.g. appositions, possessives), but overlook many others. Yet a recent handcrafted benchmark in OIE [11] shows that 39% of its triples' relations come from implicit semantics in the text, and that 54% of these triples contain tokens absent from the source text.

For instance, take the excerpt *Barcelona's surrender at the hands of the Nationalist forces of General Francisco Franco*. We understand at the very least that (*Barcelona*, *fall to*, *Franco*), even though there are no verbs in the original text and the words *fall to* never occur within it. The hint that allows us humans to do this mental extraction is the presence of the token sequence starting with *'s surrender at the hands of* surrounded by two easily recognizable named entities. A machine could yet learn the association between that token sequence and the relation at hand, by mining large quantities of text for pairs of entities for which it *knows* that the relation *fall to* holds. The intervening tokens between these two named entities could be paraphrases of that relation.

In this paper, we propose a way of harnessing these lexical hints, in order to generate rules whose purpose is to extract and rephrase elements of meaning "buried" a little more deeply. We suggest a means of extending the expressivity of current extraction heuristics, without much supervision, while at the same time remaining as generic as possible.

Section 2 presents related work, focusing on current rule systems. In Sect. 3, we detail our acquisition methodology. We put it to the test in Sect. 4 and evaluate it in Sect. 5, before discussing the results in Sect. 6.

2 Related Work

As we mentioned in the introduction, extraction rules lie at the very heart of OIE systems. Extractors, starting with the seminal TextRunner [1], have intuitively focused on verbs to identify the relation at hand (but not always exclusively).

TextRunner first identifies interesting noun phrases and then uses a classifier to label the intervening words as part of the relation or not. ReVerb [7] seeks to sanitize the approach by introducing manually crafted regular expressions over the verb-based relational phrase and its arguments. Ollie [12] learns lexicalized and unlexicalized extraction rules using a large number of highly scored ReVerb extractions as seeds. It uses a dependency parser to delve deeper into the syntactic structure of sentences and find long-range dependencies between relations and arguments. Ollie can handle appositions and a few other non-verb-mediated relations, but up to a point. Its rule framework is nonetheless very powerful.

Other systems make use of rules applied to other text elements. ClausIE's hand-crafted rules [6] are applied to a sentence's parse tree to identify meaningful sentence clauses and to cast them into appropriate extraction tuples. Its successor MinIE [9] further processes ClausIE's output with manual rules to produce

minimized, semantically annotated OIE tuples. Its authors use various types of heuristics, including rewrite rules for relations, non-verb-mediated extractors adapted from FINET [5], and word lists indicating a fact's polarity (positive or negative) and certainty. In a recent evaluation [11], MinIE was found to perform best among 7 extractors. Other simplifying extractors include the more recent Graphene [4], which implements a handcrafted simplification and decomposition of sentences in order to yield core tuples as well as accompanying contexts that are semantically linked by rhetorical relations (e.g. temporal).

Non-verb mediated OIE has been the focus of fewer systems. While Ollie or MinIE do handle some cases, more recent systems like RelNoun 1.1 and 2.2 [15] attempt to increase recall by nominal OIE, relying on a set of POS and NP-chunk patterns, as well as additional resources built specifically to handle capitalized relational nouns (e.g. *Paris Mayor Chirac*), demonyms (e.g. *Canadian*) and compound relational nouns (e.g. *health minister*).

Our approach bears some resemblance to PATTY [14], a large resource of textual patterns denoting relations between entities. The patterns are acquired on large corpora and include semantic types for their arguments. PATTY uses seeds from knowledge bases and links their named entities to tokens found in sentence parses in order to infer patterns. These patterns are generalized using sequence of words, part-of-speech tags, wildcards, and ontological types. In our case, we wanted to investigate whether we could do away with parsers, tackle larger corpora, and capture surface patterns including typographical marks like punctuation. Moreover, in this study, we do not rely on an external knowledge base: the process is entirely data-driven.

In a similar vein, the Coupled Pattern Learner (CPL) algorithm [3] of the "never-ending language learner" (NELL) [2] is a sophisticated algorithm designed to learn to extract both relation instances and argument categories (e.g. Movie, Athlete) from unstructured text. While related, it is quite different from our (more lightweight) approach, as it iteratively learns extraction patterns in a semi-supervised way, bootstrapped by a manually crafted input ontology and related seed instances and patterns.

We are also indebted to previous studies using external sources of structured knowledge to label a corpus in order to bootstrap a learning algorithm. See for instance the 2011 system MultiR [10], which uses a probabilistic approach to handle overlapping relations. The authors match Freebase facts in a corpora from the *New York Times* to achieve weak supervision.

3 Semi-supervised Acquisition of Extraction Rules

3.1 Extraction Rules

In this work, an *extraction rule* is a lexical pattern between two named entities (NEs) that should trigger the creation of an OIE triple. Such a rule consists of two elements. The first one is a *pattern*, a sequence of consecutive lemmas flanked on both sides by NEs. The second one is a corresponding *template*, a shell of a triple with the relation already specified, and placeholders for each named entity matched by the pattern.

For instance, the extraction rule NE_1 `'s defeat by` $NE_2 \rightarrow$ (NE_1, *fall to*, NE_2) will match against the excerpt *Constantinople's defeat by the Turks in 1453.* and trigger the production of the triple (*Constantinople*, *fall to*, *the Turks*). It is worth noting that the relation *fall to* expressed in the triple is not actually present in the original sentence, i.e. its tokens are absent, "conjured up" during the extraction process. Also worth mentioning is the presence of the non-word `'s` in the pattern. A pattern could in theory contain only typographical marks, e.g. an opening parenthesis.

3.2 Acquisition Process of Rules for an Arbitrary Relation

Principle. The method we propose to create such extraction rules rely on mining large corpora to find lexical hints that a given relation holds between two named entities. This acquisition proceeds with relatively little supervision. The method's steps are detailed below. Figure 1 illustrates the process for the relation *fall to*.

1. Select an arbitrary relation r for which extraction rules are to be acquired.
2. In a large corpus C, find pairs of named entities (NE_1, NE_2) such that the lemma sequence NE_1 r NE_2 is found verbatim within a sentence.
3. For each pair of named entities (NE_1, NE_2), find sentences in C where NE_1 and NE_2 are present, regardless of their respective positions in the sentences. Collect at most k different sentences matching these criteria, with k being a hyperparameter of the process. We use $k = 200,000$ in this study.
4. For each sentence found in the previous step, gather candidate patterns, i.e. the sequence of tokens positioned between NE_1 and NE_2. Collect and count the candidate patterns.
5. Filter and sort the candidate pattern list (see below for details). Pick the best patterns and associate them with the extraction template (NE_1, r, NE_2).

Filtering and Ranking Extraction Patterns. The noisy process leading up to Step 5 above produces a long list of candidate extraction patterns, many of which are irrelevant. Filtering and ranking these candidates is crucial.

Relying on mere frequency to rank these is fruitless, since the most frequent patterns are also the least specific, occurring for any given relation. For instance, for the relation *fall to*, the most frequent patterns are `fall to` (which is correct, but obviously tautological), `to`, `and`, `by`, and the comma.

We experimented with 3 algorithms in order to rank candidate patterns in decreasing order of specificity. We tested tf–idf, relative frequency and a chi-squared test. The latter was the most effective. This test allows us to compare the statistical distribution of candidate extraction patterns for a given relation r with their distribution in a generic corpus. In our case the latter is formed by the patterns for relations different than r. The contingency table for the chi-squared test is shown in Table 1, for a given relation r and a candidate pattern p for r. We use Yates's correction for continuity for counts < 5. We refer the reader to a statistics handbook for the complete formula for the chi-squared test.

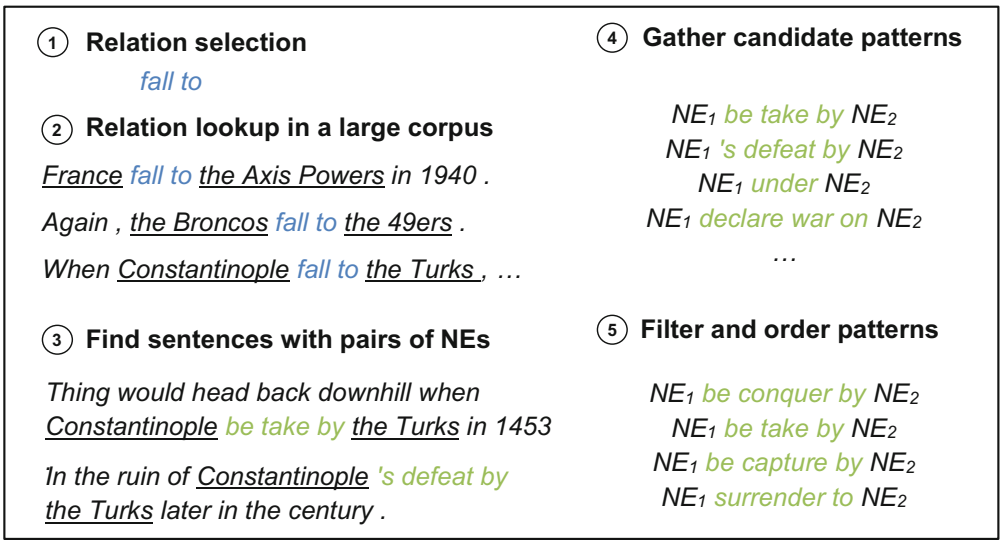

Fig. 1. Schema of the rule acquisition process for the relation *fall to*. Named entities are underlined in the figure. The best rule produced by the example shown would be NE_1 `be conquer by` $NE_2 \rightarrow (NE_1, \textit{fall to}, NE_2)$. All tokens are lemmatized.

We can then sort the candidate patterns in reverse order of their chi-squared score. To further sanitize the resulting list, we use simple additional filters:

- Remove all patterns with more than 7 tokens, e.g. `, r-texas , reveal his pick for a secret service nickname tuesday night on` for the relation *appear on*. In this case, the pattern is correct, but overly specific to appear in an extraction rule.
- Remove all patterns that appear in less than 2% of all relations studied. This indicates a pattern suspiciously specific to a relation, usually the result of a systematic extraction error for that relation.
- Remove the pattern that is identical to the relation at hand, like the pattern `fall to` for the relation *fall to*. Its presence in the candidate list could be viewed as auspicious nonetheless.
- Remove patterns that are stop words or contain a proper noun.

Table 1. Contingency table used for a chi-squared test assessing the specificity of a candidate pattern p for a relation r. Variables a, b, c, and d are frequencies (counts).

	In list of patterns for relation r	In list of patterns for all relations except r
Pattern p	a	b
Patterns different than p	c	d

4 Results for Rule Acquisition

4.1 Selection of Relations

Our acquisition method does not make any assumption about the kinds of relations we want extraction rules for. One starts by specifying them arbitrarily, which is not a form of supervision. In this work, the selection of these relations is dictated by the evaluation methodology we propose later on, in Sect. 5. In other words, we chose relations for which an evaluation is eventually possible. This means finding relations belonging to very high-quality OIE triples, whose veracity is indubitable. We will then be able to compute a recall measure over those triples, using the newly acquired extraction rules.

For the moment, suffice it to say that, to find these triples, we turned to a large collection of 15 million triples[1] extracted by the OIE system ReVerb [7] over the ClueWeb09 dataset. The latter contains 500M web pages collected in 2009. The triple collection was sanitized by its authors, using a threshold on a confidence score produced by ReVerb for each triple, and by additional reasonable heuristics. We then sorted these triples in reverse order of frequency, reasoning that frequent triples would prove more trustworthy.

Finding reliable triples in this list proved unexpectedly difficult. For instance, the most frequent triple is the odd (*Princeton*, *looks for, Lucy*) with a frequency of 3587. Other frequent examples include (*Red found in*, *strawberries*) and (*A*, *means to*, *an end*). This apparent cacophony has multiple explanations. First and foremost, ClueWeb09 is a relatively representative snapshot of the Internet circa 2009, including the inevitable repetitious and malicious content of defaced websites. Second, the extraction process is inherently noisy, even with a tool like ReVerb. Third, even a triple correctly extracted loses its meaning without its context, e.g. (*reprint*, *must include*, *byline*). Our search is further complicated by the fact that we seek triples with named entities for both arguments.

For these reasons, we had to sift through the top 2000 triples to find 128 triples, accounting for 43 different relations. Because 32 of these relations were all of the type *be a X* or *be the X* (e.g. *be a province of*), we explored a further 2000 triples to end up with a more varied final list of 179 triples, accounting for 81 different relations. Examples of such triples are (*ftp*, *stand for*, *file transfer protocol*), (*toronto*, *be the largest city in*, *canada*), and (*einstein*, *be born in*, *ulm*).

4.2 Finding Candidate Patterns in ClueWeb12

Once these 81 relations are selected, we can acquire associated extraction patterns. This entails the lookup of the relations in a large corpus, as previously explained in Sect. 3.2, starting with Step 2. Here, we decided to use ClueWeb12 to find the extraction patterns.[2] The dataset consists of 733M English web pages, collected between February 2012 and May 2012. By using this corpus, we wanted

[1] http://reverb.cs.washington.edu/.
[2] https://lemurproject.org/clueweb12/index.php.

to tackle unstructured text as is commonly found in the wild, and not only in curated texts like Wikipedia. This could allow more variety in the extraction rules ultimately yielded. Moreover, the sheer volume of ClueWeb12 increases our chances of finding relevant information in human-produced English documents.

We started by lemmatizing and indexing all 1.6×10^{10} sentences of the dataset, using Apache Lucene. We also performed part-of-speech tagging on all sentences to detect named entities. We distributed the task on 12 computers.

During **Step 2**, we looked up sentences with named entities flanking each relation, like so: NE_1 *r* NE_2. On average, we gathered 3400 pairs of named entities per relation (min = 61, max = 5000). The average frequency of any given NE pair is 1.1 (min = 1, max = 412 for the pair *Jesus* and *God* in the relation *be the son of*). Most pairs are found only once in co-occurrence with the relation, which is surprising given the large volume of text searched. A cursory examination indicates the pairs to be valid. This step is by far the longest and takes a day.

Step 3 consists in finding sentences in ClueWeb that contain the entity pairs selected during Step 2. We limited the total number of sentences for a given relation at 200k sentences, for performance reasons. We also quickly realized that we could not choose the most frequent entity pairs found in the previous step and find as many sentences as possible that contain them: A variety of entity pairs must be sampled for the overall rule acquisition pipeline to work correctly. Intuitively, this is necessary to gather as many different contexts as possible for these entity pairs, and to avoid oversampling entity pairs which may contain an extraction error. Therefore, we set a limit of 1000 total number of sentences sampled per entity pair. Per relation, we gathered 47k sentences on average (min = 6900 for *be the first wife of*, max = 200k for *be locate in*).

Step 4 sees the first candidate extraction patterns emerge. We obtain an average of 9794 extraction patterns, with a lot of variation ($\sigma = 10,820$). Here, we only consider patterns with a frequency of 5 or more, which gives us 409 extraction patterns on average ($\sigma = 369$).

Finally, **Step 5** filters and ranks extraction patterns. We only consider the top 20 patterns in this study, for a total of $81 \times 20 = 1620$ rules[3]. We show the top 10 patterns for 3 relations in Table 2. For instance, the top extraction rule for *die in* can be read off Table 2 as NE_1 `'s death in` $NE_2 \rightarrow (NE_1$, *die in*, $NE_2)$. We observe that these patterns are generally relevant and specific, but not always, e.g. the pattern `the great in` for the relation *die in*. Out of 1620 rules, we identified 809 rules (50%) whose pattern is non-verbal, like the previous example. We found 531 rules (32.7%) containing typographical marks, e.g. `- owner of`.

5 Evaluation

Evaluation is a delicate topic in OIE, as there are no gold standards or metrics that are agreed upon in the community. Some researchers will annotate the

[3] Download them here: http://rali.iro.umontreal.ca/rali/oie-pararules.

Table 2. Top 10 extraction patterns for 3 relations, in decreasing order of relevance. The top extraction rule for the relation *die in* can be read off column 2 as NE_1 `'s death in` $NE_2 \rightarrow (NE_1, die\ in, NE_2)$. Non-verbal patterns are underlined.

fall to	*die in*	*be the owner of*
be conquer by	's death in	, owner of
be destroy by	the great in	, the owner of
be capture by	die on	own
at the hand of	society of	, who own
be take by	be assassinate in	(owner of
sink	's grave in	, which own
fall into the hand of	died in	- owner of
be occupy by	die last	be the president of
be defeat by	pass away in	a very
surrender to	be bury in	, who also own

output of their system for correctness, e.g. [12] or [16], which yields a precision measure without being able to offer a sense of the recall. Others have created benchmarks automatically, like the increasingly cited paper of Stanovsky and Dagan [17]. However such benchmarks are usually created with verb phrases as the focus of tuple extraction. They have other problems, described in [11].

In our case, we deemed it crucial to assess both precision and recall, while at the same time steer clear of any methodology that would only look at verbs in the source sentences. Indeed, the acquisition process we propose here can easily err on the side of recall at the expense of precision. If, for instance, we were to use a rule such as NE_1 `at` $NE_2 \rightarrow (NE_1, locate\ in, NE_2)$, then all occurrences of *at* would erroneously trigger the rule. The recall would be high for this relation, while at the same time extremely wanting in precision.

Moreover, verb-centric benchmarks like the one described in [17] would not be sensitive enough to measure the impact of extractions triggered by non-verbal clues, like the aforementioned NE_1 `'s death in` $NE_2 \rightarrow (NE_1, die\ in, NE_2)$.

We propose an evaluation protocol taking these issues into account. Importantly, we did not create a full-fledged OIE system just for the purpose of evaluation. Rather, we used reasonable proxies to measure the performance of such a system were it to be programmed on the basis of the extraction rules found in the previous section.

5.1 Recall

Recall must be measured over a set of triples that are assuredly true and that each uses one of the 81 relations described earlier. We need only turn to Sect. 4.1 to find this test set: the 179 triples described are hand-validated and evidently use the relations at hand. Measuring recall then consists in matching a corpus's sentences against the extraction rules acquired and measuring what proportion of the 179 triples are found.

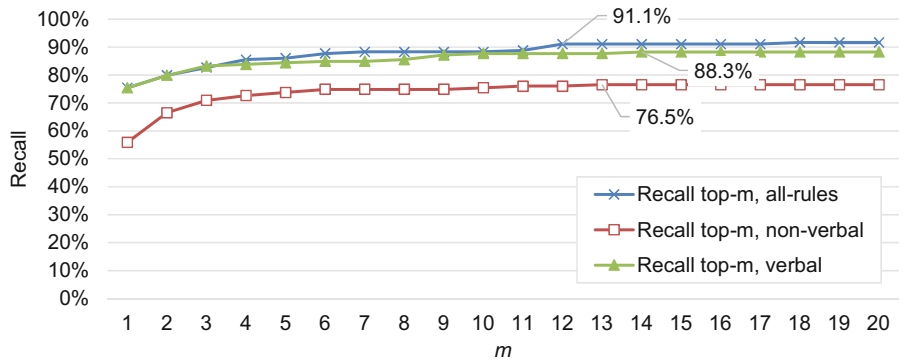

Fig. 2. Recall measure over 179 reference triples, for different values of m, the number of top rules retained for each relation. Three rule set are evaluated. **all-rules**: all the extraction rules acquired (1620 rules), **non-verbal** rules only (809 rules) and **verbal** rules only (811 rules). Each curve is labeled with the maximum recall value reached.

We applied our rules on ClueWeb12 and inspected the triples for recall over the 179 gold triples. Since we have 20 rules per relation, we also computed recall by considering only the top m rules per relation, yielding 20 additional recall metrics, Recall top-m, where $1 \leq m \leq 20$. We can further refine our metrics by considering only the subset of rules that are either verbal or non-verbal, as identified in Sect. 4.2. The complete results are shown in Fig. 2.

Unsurprisingly, the complete rule set (all-rules) achieves the highest recall, with 91.1% of all triples, a mark hit for $m \geq 12$. Not far behind is the rule set limited to verbal patterns, at 88.3% (starting at $m = 14$). The recall for the non-verbal rule set culminates at 76.5% (starting at $m = 13$). The best recall is quite high, which is encouraging. Even when only the top rule is kept for each relation ($m = 1$), the best recall is 75.4%. Because the recall starts at a relatively high value, further extraction rules have a limited effect, but they do help. Out of 179 reference triples, 16 triples cannot be found with our best system. Manual examination reveals that there are two causes for this.

The first difficulty is with triples whose relation yields relatively few candidate patterns (Step 4 in Fig. 1). For instance, the reference triple (*Aristotle*, *return to*, *Athens*) uses a relation that only generates 130 extraction patterns after filtering (the average is 409 patterns). The top patterns for this relation are dubious: `grip`, `may be from`, `profess his love for`, etc.

The second problem is actually an artefact of our evaluation procedure. For instance, we cannot find the reference triple (*britt gillette*, *be author of*, *the dvd report*) using our extraction rules because the only co-occurrence of *britt gillette* and *the dvd report* in ClueWeb sentences is in the token sequence *britt gillette be author of the dvd report*. Since we exclude the relation itself (*be author of* here) from the list of extraction rules evaluated, we miss this triple. If we compensate for this, the aforementioned 91.1% recall is bumped up to 92.7%.

The verbal patterns outperform the non-verbal ones by a significant margin of 11.8% absolute, even if their respective counts are almost the same.

This confirms the intuition that verb phrases are stronger hints of the presence of a relation, while at the same time shows that non-verbal patterns are fertile on their own, recalling at most 76.5% of triples.

5.2 Precision

Assessing precision in OIE is tricky, for the reasons put forward at the beginning of this section. We could not rely on an automatic metric like we did for recall and instead had to manually assess the quality of the output. Since our extraction rules are not yet part of a full-fledged OIE system, we had to simulate the processing performed by such a system manually.

For each relation r, we enumerated its 20 extraction patterns p_i and randomly selected from ClueWeb12 up to 10 sentences that matched the pattern $NE_1\ p_i\ NE_2$. We then labeled the corresponding extractions (NE_1, r, NE_2) as either correct or incorrect. A correct extraction is one whose meaning is clearly stated in the original sentence. The task was very time-consuming, so we resorted to a random sample of 20 out of 81 relations to carry out the assessment.

For the 20 relations evaluated (and their associated 400 extraction rules), we labeled 3559 triples, an average of 8.9 triples per rule. Like we did for recall, we also computed precision by considering the top m rules per relation, yielding a precision measure for each m, where $1 \leq m \leq 20$. Figure 3 shows the results.

Precision decreases from 62.4% for $m = 2$ down to 36.4% for $m = 20$. This is expected, because the lower an extraction rule is in the list associated with a relation, the lower its quality will be. An F-measure cannot be computed here, since precision and recall are obtained from heterogeneous processes. Ultimately, for $m = 2$ (top 2 rules only), we have a recall of 79.9% and a precision of 62.4%. Non-verbal patterns' precision is disappointingly low.

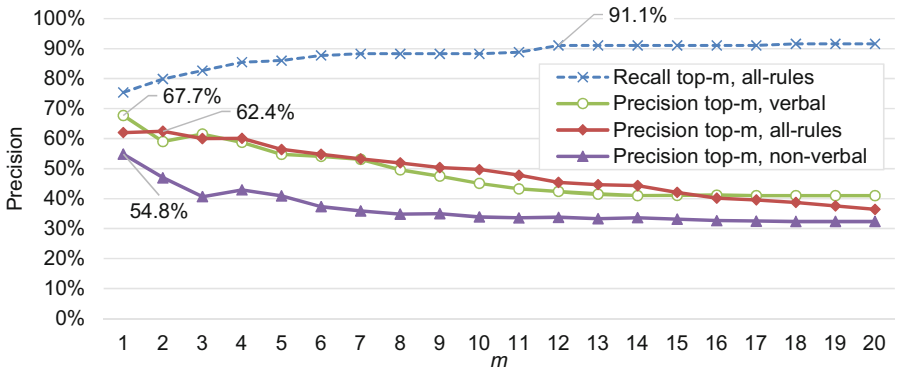

Fig. 3. Precision assessment over 3559 triples, for different values of m, the number of top rules retained for each relation. The recall measure is copied from Fig. 2 to provide a more complete picture of the evaluation.

Two types of error decrease precision. Firstly, erroneously acquired extraction patterns corrupt extraction. A rule like NE_1 `society of` $NE_2 \rightarrow (NE_1, \textit{die in}, NE_2)$ will always fail. This is the case for 50.7% of rules, a high figure, but it is crucial to point out that erroneous rules are typically ranked much lower, with a rank of 11.0 on average. Indeed, 97.8% of rules at rank 1 are valid. Secondly, some rules are correctly triggered in certain contexts, but not in others. For instance, the rule NE_1 `then part of` $NE_2 \rightarrow (NE_1, \textit{be ruled by}, NE_2)$ should be triggered in *Sicily, then part of the Roman Empire* but not for *Haifa, then part of British Palestine*. A manual examination of the non-verbal patterns show that they tend to be shorter on average (2.7 words vs. 3.0 words for verbal patterns) and tend to employ a more generic vocabulary (e.g. patterns like `mayor`, `within`), which may explain why they are triggered unnecessarily.

6 Discussion

In this work, we set out to extend the heuristic possibilities currently used in OIE by exploring the potential of arbitrary lemma patterns to trigger cogent extractions. We further wanted to acquire these additional rules without too much supervision, and in a generic framework consistent with the goals of OIE.

Our experiments show, at the very least, that it is possible to acquire these rules using data mining over a large corpus. Our manual evaluation shows that for 81 relations the top rule is sensible in 97.8% of the cases, comprising interesting non-verbal paraphrases like *Smith's death in Mali* $\rightarrow$ (*Smith*, *die in*, *Mali*). Our strategy can extract overlapping facts, a common occurrence according to [10].

While evaluation remains difficult in OIE, we managed to devise a proxy to a full-fledged extractor and apply it to measure recall and precision. The most precise configuration retains the top 2 rules for each relation, for a precision of 62.4% and a recall of 79.9%. The top precision and recall across all configurations is respectively 67.7% and 91.1%. Verbal patterns tend to outperform non-verbal ones in both respect, which is somewhat disappointing, but not unexpected.

While the results leave room for improvement, especially regarding precision, one must also consider how little supervision went into the acquisition process. By merely amplifying the signal given by a seed consisting of entity pairs linked by a relation, we can gather additional extraction rules by mining data.

We could nonetheless ameliorate our work by adding entity types (e.g. person, place) to our extraction rules, and by relying less on named entities, in order to extract desirable triples like (*meteors*, *occur in*, *the mesosphere*). It also remains to be seen how the strategy we propose can be extended to a very large number of relations. Data-mining approaches are sensitive to the amount of text at their disposal to accomplish their task. Here, we sifted through high-frequency ReVerb triples in order to find enough acceptable triples and their associated relations. It could prove more difficult to acquire rules in the long tail of rarer triples. This is important, because OIE strives to be generic.

References

1. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007, India, pp. 2670–2676 (2007)
2. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr., E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010), p. 3, July 2010
3. Carlson, A., Betteridge, J., Wang, R.C., Hruschka, Jr., E.R., Mitchell, T.M.: Coupled semi-supervised learning for information extraction. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM 2010, pp. 101–110. ACM, New York (2010)
4. Cetto, M., Niklaus, C., Freitas, A., Handschuh, S.: Graphene: semantically-linked propositions in open information extraction. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2300–2311. ACL (2018)
5. Del Corro, L., Abujabal, A., Gemulla, R., Weikum, G.: FINET: context-aware fine-grained named entity typing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 868–878. ACL (2015)
6. Del Corro, L., Gemulla, R.: ClausIE: clause-based open information extraction. In: Proceedings of the 22nd International Conference on World Wide Web, WWW 2013, pp. 355–366. ACM, New York (2013)
7. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Empirical Methods in Natural Language Processing, EMNLP 2011, pp. 1535–1545 (2011)
8. Fader, A., Zettlemoyer, L., Etzioni, O.: Open question answering over curated and extracted knowledge bases. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, pp. 1156–1165. ACM, New York (2014)
9. Gashteovski, K., Gemulla, R., Del Corro, L.: MinIE: minimizing facts in open information extraction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2630–2640. ACL (2017)
10. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies - Volume 1, HLT 2011, pp. 541–550. ACL (2011)
11. Léchelle, W., Gotti, F., Langlais, P.: WiRe57 : A Fine-Grained Benchmark for Open Information Extraction. arXiv:1809.08962 [cs], September 2018
12. Mausam Schmitz, M., Bart, R., Soderland, S., Etzioni, O.: Open language learning for information extraction. In: Joint Conference on Empirical Methods in NLP and Computational Natural Language Learning, pp. 523–534 (2012)
13. Mishra, B.D., Tandon, N., Clark, P.: Domain-targeted, high precision knowledge extraction. Trans. ACL **5**, 233–246 (2017)
14. Nakashole, N., Weikum, G., Suchanek, F.: PATTY: a taxonomy of relational patterns with semantic types. In: Joint Conference on Empirical Methods in NLP and Computational Natural Language Learning, pp. 1135–1145 (2012)
15. Pal, H., Mausam: demonyms and compound relational nouns in nominal open IE. In: AKBC@NAACL-HLT (2016)

16. Saha, S., Pal, H., Mausam: bootstrapping for numerical open IE. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 317–323. ACL, Vancouver (2017)
17. Stanovsky, G., Dagan, I.: Creating a large benchmark for open information extraction. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2300–2305. ACL, Austin (2016)