Automatic Evaluation of Relation Extraction Systems on Large-scale

Mirko Bronzi [†], **Zhaochen Guo** [‡], **Filipe Mesquita** [‡], **Denilson Barbosa** [‡], **Paolo Merialdo** [†]
[†]Università degli Studi Roma Tre
Via della Vasca Navale, 79
Rome, Italy
{bronzi,merialdo}@dia.uniroma3.it

[‡]University of Alberta
2-32 Athabasca Hall
Edmonton, Canada
{zhaochen,mesquita,denilson}@ualberta.ca

Abstract

The extraction of relations between named entities from natural language text is a long-standing challenge in information extraction, especially in large-scale. A major challenge for the advancement of this research field has been the lack of meaningful evaluation frameworks based on realistic-sized corpora. In this paper we propose a framework for large-scale evaluation of relation extraction systems based on an automatic annotator that uses a public online database and a large web corpus.

1 Introduction

It is envisioned that in the future, the main source of structured data to build knowledge bases will be automatically extracted from natural language sources (Doan et al., 2009). One promising technique towards this goal is Relation Extraction (RE): the task of identifying relations among named entities (e.g., people, organizations and geo-political entities) from natural language text. Traditionally, RE systems required each target relation to be given as input along with a set of examples (Brin, 1998; Agichtein and Gravano, 2000; Zelenko et al., 2003). A new paradigm termed *Open RE* (Banko and Etzioni, 2008) has recently emerged to cope with the scenario where the number of target relations is too large or even unknown. Open RE systems try to extract every relation described in the text, as opposed to focusing on a few relations (Zhu et al., 2009; Banko and Etzioni, 2008; Hasegawa et al., 2004; Rosenfeld and Feldman, 2007; Chen et al., 2005; Mesquita et al., 2010; Fader et al., 2011).

One challenge in advancing the state-of-the-art in open RE (or any other field for that matter) is having meaningful and fair ways of evaluating and comparing different systems. This is particularly difficult when it comes to evaluating the *recall* of such systems, as that requires one to enumerate all relations described in a corpus.

In order to scale, a method for evaluation of open RE must have no human involvement. One way to automatically produce a benchmark is to use an existing database as ground truth (Agichtein and Gravano, 2000; Mintz et al., 2009; Mesquita et al., 2010) . Although a step in the right direction, this approach limits the evaluation to those relations that are present in the database. Another shortcoming is that the database does not provide "true" recall, since it often contains many more facts (for the relations it holds) than described in the corpus.

Measuring true precision and recall In this paper we discuss an automatic method to estimate true precision and recall of open RE systems. We propose the use of an automatic annotator: a system capable of verifying whether or not a fact was correctly extracted. This is done by leveraging external sources of data and text, which are not available to the systems being evaluated. The external database used in this work is Freebase, a curated online database maintained by an active community. In addition to the external database, our automatic annotator leverages Pointwise Mutual Information (PMI) (Turney, 2001) from the web. PMI has been widely accepted to measure the confidence score of an extraction (Etzioni et al., 2005). We show that

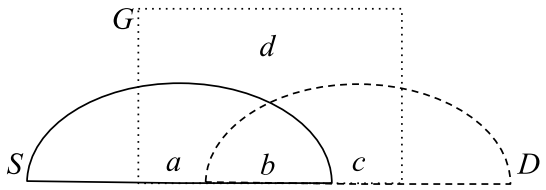

Figure 1: Venn diagram showing the interaction between an external database D (Freebase), the ground truth G and a system output S.

PMI is also useful to evaluate systems automatically.

Using our method, we compare two state-of-the-art open RE systems, ReVerb (Fader et al., 2011) and SONEX (Mesquita et al., 2010), applied to the same corpus, namely the New York Times Corpus (Sandhaus, 2008).

2 Evaluation Methodology

We now describe how our method measures both true precision and true recall, using a database and the web (as a large external text corpus). A *fact* is a triple $f_i = \langle e_1, r, e_2 \rangle$ associating entities e_1 and e_2 via relation r. We measure precision by assessing how many of the facts produced by the system have been correctly extracted. A fact is said to be *correct* if (1) we can find the fact in the database or (2) we can detect a statistically significant association between e_1, e_2 and r on the web. To measure recall, we estimate the size of the ground truth (i.e., the collection of *all* facts described in the corpus).

2.1 Interactions between the system, database and ground truth

Now, we discuss our method to evaluate open RE systems. Given a corpus annotated with named entities, an open RE system must produce a set of facts $S = \{f_1, f_2, \ldots, f_{|S|}\}$. An example of fact is ⟨"Barack Obama","married to","Michelle Obama"⟩. In order to evaluate the precision of S, we partially rely on an external database $D = \{f_1, f_2, \ldots, f_{|D|}\}$. In order to measure recall, we try to estimate the set of facts described in the input corpus. This set corresponds to the ground truth and it is denoted by $G = \{f_1, f_2, \ldots, f_{|G|}\}$.

In Figure 1, we present a Venn diagram that illustrates the interactions between the system output (S), the ground truth (G) and the external database (D). There are four marked regions (a, b, c, d) in this diagram. We need to estimate the size of these regions to measure the true precision and recall of a system. We discuss each marked region as follows.

- a contains correct facts from the system output that are not in the database.
- b is the intersection between the system output and the database ($S \cap D$). We assume that this region is composed by correct facts only, i.e., facts that are in the ground truth. This is because it is unlikely for a fact mistakenly extracted by a system to be found in the database.
- c contains the database facts described in the corpus but not extracted by the system.
- d contains the facts described in the corpus that are not in the system output nor in the database.

Precision and recall Observe that all true positives are in regions a and b, while all false negatives are in regions c and d. Considering that $|G|$ = $|a| + |b| + |c| + |d|$, we can define precision (P), recall (R) and F-measure (F) as follows.

$$P = \frac{|a| + |b|}{|S|} \qquad R = \frac{|a| + |b|}{|a| + |b| + |c| + |d|}$$

$$F = \frac{2 \cdot P \cdot R}{P + R}$$

The need for the web An evaluation method that relies exclusively on a database can only determine the size of regions b and c. Therefore, in order to compute true precision and recall we need to evaluate those facts that are not in the database. The whole web would be the ideal candidate for this task since it is by far the most comprehensive source of information. In our preliminary experiments, more than 97% of the extractions cannot be evaluated using a database only.

2.2 Estimating precision

To measure precision, we need to estimate the size of the regions a and b.

Using the external database We calculate the size of region b by determining, for each fact $f = \langle e_1, r, e_2 \rangle$ in S, whether f is in D. In our experiments, D corresponds to Freebase, which contains data from many sources, including Wikipedia. Freebase provides Wikipedia ids for many of its entities. Since we perform entity disambiguation with Wikipedia as a preprocessing step, finding e_1 and e_2 in Freebase is trivial.

On the other hand, we are required to match r to a relation in Freebase. We perform this matching by using a widely-used semantic similarity measure proposed by Jiang and Conrath (Jiang and Conrath, 1997). This measure uses a lexical terminology structure (WordNet) with corpus statistics. Given a relation r' in Freebase, we determine the similarity between r and r' by the maximum similarity between the words that compose r and r'. We select the relation r' with maximum similarity with r and consider that $r = r'$ if their similarity score is above a predetermined threshold.

Using the web We estimate $|a|$ by leveraging Pointwise Mutual Information (PMI) on web documents. In particular, we use an adaptation of the PMI-IR (Turney, 2001), which computes PMI using a web search engine. The PMI of a fact $f = \langle e_1, r, e_2 \rangle$ measures the likelihood of observing f, given that we observed e_1 and e_2, i.e,

$$\text{PMI}(e_1, r, e_2) = \frac{Count(e_1 \text{ AND } r \text{ AND } e_2)}{Count(e_1 \text{ AND } e_2)} \quad (1)$$

where $Count(q)$ is the number of documents returned by the query q. PMI values range from 0 (when f is not observed) to 1 (when f is observed for every occurrence of the pair e_1 and e_2). We use the PMI function to determine whether a fact was correctly extracted. The underlying intuition is that facts with high (relative) frequency are more likely to be correct.

There are different ways one can estimate the result of the $Count(\cdot)$ function. One may use the hit counts of a web search engine, such as Google or Bing. Another option is to use a local search engine, such as *Lucene*[1], on a large sample of the web, such as the ClueWeb09 corpus.

[1] http://lucene.apache.org/

We consider two versions of the PMI function, which differ by how their queries are defined. Equation 1 presents the CLASSIC version, which uses the AND operator. This simple approach is efficient but ignores the locality of query elements. It is known that query elements close to each other are more likely to be related than those sparsely distributed throughout the document. The second version of PMI, called PROXIMITY, relies on proximity queries, which consider the locality aspect. In this version, queries are of the form "e_1 NEAR:X r NEAR:X e_2", where X is the maximum number of words between the query elements. In Figure 2 we see an example of proximity query.

We deem a fact as correct if its PMI value is above a threshold t, determined experimentally [2] . By calculating the PMI of extracted facts that are not in the region b, we are able to estimate $|a|$. With both $|a|$ and $|b|$, we estimate the precision of the system.

2.3 Estimating recall

To provide a trustworthy estimation of recall, we need to estimate the size of regions c and d. We produce a superset G' of the ground truth G ($G' \supseteq G$). Note that G' contains real facts (G) as well as wrongly generated facts ($G' \setminus G$). We approximate G by removing these wrong facts, either exploiting the external database and the PMI function.

One way to produce G' is to perform a Cartesian product of all possible entities and relations. Let $E = \{e_1, e_2, \ldots, e_m\}$ be the set of entities and $R = \{r_1, r_2, \ldots, r_n\}$ be set of relations found in the input corpus. The superset of G produced by Cartesian product is $G' = E \times R \times E$. For example, the facts extracted from the sentence "Barack Obama is visiting Rome to attend the G8 Summit" are presented in Figure 3, where the correct facts are highlighted. The shortcoming of this approach is the huge size of the resulting G'. Even so, we remove many incorrect facts thanks to heuristics; e.g., we do not consider entities from different sentences.

Once G' is produced, we estimate $|G \cap D| = |b| + |c|$ by looking for facts in G' that match a fact in the database D, as before. Once we have $|b|$ and $|G \cap D|$, we can estimate $|c| = |G \cap D| - |b|$. By applying

[2] Threshold t is domain-independent, as shown by other important works such as (Hearst, 1992; Banko et al., 2007; Banko and Etzioni, 2008).

1	2	3	4	5	6	7	8
Valerie Jarrett	was	appointed	as	senior	advisor	by	Barack Obama

Figure 2: A sentence matching the query "(Valerie Jarrett) NEAR:4 (advisor) NEAR:4 (Barack Obama)". Grey words represent matching terms, while white words are noise.

e_1	r	e_2
Barack Obama	**visit**	**Rome**
Barack Obama	visit	G8 Summit
Barack Obama	attend	Rome
Barack Obama	**attend**	**G8 Summit**
Rome	visit	G8 Summit
Rome	attend	G8 Summit

Figure 3: Facts produced for the superset G' for "Barack Obama is visiting Rome to attend the G8 Summit". Facts in the ground truth G are highlighted in bold.

Annotators	Agreement
$H_0 - H_1$	80.8%
$H_1 - H_2$	80.3%
$H_0 - H_2$	78.0%
$A - H_0$	71.9%
$A - H_1$	68.8%
$A - H_2$	72.8%
$A - H_{12}$	75.9%

Table 1: Agreement between human and automatic annotators.

the PMI of the facts not in the database ($G' \setminus D$) we can determine $|G \setminus D|$. Finally, we can estimate $|d| = |G \setminus D| - |a|$. Now that we have estimated the sizes of regions a, b, c and d, we can determine the true recall of the system.

2.4 PMI Effectiveness

To measure PMI effectiveness, we compare the results of our evaluation system (A) and a human (H_0) over a set of 558 facts. To this end, we defined the agreement between A and H_0 as follows.

$$\text{Agreement} = \frac{\text{Number of facts where } A = H_0}{\text{Number of facts}}$$

Our system achieved an agreement of 73% with respect to the human evaluation; the agreement increases up to 80% if we consider only popular facts. This is a well-known property of PMI: when dealing with small hit count numbers, the PMI function is very sensible to changes, amplifying the effect of errors.

We also compare how distant the agreement achieved with the automatic annotator (A) is from the agreement between humans. For this experiment, we asked two additional volunteers (H_1 and H_2) to evaluate the set of 558 facts as before. For a more reliable measurement we created an additional annotator (H_{12}) by selecting the facts where H_1 and H_2 agreed. We also include the human annotations (H_0) from the previous experiment.

Table 1 shows the agreement between humans and the automatic annotator. While the agreement between humans varies between 78% and 81%, the agreement between human and automatic annotators varies between 69% and 73%. These results show that our automatic annotator is promising and could potentially achieve human levels of agreement with little improvement. In addition, the agreement with the more reliable annotator H_{12} is quite high at 76%.

2.5 The Difference Between Extracting and Evaluating Relations

The tasks of extracting relations from a corpus (e.g., New York Times) and evaluating relations using a corpus (e.g., the web) are virtually the same. However, we stress how an evaluation process is performed in an easier scenario, thus more effective.

In order to measure precision, we judge a fact as correct or wrong by looking for mentions in the external sources. This process is easier than extracting a fact: first, we already know the fact we are looking for; second, this fact is probably going to be replicated many times in several different ways, and so easy to spot. This is not true for a generic extraction process, where the fact may be published only once and in a particular difficult form.

For measuring recall, our evaluation system has both to generate and validate facts; as a consequence, it has to perform as a real extraction system. Even so, our system still performs in a easier sce-

nario: in fact, to materialize the extracted data, we randomly generate facts, and then we filter out the ones that are not replicated anywhere else. Note that our system can hardly be used as an extraction system: we only validate facts already published somewhere else, i.e., we do not generate any new information, that is the main goal of an extraction system; moreover, we require several additional information sources.

3 Comparing ReVerb and SONEX

We now use our evaluation method to compare two open RE systems: ReVerb and SONEX. The input corpus for this comparison is the New York Times corpus, composed by 1.8 million documents.

ReVerb (Fader et al., 2011) extracts relational phrases using rules over part-of-speech tags and noun-phrase chunks. It also employs a logistic regression classifier to produce a confidence score for each extracted fact; an extracted fact is only included in the output if above a user-defined threshold. SONEX (Mesquita et al., 2010) tries to find sets of entity pairs that share the same relation by clustering them. SONEX uses a hierarchical agglomerative clustering algorithm (Manning et al., 2008).

3.1 Results

We run ReVerb with five different confidence thresholds (0.2, 0.4, 0.6, 0.8, 0.95) and report the output with highest F-measure (0.2 in our case). SONEX uses a user-defined threshold to stop the agglomerative clustering. We try five different thresholds (0.1, 0.2, 0.3, 0.4, 0.5) and report the output with highest F-measure (0.4 in our case). For each run, we randomly select 10 thousand facts from the output of each system. These are used to estimate the sizes of regions a and b. We also randomly select 40 thousand facts from G' to estimate the sizes of c and d.

Reverb produced about 2.6 million facts, while SONEX produced over 3.2 million facts. We found about 63 million facts in G', the superset of the ground truth G. Table 2 presents the size of all regions for ReVerb and SONEX. Note that Freebase (regions b and c) plays a minor role in this estimation when compared to PMI (regions a and d): more than 97% of the ground truth is defined by using PMI. This behaviour can be explained by the small number of facts with two entities with a corresponding entry in Wikipedia: 1.6% for ReVerb, 0.9% for SONEX, 1.7% for G'. The importance of the external database may be higher for other corpora (e.g., Wikipedia) better covered by the database (e.g., Freebase).

Systems	a	b	c	d	S	D	G'
ReVerb	77	3	41	1,944	2,643	3,926	62,930
SONEX	259	4	40	1,763	3,288	3,926	62,930

Table 2: The size of all regions for ReVerb and SONEX, in thousands of facts.

Systems	Precision	Recall	F-measure
ReVerb	3.1%	3.9%	3.4%
SONEX	8.0%	12.8%	9.8%

Table 3: Performance results for ReVerb and SONEX.

Table 3 shows the precision, recall and F-measure for ReVerb and SONEX. Observe that SONEX achieves more than double the precision and recall presented by ReVerb; however both systems presented low results. These results not only illustrate but also quantify the challenges of dealing with large corpora. Moreover, they underscore the pressing need for more robust and effective open RE tools. Finally, they yield a vast amount of incorrect extractions, which are in turn an invaluable source of open problems in this field.

4 Conclusion and Future Work

This paper introduces the first automatic method for large-scale evaluations of open RE systems that estimates true precision *and* recall. Our method scales to realistic-sized corpora with million of documents, instead of the few hundreds of previous evaluations.

Our contributions indicate that a fully automatic annotator can indeed be used to provide a fair and direct evaluation of competing open RE systems. Moreover, we stress how an automatic evaluation tool represents an invaluable resource in aiding and speeding-up the development process of open RE systems, by removing the tedious and error-prone task of manual evaluation.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: extracting relations from large plain-text collections. In *Proceedings of the ACM Conference on Digital libraries*, pages 85–94. ACM.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of the Annual Meeting of the ACL*, pages 28–36, Columbus, Ohio, June. Association for Computational Linguistics.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the Web. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2670–2676.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases, International Workshop*, pages 172–183.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2005. Unsupervised feature selection for relation extraction. In *Proceedings of the International Joint Conference on Natural Language Processing*. Springer.

A. Doan, J. F. Naughton, A. Baid, X. Chai, F. Chen, T. Chen, E. Chu, P. Derose, B. Gao, C. Gokhale, J. Huang, W. Shen, and B. Vuong. 2009. The Case for a Structured Approach to Managing Unstructured Data. In *Proc. CIDR*.

Oren Etzioni, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying Relations for Open Information Extraction. In *EMNLP*.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the Annual Meeting of the ACL*, page 415. Association for Computational Linguistics.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *In Proceedings of the 14th International Conference on Computational Linguistics*, pages 539–545.

Jay J Jiang and David W Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *Computational Linguistics*, cmp-lg/970(Rocling X):15.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July.

Filipe Mesquita, Yuval Merhav, and Denilson Barbosa. 2010. Extracting information networks from the blogosphere: State-of-the-art and challenges. In *Proceedings of the Fourth AAAI Conference on Weblogs and Social Media (ICWSM), Data Challenge Workshop*.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, pages 1003–1011, Morristown, NJ, USA. Association for Computational Linguistics.

Benjamin Rosenfeld and Ronen Feldman. 2007. Clustering for unsupervised relation identification. In *Proceedings of the ACM Conference on Information and Knowledge Management*, pages 411–418. ACM.

Evan Sandhaus. 2008. The new york times annotated corpus. `http://ldc.upenn.edu/Catalog/docs/LDC2008T19`.

Peter D. Turney. 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, London, UK. Springer-Verlag.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106.

Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. 2009. Statsnowball: a statistical approach to extracting entity relationships. In *Proceedings of the International Conference on World Wide Web*, pages 101–110. ACM.