

A Framework for Entailed Relation Recognition

Dan Roth Mark Sammons V.G.Vinod Vydiswaran

University of Illinois at Urbana-Champaign

{danr|mssammon|vgvinodv}@illinois.edu

Abstract

We define the problem of recognizing entailed relations – given an open set of relations, find **all** occurrences of the relations of interest in a given document set – and pose it as a challenge to scalable information extraction and retrieval. Existing approaches to relation recognition do not address well problems with an open set of relations and a need for high recall: supervised methods are not easily scaled, while unsupervised and semi-supervised methods address a limited aspect of the problem, as they are restricted to frequent, explicit, highly localized patterns. We argue that textual entailment (*TE*) is necessary to solve such problems, propose a scalable TE architecture, and provide preliminary results on an Entailed Relation Recognition task.

1 Introduction

In many information foraging tasks, there is a need to find all text snippets relevant to a target concept. Patent search services spend significant resources looking for prior art relevant to a specified patent claim. Before subpoenaed documents are used in a court case or intelligence data is declassified, all sensitive sections need to be redacted. While there may be a specific domain for a given application, the set of target concepts is broad and may change over time. For these knowledge-intensive tasks, we contend that feasible automated solutions require techniques which approximate an appropriate level of natural language understanding.

Such problems can be formulated as a relation recognition task, where the information need is expressed as tuples of arguments and relations. This structure provides additional information which can be exploited to precisely fulfill the information need. Our work introduces the *Entailed Relation Recognition* paradigm, which leverages a textual entailment system to try to extract *all* relevant passages for a given structured query

without requiring relation-specific training data. This contrasts with Open Information Extraction (Banko and Etzioni, 2008) and On-Demand Information Extraction (Sekine, 2006), which aim to extract large databases of open-ended facts, and with supervised relation extraction, which requires additional supervised data to learn new relations.

Specifically, the contributions of this paper are: 1. Introduction of the *entailed relation recognition* framework; 2. Description of an architecture and a system which uses structured queries and an existing entailment engine to perform relation extraction; 3. Empirical assessment of the system on a corpus of entailed relations.

2 Entailed Relation Recognition (ERR)

In the task of Entailed Relation Recognition, a corpus and an information need are specified. The corpus comprises all text spans (e.g. paragraphs) contained in a set of documents. The information need is expressed as a set of tuples encoding relations and entities of interest, where entities can be of arbitrary type. The objective is to retrieve *all* relevant text spans that a human would recognize as containing a relation of interest. For example:

Information Need: An organization acquires weapons.

Text 1: ...the recent theft of 500 assault rifles by FARC...

Text 2: ...the report on FARC activities made three main observations. First, their allies supplied them with the 3" mortars used in recent operations. Second, ...

Text 3: Amnesty International objected to the use of artillery to drive FARC militants from heavily populated areas.

An automated system should identify Texts 1 and 2 as containing the relation of interest, and Text 3 as irrelevant. The system must therefore detect relation instances that cross sentence boundaries (“them” maps to “FARC”, Text 2), and that require inference (“theft” implies “acquire”, Text 1). It must also discern when sentence structure precludes a match (“Amnesty International... use... heavy artillery” does not imply “Amnesty Interna-

tional acquires heavy artillery”, Text 3).

The problems posed by instances like Text 2 are beyond the scope of traditional unsupervised and semi-supervised relation-extraction approaches such as those used by Open IE and On-Demand IE, which are constrained by their dependency on limited, sentence-level structure and high-frequency, highly local patterns, in which relations are explicitly expressed as verbs and nouns. Supervised methods such as (Culotta and Sorensen, 2004) and (Roth and Yih, 2004) provide only a partial solution, as there are many possible relations and entities of interest for a given domain, and such approaches require new annotated data each time a new relation or entity type is needed. Information Retrieval approaches are optimized for document-level performance, and enhancements like pseudo-feedback (Rocchio, 1971) are less applicable to the localized text spans needed in the tasks of interest; as such, it is unlikely that they will reliably retrieve all correct instances, and not return superficially similar but incorrect instances (such as Text 3) with high rank.

Attempts have been made to apply Textual Entailment in larger scale applications. For the task of Question Answering, (Harabagiu and Hickl, 2006) applied a TE component to rerank candidate answers returned by a retrieval step. However, QA systems rely on redundancy in the same way Open IE does: a large document set has so many instances of a given relation that at least some will be sufficiently explicit and simple that standard IR approaches will retrieve them. A single correct instance suffices to complete the QA task, but does not meet the needs of the task outlined here.

Recognizing relation instances requiring inference steps, in the absence of labeled training data, requires a level of text understanding. A suitable proxy for this would be a successful Textual Entailment Recognition (TE) system. (Dagan et al., 2006) define the task of Recognizing Textual Entailment (RTE) as: *...a directional relation between two text fragments, termed T – the entailing text, and H – the entailed text. T entails H if, typically, a human reading T would infer that H is most likely true.* For relation recognition, the relation triple (e.g. “Organization acquires weapon”) is the hypothesis, and a candidate text span that might contain the relation is the text.

The definition of RTE clearly accommodates the range of phenomena described for the examples above. However, the more successful TE systems (e.g. (Hickl and Bensley, 2007)) are typically resource intensive, and cannot scale to large retrieval tasks if a brute force approach is used.

We define the task of Entailed Relation Recognition thus: *Given a text collection D , and an information need specified in a set of [argument, relation, argument] triples S : for each triple $s \in S$, identify all texts $d \in D$ such that d entails s .*

The information need triples, or queries, encode relations between arbitrary entities (specifically, these are *not* constrained to be Named Entities).

This problem is distinct from recent work in Textual Entailment as we constrain the structure of the Hypothesis to be very simple, and we require that the task be of a significantly larger scale than the RTE tasks to date (which are typically of the order of 800 Text-Hypothesis pairs).

3 Scalable ERR Algorithm

Our scalable ERR approach, *SERR*, consists of two stages: expanded lexical retrieval, and entailment recognition. The SERR algorithm is presented in Fig. 1. The goal is to scale Textual Entailment up to a task involving large corpora, where hypotheses (queries) may be entailed by multiple texts. The task is kept tractable by decomposing TE capabilities into two steps.

The first step, Expanded Lexical Retrieval (*ELR*), uses shallow semantic resources and similarity measures, thereby incorporating some of the semantic processing used in typical TE systems. This is required to retrieve, with high recall, semantically similar content that may not be lexically similar to query terms, to ensure return of a set of texts that are highly likely to contain the concept of interest.

The second step applies a textual entailment system to this text set and the query in order to label the texts as ‘relevant’ or ‘irrelevant’, and requires deeper semantic resources in order to discern texts containing the concept of interest from those that do not. This step emphasizes higher precision, as it filters irrelevant texts.

3.1 Implementation of SERR

In the ELR stage, we use a structured query that allows more precise search and differential query expansion for each query element. Semantic units

| |
|--|
| <p>SERR Algorithm</p> <p>SETUP:</p> <p>Input: Text set D</p> <p>Output: Indices $\{I\}$ over D</p> <p>for all texts $d \in D$</p> <p style="padding-left: 20px;">Annotate d with local semantic content</p> <p>Build Search Indices $\{I\}$ over D</p> <p>APPLICATION:</p> <p>Input: Information need S</p> <p>EXPANDED LEXICAL RETRIEVAL (ELR)(s):</p> <p>$R \leftarrow \emptyset$</p> <p>Expand s with semantically similar words</p> <p>Build search query q_s from s</p> <p>$R \leftarrow k$ top-ranked texts for q_s using indexes $\{I\}$</p> <p>return R</p> <p>SERR:</p> <p>Answer set $A \leftarrow \emptyset$</p> <p>for all queries $s \in S$</p> <p style="padding-left: 20px;">$R \leftarrow \text{ELR}(s)$</p> <p style="padding-left: 20px;">Answer set $A_s \leftarrow \emptyset$</p> <p style="padding-left: 20px;">for all results $r \in R$</p> <p style="padding-left: 40px;">Annotate s, r with NLP resources</p> <p style="padding-left: 40px;">if r entails s</p> <p style="padding-left: 60px;">$A_s \leftarrow A_s \cup r$</p> <p>$A \leftarrow A \cup \{A_s\}$</p> <p>return A</p> |
|--|

Figure 1. SERR algorithm

in the texts (e.g. Named Entities, phrasal verbs) are indexed separately from words; each index is a hierarchical similarity structure based on a type-specific metric (e.g. WordNet-based for phrasal verbs). Query structure is also used to selectively expand query terms using similarity measures related to types of semantic units, including distributional similarity (Lin and Pantel, 2001), and measures based on WordNet (Fellbaum, 1998).

We assess three different Textual Entailment (TE) components: **LexPlus**, a lexical-level system that achieves relatively good performance on the RTE challenges, and two variants of Predicate-based Textual Entailment, **PTE-strict** and **PTE-relaxed**, which use a predicate-argument representation. The former is constrained to select a single predicate-argument structure from each result, which is compared to the query component-by-component using similarity measures similar to the LexPlus system. PTE-relaxed drops the single-predicate constraint, and can be thought of as a ‘bag-of-constituents’ model. In both, features are extracted based on the predicate-argument components’ match scores and their connecting structure, and the rank assigned by the ELR component. These features are used by a classifier that labels each result as ‘relevant’ or ‘irrelevant’. Train-

ing examples are selected from the top 7 results returned by ELR for queries corresponding to entailment pair hypotheses from the RTE development corpora; test examples are similarly selected from results for queries from the RTE test corpora (see section 3.2).

3.2 Entailed Relation Recognition Corpus

To assess performance on the entailed relation recognition task, we derive a corpus from the publicly available RTE data. The corpus consists of a set S of information needs in the form of [argument, relation, argument] triples, and a set D of text spans (short paragraphs), half of which entail one or more $s \in S$ while the other half are unrelated to S . D was generated by taking all 1,950 Texts from the *IE* and *IR* subtasks of the RTE Challenge 1–3 datasets. The shorter hypotheses in these examples allow us to automatically induce their structured query form from their shallow semantic structure. S was automatically generated from the positive entailment pairs in the same subset of RTE data, by annotating them with a publicly available SRL tagger (Punyakanok et al., 2008) and inferring the relation and two main arguments for each hypothesis.

Since some Hypotheses and Texts appear multiple times in the RTE corpora, we automatically extract mappings from positive Hypotheses to one or more Texts by comparing hypotheses and texts from different examples, providing the labeling needed for evaluation. In the resulting corpus, a wide range of relations are sparsely represented; they exemplify many linguistic and semantic characteristics required to infer the presence of non-explicit relations.

4 Results and Discussion

| Top # | Basic | ELR | Rel.Impr. | Err.Redu. |
|-------|-------|-------|-----------|-----------|
| 1 | 48.1% | 55.2% | +14.8% | 13.7% |
| 2 | 68.1% | 72.8% | +6.9% | 14.7% |
| 3 | 75.2% | 78.5% | +4.4% | 17.7% |

Table 1. Change in relevant results retrieved in top 3 positions for basic and expanded lexical retrieval

Table 1 compares the results of SERR with and without the ELR’s semantic enhancements. For each rank k , the entries represent the proportion of queries for which the correct answer was returned in the top k positions. The semantic enhancements

| # System | RTE 1 | RTE 2 | RTE 3 | Avg. Acc. |
|-------------|----------------|----------------|----------------|-----------------------|
| LexPlus | 49.0 | 65.2 [3] | 76.5 [2] | 66.3 |
| PTE-relaxed | 54.5 (1.0) | 68.7 (1.5) [3] | 82.3 (2.0) [1] | 71.2 (1.2) [2] |
| PTE-strict | 64.8 (2.3) [1] | 71.2 (2.6) [3] | 76.0 (3.2) [2] | 71.8 (2.6) [2] |

Table 3. Performance (accuracy) of SERR system variants on RTE challenge examples; numbers in parentheses are standard deviations, while numbers in brackets indicate where systems would have ranked in the RTE evaluations.

| System | Acc. | Prec. | Rec. | F ₁ |
|------------|----------------------|---------------|---------------|----------------------|
| Baseline | 18.1 | 18.1 | 100.0 | 30.7 |
| LexPlus | 81.6 | 44.9 | 62.5 | 55.5 |
| PTE-relax. | 71.9 (0.1) | 37.7 (5.5) | 72.0 (6.2) | 49.0 (4.1) |
| PTE-strict | 83.6 (1.3) | 55.4 (3.4) | 61.5 (7.9) | 57.9 (2.1) |

Table 2. Comparison of performance of SERR with different TE algorithms. Numbers in parentheses are standard deviations.

improve the number of matched results at each of the top 3 positions.

Table 2 compares variants of the SERR implementation. The baseline labels every result returned by ELR as ‘relevant’, giving high recall but low precision. PTE-relaxed performs better than baseline, but poorly compared to PTE-strict and LexPlus. Our analysis shows that LexPlus has a relatively high threshold, and labels as negative some examples mislabeled by PTE-relaxed, which may match two of the three constituents in the hypothesis and label the example as positive. PTE-strict will correctly identify some such examples as it will force some match edges to be ignored, and will correctly identify some negative examples due to structural constraints even when all query terms are matched by result terms. PTE-strict strikes the best balance between precision and recall on positive examples.

Table 3 shows the accuracy of SERR’s classification of the examples from each RTE challenge; results not returned in the top 7 ranks by ELR are labeled ‘irrelevant’. PTE-strict and PTE-relaxed perform comparably overall, though PTE-strict has more uniform results over the different challenges. Both outperform the LexPlus system overall, and perform well compared to the best results published for the RTE challenges.

Table 4 shows the much greater number of comparisons required by a brute force TE approach compared to SERR: SERR performs well compared to published results for RTE challenges 1-3, but makes only 0.36% of the TE comparisons needed by standard approaches on our ERR task.

| | Comparisons |
|-------------|-------------|
| Standard TE | 3,802,500 |
| SERR | 13,650 |

Table 4. Entailment comparisons needed for standard TE vs. SERR

5 Conclusion

We have proposed an approach to solving the Entailed Relation Recognition task, based on Textual Entailment, and implemented a solution that shows that a Textual Entailment Recognition system can be scaled to a much larger IE problem than that represented by the RTE challenges. Our preliminary results demonstrate the utility of the proposed architecture, which allows strong performance in the RTE task and efficient application to a large corpus (table 4).

Acknowledgments

We thank Quang Do, Yuancheng Tu, and Kevin Small. This work is funded by a grant from Boeing and by MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

References

- [Banko and Etzioni2008] M. Banko and O. Etzioni. 2008. The Tradeoffs Between Open and Traditional Relation Extraction. In *ACL-HLT*, pages 28–36.
- [Culotta and Sorensen2004] A. Culotta and J. Sorensen. 2004. Dependency Tree Kernels for Relation Extraction. In *ACL*, pages 423–429.
- [Dagan et al.2006] I. Dagan, O. Glickman, and B. Magnini, editors. 2006. *The PASCAL Recognising Textual Entailment Challenge.*, volume 3944. Springer-Verlag, Berlin.
- [Fellbaum1998] C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- [Harabagiu and Hickl2006] S. Harabagiu and A. Hickl. 2006. Methods for Using Textual Entailment in Open-Domain Question Answering. In *ACL*, pages 905–912.
- [Hickl and Bensley2007] A. Hickl and J. Bensley. 2007. A Discourse Commitment-Based Framework for Recognizing Textual Entailment. In *ACL*, pages 171–176.
- [Lin and Pantel2001] D. Lin and P. Pantel. 2001. Induction of semantic classes from natural language text. In *SIGKDD*, pages 317–322.
- [Punyakanok et al.2008] V. Punyakanok, D. Roth, and W. Yih. 2008. The Importance of Syntactic Parsing and Inference in Semantic Role Labeling. *CL*, 34(2).
- [Rocchio1971] J. Rocchio, 1971. *Relevance feedback in Information Retrieval*, pages 313–323. Prentice Hall.

[Roth and Yih2004] D. Roth and W. Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *CoNLL*, pages 1–8.

[Sekine2006] S. Sekine. 2006. On-Demand Information Extraction. In *COLING/ACL*, pages 731–738.