Knowledge-Guided Linguistic Rewrites for Inference Rule Verification

Prachi Jain and Mausam
Indian Institute of Technology, Delhi
New Delhi, India
{csz148211,mausam}@cse.iitd.ac.in

Abstract

A corpus of inference rules between a pair of relation phrases is typically generated using the statistical overlap of argument-pairs associated with the relations (e.g., PATTY, CLEAN). We investigate knowledge-guided linguistic rewrites as a secondary source of evidence and find that they can vastly improve the quality of inference rule corpora, obtaining 27 to 33 point precision improvement while retaining substantial recall. The facts inferred using cleaned inference rules are 29-32 points more accurate.

1 Introduction

The visions of machine reading (Etzioni, 2007) and deep language understanding (Dorr, 2012) emphasize the ability to draw inferences from text to discover implicit information that may not be explicitly stated (Schubert, 2002). This has natural applications to textual entailment (Dagan et al., 2013), KB completion (Socher et al., 2013), and effective querying over Knowledge Bases (KBs).

One popular approach for fact inference is to use a set of inference rules along with probabilistic models such as Markov Logic Networks (Schoenmackers et al., 2008) or Bayesian Logic Programs (Raghavan et al., 2012) to produce human-interpretable proof chains. While scalable (Niu et al., 2011; Domingos and Webb, 2012), this is bound by the coverage and quality of the background knowledge – the set of inference rules that enable the inference (Clark et al., 2014).

The paper focuses on generating a high precision subset of inference rules over Open Information Extraction (OpenIE) (Etzioni et al., 2011) relation phrases (see Fig 1). OpenIE systems generate a schema-free KB where entities and relations are represented via normalized but not disambiguated textual strings. Such OpenIE KBs scale to the Web.

Most existing large-scale corpora of inference rules are generated using distributional similarity, like argument-pair overlap (Schoenmackers et al., 2010; Berant et al., 2012), but often eschew any linguistic or compositional insights. Our early analysis revealed that such inference rules have very low precision, not enough to be useful for many real tasks. For human-facing applications (such as IE-based demos), high precision is critical. Inference rules have a multiplicative impact, since one poor rule could potentially generate many bad KB facts.

Contributions: We investigate the hypothesis that “knowledge-guided linguistic rewrites can provide independent verification for statistically-generated Open IE inference rules.” Our system KGLR’s rewrites exploit the compositional structure of Open IE relation phrases alongside knowledge in resources like Wordnet and thesaurus. KGLR independently verifies rules from existing inference rule

<table>
<thead>
<tr>
<th>Antecedent</th>
<th>Consequent</th>
<th>Y/N?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X, make a note of, Y)</td>
<td>(X, write down, Y)</td>
<td>Y</td>
</tr>
<tr>
<td>(X, offer wide range of, Y)</td>
<td>(X, offer variety of, Y)</td>
<td>Y</td>
</tr>
<tr>
<td>(X, make full use of, Y)</td>
<td>(Y, be used by, X)</td>
<td>Y</td>
</tr>
<tr>
<td>(X, be wounded in, Y)</td>
<td>(X, be killed in, Y)</td>
<td>N</td>
</tr>
<tr>
<td>(X, be director of, Y)</td>
<td>(X, be vice president of, Y)</td>
<td>N</td>
</tr>
<tr>
<td>(X, be a student at, Y)</td>
<td>(X, be enrolled at, Y)</td>
<td>N</td>
</tr>
</tbody>
</table>

Figure 1: Sample rules verified (Y) and filtered (N) by our method. Rules #4, #5 were correctly and #6 wrongly filtered.
corpora (Berant et al., 2012; Pavlick et al., 2015) and can be seen as additional annotation on existing inference rules. The verified rules are 27 to 33 points more accurate than the original corpora and still retain a substantial recall. The precision of inferred knowledge also has a precision boost of over 29 points. We release our KGLR implementation, its annotations on two popular rule corpora along with gold set used for evaluation and the annotation guidelines for further use (available at https://github.com/dair-itd/kglr.git).

2 Related work

Methods for inference over text include random walks over knowledge graphs (Lao et al., 2011), matrix completion (Riedel et al., 2013), deep neural networks (Socher et al., 2013; Rocktäschel et al., 2015a), natural logic inference (MacCartney and Manning, 2007) and graphical models (Schoenmackers et al., 2008; Raghavan et al., 2012). Most of these need (or benefit from) a background knowledge of inference rules, including matrix completion (Rocktäschel et al., 2015b).

Inference rules are predominantly generated via extended distributional similarity – two phrases having a high degree of argument overlap are similar, and thus candidates for a unidirectional or a bidirectional inference rule. Methods vary on the base representation, e.g., KB relations (Galárraga et al., 2013; Grycner et al., 2015), Open IE relation phrases (Schoenmackers et al., 2010), syntactic-ontological-lexical (SOL) patterns (Nakashole et al., 2012), and dependency paths (Lin and Pantel, 2001). An enhancement is global transitivity (TNCF algorithm) for improving recall (Berant et al., 2012). The highest precision setting of TNCF ($\lambda = 0.1$) was released as a corpus (informally called CLEAN) of Open IE inference rules.\(^1\)

Distributional similarity approaches have two fundamental limitations. First, they miss obvious commonsense facts, e.g., $(X, \text{married}, Y) \rightarrow (X, \text{knows}, Y)$ – text will rarely say that a couple know each other. Second, they are consistently affected by statistical noise and end up generating a wide variety of inaccurate rules (see rules #4, and #5 in Figure 1).

Our early experiments with CLEAN revealed its precision to be about 0.49, not enough to be useful in practice, especially for human-facing applications.

Similar to our paper, some past works have used alternative sources of knowledge. Weisman et al. (2012) study inference between verbs (e.g., $(\text{startle} \rightarrow \text{surprise})$), but they get low (0.4) precision. Wordnet corpus to generate inference rules for natural logic (Angeli and Manning, 2014) improved noun-based inference. But, they recognize relation entailments as a key missing piece. Recently, natural logic semantics is added to a paraphrase corpus (PPDB2.0). Many of their features, e.g., lexical/orthographic, multilingual translation based, are complimentary to our method.

We test our KGLR algorithm on CLEAN and entailment/paraphrase subset of PPDB2.0 (which we call PPDBe).

3 Knowledge-Guided Linguistic Rewrites (KGLR)

Given a rule $(\langle X, r_1, Y \rangle \rightarrow \langle X, r_2, Y \rangle)$ or $(\langle X, r_1, Y \rangle \rightarrow \langle Y, r_2, X \rangle)$ we present KGLR, a series of rewrites of relation phrase $r_1$ to prove $r_2$ (e.g, in Fig 1). The last two rewrites deal with reversal of argument order in $r_2$; others are for the first case.

**Thesaurus Synonyms:** Thesauri typically provide an expansive set of potential synonyms, encompassing near-synonyms and contextually synonymous words. Thesaurus synonyms are not that helpful for generating inference rules (or else we will generate rules like $(\text{produce} \rightarrow \text{percolate})$). However, they are excellent in rule verification as they provide evidence independent from statistical overlap metrics.

We allow any word/phrase $w_1$ in $r_1$ to be replaced by any word/phrase $w_2$ from its thesaurus synsets as long as (1) $w_2$ has same part-of-speech as $w_1$ and (2) $w_2$ is seen in $r_2$ at the same distance from left of the phrase as $w_1$ in phrase $r_1$, but ignoring words dropped due to other rules whose details follows next. To define a thesaurus synset, we tag $w_1$ with its POS and look for all thesaurus synsets of that POS containing $w_1$. We allow this rewrite if $\text{PMI}(w_1, w_2) > \lambda = -2.5$ based on a devset. We calculate PMI as

$$\log \left( \frac{\# w_1 \text{ occurs in synsets of } w_2 + \# w_2 \text{ occurs in synsets of } w_1}{\# \text{ of synsets of } w_1 \times \# \text{ of synsets of } w_2} \right).$$

Some words can be both synonyms and antonyms in different situations. For example, thesaurus lists

‘bad’ as both a synonym and antonym of ‘good’. We don’t allow such antonyms in these rewrites.

Thesaurus synonyms can verify ⟨offer a vast range of → provide a wide range of⟩, since offer-provide, and vast-wide are thesaurus synonyms. We use Roget’s 21st Century Thesaurus in KGLR implementation.

**Negating rules:** We reject rules where \( r_2 \) explicitly negates \( r_1 \) or vice versa. We reject a rule if \( r_2 \) is same as \( r_1 \) if we drop ‘not’ from one of them. For example, the rule ⟨be the president of → be not the president of⟩, will be rejected.

**Wordnet Hypernym:** We replace word/phrase \( w \) in \( r_1 \) by its Wordnet hypernym if it is in \( r_2 \). We prove ⟨be highlight of → be component of⟩, as Wordnet lists ‘component’ as a hypernym of ‘highlight’.

**Dropping Modifiers:** We drop any adjective, adverb, superlatives or comparatives (e.g., ‘more’, ‘most’) from \( r_1 \). This lets us verify ⟨be most important part of → be part of⟩.

**Gerund-Infinitive Equivalence:** We convert infinitive constructions into gerunds and vice versa. For example, ⟨starts to drink ↔ starts drinking⟩.

**Deverbal Nouns:** We use Wordnet’s derivationally related forms to compute a verb-noun pair list. We allow back and forth conversions from “be noun of” to related verb. So, we verify ⟨be cause of → cause⟩.

**Light Verbs and Serial Verbs:** If a light verb precedes a word with derivationally related noun sense, we delete it. Similarly, if a serial verb precedes a word with derivationally related verb sense, we delete it. We identify light verbs via the verbs that frequently precede a ⟨(a|an) (verb|deverbal noun)⟩ pair in Wikipedia. Serial verbs are identified as the verbs that frequently precede another verb in Wikipedia. Thus we can convert ⟨take a look at → look at⟩.

**Preposition Synonyms:** We manually create a list of preposition near-synonyms such as into-to, in-at, at-near. We replace a preposition by its near-synonym. This proves ⟨translated into → translated to⟩.

**Be-Words & Determiners:** We drop be-words (‘is’, ‘was’, ‘be’, etc.) and determiners from \( r_1 \) and \( r_2 \).

**Active-Passive:** We allow \( (X, \text{ verb, } Y) \) to be rewritten as \( (Y, \text{ be verb by, } X) \).

**Redundant Prepositions:** We find that often prepositions other than ‘by’ can be alternatively used with passive forms of some verbs. Moreover, some prepositions can be redundantly used in active forms too. For example, ⟨(X, absorb, Y) ↔ (Y, be absorbed in, X)⟩, or similarly, ⟨(X, attack, Y) ↔ (X, attack on, Y)⟩. To create such a list of verb-preposition pairs, we simply trust the argument-overlap statistics. Statistics here does not make that many errors since the base verb in both relations is the same.

### 3.1 Implementation

KGLR allows repeated application of these rewrites to modify \( r_1 \) and \( r_2 \). If it achieves \( r_1 = r_2 \) it verifies the inference rule. For tractable implementation KGLR uses a depth first search approach where a search node maintains both \( r_1 \) and \( r_2 \). Search does not allow rewrites that introduce any new lexical (lemmatized) entries not in original words\( (r_1) \cup \text{words}(r_2) \). If it can’t apply any rewrite to get a new node, it returns failure.

Many rules are proved by a sequence of rewrites. E.g., to prove \( (X, \text{ be a major cause of, } Y) \rightarrow (Y, \text{ be caused by, } X) \), the proof proceeds as: \( (X, \text{ be a major cause of, } Y) \rightarrow (X, \text{ be major cause of, } Y) \rightarrow (X, \text{ be cause of, } Y) \rightarrow (X, \text{ cause, } Y) \rightarrow (Y, \text{ be caused by, } X) \) by dropping determiner, dropping adjective, deverbal noun, and active-passive transformation respectively. Similarly, \( (X, \text{ helps to protect, } Y) \rightarrow (X, \text{ look after, } Y) \) follows from gerund-infinitive conversion (helps protect), dropping support from serial verbs (protect), and thesaurus synonym (look after).

### 4 Experiments

KGLR verifies a subset of rules from \textit{CLEAN} and \textit{PPDB}_e to produce, \textit{VCLEAN} and \textit{VPPDB}_e. Our experiments answer these research questions: (1) What is the precision and size of the verified subsets compared to original corpora?, (2) How does additional knowledge generated after performing inference using these rules compare with each other? and (3) Which rewrites are critical to KGLR performance?

**Comparison of \textit{CLEAN} and \textit{VCLEAN}:** The original \textit{CLEAN} corpus has about 102K rules. KGLR verifies about 36K rules and filter 66K rules out. To estimate the precisions of \textit{CLEAN} and \textit{VCLEAN} we independently sampled a random subset of 200 inference rules from each and asked two annotators (graduate level NLP students) to label the rules as...
correct or incorrect. Rules were mixed together and the annotators were blind to the system that generated a rule. Our initial annotation guideline was similar to that of textual entailment – label a rule as correct if the consequent can usually be inferred given the antecedent, for most naturally occurring argument-pairs for the antecedent.

Our annotators faced one issue with the guideline – some inference rules were valid if \((X, Y)\) were bound to specific types, but not for others. For example, \(\langle (X, \text{be born in}, Y) \rightarrow (X, \text{be birthplace of}, Y) \rangle\) is valid if \(Y\) is a location, not if \(Y\) is a year. Even seemingly correct inference rules, e.g., \(\langle (X, \text{is the father of}, Y) \rightarrow (Y, \text{is the child of}, X) \rangle\), can make unusual incorrect inferences: (Gandhi, is the father of, India) does not imply (India, is the child of, Gandhi). Unfortunately, these corpora don’t associate argument-type information with their inference rules.

To mitigate this we refined the annotation guidelines to accept inference rules as correct as long as they are valid for some type-pair. The inter-annotator agreement with this modification was 94\(^\circ\) \((\kappa = 0.88)\). On the subset of the tags where the two annotators agreed we find the precision of \textsc{clean} to be 48.9\%, whereas \textsc{vclean} was evaluated to be 82.5\% precise – much more useful for real-world applications. Multiplying the precision with their sizes, we find the effective yield\(^2\) of \textsc{clean} to be 50K compared to 30K for \textsc{vclean}. Overall, we find that \textsc{vclean} obtains a 33 point precision improvement with an effective yield of about 60\%.

\textit{Error Analysis:} Most of \textsc{vclean} errors are due to erroneous (or unusual) thesaurus synonyms. For missed recall, we analyzed \textsc{clean}’s sample missed by \textsc{vclean}. We find that only about 13\% of those are world knowledge rules (e.g., rule #6 in Figure 1). Other missed recall is because of some missing rewrites, missing thesaurus synonymy, spelling mistakes. These can potentially be captured by using other resources and adding rewrite rules.

\textbf{Comparison of \textsc{PPDB\textsubscript{e}}, and \textsc{VPPDB\textsubscript{e}}:} Unlike \textsc{clean}, \textsc{PPDB\textsubscript{2.0}} associates a confidence value for each rule, which can be varied to obtain different levels of precision and yield. We control for yield so that we can compare precisions directly.

We operate on \textsc{PPDB\textsubscript{e}} subset that has an Open IE-like relation phrase on both sides; this was identified by matching to ReVerb syntactic patterns (Etzioni et al., 2011). This subset is of size 402K. \textsc{KGLR} on this produces 85K verified rules (\textsc{VPPDB\textsubscript{e}}). We find the threshold for confidence values in \textsc{PPDB\textsubscript{e}} that achieves the same yield (confidence \(> 0.342\)).

We perform annotation on \textsc{PPDB\textsubscript{e}}(0.342) and \textsc{VPPDB\textsubscript{e}} using same annotation guidelines as before. The inter-annotator agreement was 91\% \((\kappa = 0.82)\). On the subset of the tags where the two annotators agreed we find the precision of \textsc{PPDB\textsubscript{e}} to be low – 44.2\%, whereas \textsc{VPPDB\textsubscript{e}} was evaluated to be 71.4\% precise. We notice that about 4 in 5 \textsc{PPDB\textsubscript{e}} relation phrases are of length 1 or 2 (whereas 50\% of \textsc{clean} relation phrases are of length \(\geq 3\)). This contributes to a slightly lower precision of \textsc{VPPDB\textsubscript{e}}, as most rules are proved by thesaurus synonymy and the power of \textsc{KGLR} to handle compositionality of longer relation phrases does not get exploited.

\textbf{Comparison of Inferred Facts:} A typical use case of inference rules is in generating new facts by applying inference rules to a KB. We independently apply \textsc{vclean}’s, \textsc{clean}’s, \textsc{PPDB\textsubscript{2.0}}’s and \textsc{VPPDB\textsubscript{e}}’s inference rules on a public corpus of 4.2 million ReVerb triples.\(^3\) Since ReVerb itself has significant extraction errors (our estimate is 20\% errors) and our goal is to evaluate the quality of inference, we restrict this evaluation to only the subset of accurate ReVerb extractions.

\textsc{Vclean and clean facts:} We sampled about 200 facts inferred by \textsc{vclean} rules and \textsc{clean} rules each (applied over accurate ReVerb extractions) and gave the original sentence as well as inferred facts to two annotators. We obtained a high inter-annotator agreement of 96.3\%\((\kappa = 0.92)\) and we discarded disagreements from final analysis. Overall, facts inferred by \textsc{clean} achieved a precision of about 49.1\% and those inferred by \textsc{vclean} obtained a 81.6\% precision. The estimated yields of fact corpora (precision \(\times\) size) are 7 and 4.5 million for \textsc{clean} and \textsc{vclean} respectively. This yield estimate does not include the initial 4.2 million facts.

\textsc{PPDB\textsubscript{e}} and \textsc{VPPDB\textsubscript{e}} facts: As done previously, we sampled 200 facts inferred by \textsc{PPDB\textsubscript{e}} and \textsc{VPPDB\textsubscript{e}} rules, which were annotated by two annotators. We obtained a good inter annotator agree-

\(^2\)Yield is proportional to recall

\(^3\)Available at http://reverb.cs.washington.edu
## 5 Discussion

KGLR’s value is in precision-sensitive tasks such as a human-facing demo, or downstream NLP application (like question answering) where error multiplication is highly undesirable. Along with high precision, it still obtains acceptably good yield.

Our annotators observe the importance of type-restriction of arguments for inference rules (similar to rules in (Schoenmackers et al., 2010)). Type annotation of existing inference rule corpora is an important step for obtaining high precision and clarity.

Inference rules are typically of two types – linguistic/synonym rewrites, which are captured by our work, and world knowledge rules (see rule #6 in Fig 1), which are not. We were surprised to estimate that about 87% of CLEAN, which is a statistically-generated corpus, is just linguistic rewrites! Obtaining world knowledge or common-sense rules at high precision and scale continues to be the key NLP challenge in this area.

## 6 Conclusions

We present Knowledge-guided Linguistic Rewrites (KGLR) which exploits the compositionality of relation phrases, guided by existing knowledge sources, such as Wordnet and thesaurus to identify a high precision subset of an inference rule corpus. Validated CLEAN has a high precision of 83% (vs 49%) at a yield of 60%. Validated PPDB has a precision of 71% (vs 44%) at same yield. The precision of inferred facts has about 29-32 pt precision gain. We expect KGLR to be effective for precision-sensitive applications of inference. The complete code and data has been released for the research community.

## Acknowledgments

We thank Ashwini Vaidya and the anonymous reviewers for their helpful suggestions and feedback. We thank Abhishek, Aditya, Ankit, Jatin, Kabir, and Shikhar for helping with the data annotation. This work was supported by Google language understanding and knowledge discovery focused research grants to Mausam, a KISTI grant and a Bloomberg grant also to Mausam. Prachi was supported by a TCS fellowship.
References


Peter Clark, Niranjan Balasubramanian, Sumithra Bhavathsalam, Kevin Humphreys, Jesse Kinkead, Ashish Sabharwal, and Oyvind Tafjord. 2014. Automatic construction of inference-supporting knowledge bases.


Dekang Lin and Patrick Pantel. 2001. Dirt@ sbt@ discovery of inference rules from text. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 323–328. ACM.


