SherLIiC: A Typed Event-Focused Lexical Inference Benchmark for Evaluating Natural Language Inference

Martin Schmitt and **Hinrich Schütze**
Center for Information and Language Processing (CIS)
LMU Munich, Germany
martin@cis.lmu.de

Abstract

We present SherLIiC,[1] a testbed for lexical inference in context (LIiC), consisting of 3985 manually annotated *inference rule candidates* (InfCands), accompanied by (i) ~960k unlabeled InfCands, and (ii) ~190k *typed textual relations* between Freebase entities extracted from the large entity-linked corpus ClueWeb09. Each InfCand consists of one of these relations, expressed as a lemmatized dependency path, and two argument placeholders, each linked to one or more Freebase types. Due to our candidate selection process based on strong distributional evidence, SherLIiC is much harder than existing testbeds because distributional evidence is of little utility in the classification of InfCands. We also show that, due to its construction, many of SherLIiC's correct InfCands are novel and missing from existing rule bases. We evaluate a number of strong baselines on SherLIiC, ranging from semantic vector space models to state of the art neural models of natural language inference (NLI). We show that SherLIiC poses a tough challenge to existing NLI systems.

1 Introduction

Lexical inference (LI) can be seen as a focused variant of natural language inference (NLI), also called recognizing textual entailment (Dagan et al., 2013). Recently, Gururangan et al. (2018) showed that annotation artifacts in current NLI testbeds distort our impression of the performance of state of the art systems, giving rise to the need for new evaluation methods for NLI. Glockner et al. (2018) investigated LI as a way of evaluating NLI systems and found that even simple cases are challenging to current systems. In this paper, we release SherLIiC, a testbed specifically designed for evaluating a system's ability to solve the hard problem of modeling lexical entailment in context.

[1] https://github.com/mnschmit/SherLIiC

(1)	troponymy	ORGF[A] is granting to EMPL[B] ⇒ ORGF[A] is giving to EMPL[B]
(2)	synonymy + derivation	ORGF[A] is supporter of ORGF[B] ⇒ ORGF[A] is backing ORGF[B]
(3)	typical actions	AUTH[A] is president of LOC[B] ⇒ AUTH[A] is representing LOC[B]
(4)	script knowledge	PER[A] is interviewing AUTH[B] ⇒ PER[A] is asking AUTH[B]
(5)	common sense knowledge	ORGF[A] claims LOC[B] ⇒ ORGF[A] is wanting LOC[B]

Table 1: Examples of SherLIiC InfCands and NLI challenges they cover. ORGF=organization founder, EMPL=employer, AUTH=book author, LOC=location, POL=politician, PER=person.

Levy and Dagan (2016) identified context-sensitive – as opposed to "context-free" – entailment as an important evaluation criterion and created a dataset for LI in context (LIiC). In their data, WordNet (Miller, 1995; Fellbaum, 2005) synsets serve as context for one side of a binary relation, but the other side is still instantiated with a single concrete expression. We aim to improve this setting in two ways.

First, we type our relations on *both sides*, thus making them more general. Types provide a context that can help in disambiguation and at the same time allow generalization over contexts because arguments of the same type are represented abstractly. An example for the need for disambiguation is the verb "run". "run" entails "lead" in the context of PERSON / COMPANY ("Bezos runs Amazon"). But in the context of COMPUTER / SOFTWARE, "run" entails "execute"/"use" ("my mac runs macOS"). Here, types help find the right interpretation.

Second, *we only consider relations between named entities* (NEs). Inference mining based on non-NE types such as WordNet synsets (e.g., ANIMAL, PLANT LIFE) primarily discovers *facts* like "parrotfish feed on algae". In contrast, the focus

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 902–914
Florence, Italy, July 28 - August 2, 2019.

on NEs makes it more likely that we will capture *events* like "Walmart closes gap with Amazon" and thus knowledge about event entailment like ["A is closing gap with B" $\Rightarrow$ "B is having lead over A"] that is substantially different from knowledge about general facts.

In more detail, we create SherLIiC as follows. First, we extract verbal relations between Freebase (Bollacker et al., 2008) entities from the entity-linked web corpus ClueWeb09 (Gabrilovich et al., 2013).[2] We then divide these relations into typable subrelations based on the most frequent Freebase types found in their extensions. We then create a large set of inference rule candidates (InfCands), i.e., premise-hypothesis-pairs of verbally expressed relations. Finally, we use Amazon Mechanical Turk to classify each InfCand in a randomly sampled subset as *entailment* or *non-entailment*.

In summary, our contributions are the following: (1) We create SherLIiC, a new resource for LIiC, consisting of 3985 manually annotated InfCands. Additionally, we provide ~960k unlabeled InfCands (SherLIiC-InfCands), and the typed event graph SherLIiC-TEG, containing ~190k typed textual binary relations between Freebase entities. (2) SherLIiC is harder than existing testbeds because distributional evidence is of limited utility in the classification of InfCands. Thus, SherLIiC is a promising and challenging resource for developing NLI systems that go beyond shallow semantics. (3) Human-interpretable knowledge graph types serve as context for both sides of InfCands. This makes InfCands more general and boosts the number of event-like relations in SherLIiC. (4) SherLIiC is complementary to existing collections of inference rules as evidenced by the low recall these resources achieve (cf. Table 3). (5) We evaluate a large number of baselines on SherLIiC. The best-performing baseline makes use of typing. (6) We demonstrate that existing NLI systems do poorly on SherLIiC.

2 Generation of InfCands

This section describes creation (§ 2.1) and typing (§ 2.2) of the typed event graph SherLIiC-TEG and then the generation of SherLIiC-InfCands (§ 2.3).

2.1 Relation Extraction

For each sentence s in ClueWeb09 that contains at least two entity mentions, we use MaltParser (Nivre et al., 2007) to generate a dependency graph, where nodes are labeled with their lemmas and edges with dependency types. We take all shortest paths between all combinations of two entities in s and represent them by alternating edge and node labels. As we want to focus on relations that express events, we only keep paths with a nominal subject on one end. We also apply heuristics to filter out erroneous parses. See Appendix A for heuristics and Table 5 for examples of relations.

Notation. Let $\mathcal{R}$ denote the set of extracted relations. A relation $R \in \mathcal{R}$ is represented as a set of pairs of Freebase entities (its extension): $R \subseteq \mathcal{E} \times \mathcal{E}$, with $\mathcal{E}$ the set of Freebase entities. Let π_1, π_2 be functions that map a pair to its first or second entry, respectively. By abuse of notation, we also apply them to sets of pairs. Finally, let $\mathcal{T}$ be the set of Freebase types and $\tau \colon \mathcal{E} \to 2^{\mathcal{T}}$ the function that maps an entity to the set of its types.

2.2 Typing

We define a *typable subrelation* of $R \in \mathcal{R}$ as a subrelation whose entities in each argument slot share at least one type, i.e., an $S \subseteq R$ such that:

$$\forall i \in \{1, 2\} : \exists t \in \mathcal{T} \colon t \in \bigcap_{e \in \pi_i(S)} \tau(e)$$

We compute the set $\mathrm{Type}_{k^2}(R)$ of the (up to) k^2 largest typable subrelations of R and use them instead of R. First, for each argument slot i of the binary relation R, the k types t_j^i (with $1 \leq j \leq k$) are computed that occur most often in this slot:

$$t_j^i := \arg\max_t \left| \{ p \in R \mid t \in \tau_j^i(\pi_i(p)) \} \right|$$

with

$$\tau_1^i(e) = \tau(e)$$
$$\tau_{j+1}^i(e) = \tau_j^i(e) - \{t_j^i\}$$

Then, for each pair

$$(s, u) \in \{ (t_j^1, t_l^2) \mid j, l \in \{1, \ldots, k\} \}$$

of these types, we construct a subrelation

$$R_{s,u} := \{ (e_1, e_2) \in R \mid s \in \tau(e_1), u \in \tau(e_2) \}$$

If $|R_{s,u}| \geq r_{\text{min}}$, $R_{s,u}$ is included in $\mathrm{Type}_{k^2}(R)$. In our experiments, we set $k = r_{\text{min}} = 5$.

[2] http://lemurproject.org/clueweb09

The type signature (tsg) of a typed relation T is defined as the pair of sets of types that is common to first (resp. second) entities in the extension:

$$\operatorname{tsg}(T) = \left(\bigcap_{e \in \pi_1(T)} \tau(e), \bigcap_{e \in \pi_2(T)} \tau(e) \right)$$

Incomplete type information. Like all large knowledge bases, Freebase suffers from incompleteness: Many entities have no type. To avoid losing information about relations associated with such entities, we introduce a special type $\top$ and define $\arg\max_t |\emptyset| := \top$. We define the relations $R_{s,\top}$, $R_{\top,u}$ and $R_{\top,\top}$ to have no type restriction on entities in a $\top$ slot. This concerns approximately 17.6% of the relations in SherLIiC-TEG.

2.3 Entailment Discovery

Our discovery procedure is based on Sherlock (Schoenmackers et al., 2010). For the InfCand $A \Rightarrow B$ ($A, B \in \mathcal{R}$), we define the relevance score Relv, a metric expressing Sherlock's stat. relevance criterion $P(B \mid A) \gg P(B)$ (cf. Salmon, 1971).

$$\operatorname{Relv}(A, B) := \frac{P(B \mid A)}{P(B)} = \frac{|A \cap B| \, |\mathcal{E} \times \mathcal{E}|}{|A| \, |B|}$$

Our significance score $\sigma(A, B)$ is a scaled version of the significance test lrs used by Sherlock:

$$P(B \mid A) \operatorname{lrs}(A, B) = \frac{|A \cap B| \operatorname{lrs}(A, B)}{|A|}$$

with $\operatorname{lrs}(A, B)$ (likelihood ratio statistic) defined as

$$2 \, |A| \sum_{H \in \{B, \neg B\}} P(H \mid A) \log(\operatorname{Relv}(A, H)).$$

Additionally, we introduce the *entity support ratio*:

$$\operatorname{esr}(A, B) := \frac{\left| \bigcup_{i \in \{1,2\}} \pi_i(A \cap B) \right|}{2 \, |A \cap B|}$$

This score measures the diversity of entities in $A \cap B$. We found that many InfCands involve a few frequent entities and so obtain high Relv and σ scores even though the relations of the rule are semantically unrelated. esr penalizes such InfCands.

We apply our three scores defined above to all possible pairs of relations $(A, B) \in \mathcal{R} \times \mathcal{R}$ and accept a rule iff all of the following criteria are met:

1. $\forall i \in \{1, 2\} : \pi_i(\operatorname{tsg}(A \Rightarrow B)) \neq \emptyset$
2. $|A \cap B| \geq r_{\min}$

Fact: location[B] is annexing location[A] .
Examples for location[B]: *Russia / USA / Indonesia*
Examples for location[A]: *Cuba / Algeria / Crimea*

☐ fact incomprehensible

Please answer the following questions:

Is it certain that location[B] is taking control of location[A]? ○ yes ○ no ○ incomprehensible
Is it certain that location[B] is taking location[A]? ○ yes ○ no ○ incomprehensible
Is it certain that location[B] is bordered by location[A]? ○ yes ○ no ○ incomprehensible

Figure 1: Annotation Interface on Amazon MTurk

3. $\forall i \in \{1, 2\} : |\pi_i(A \cap B)| \geq r_{\min}$
4. $\operatorname{Relv}(A, B) \geq \vartheta_{relv}$
5. $\sigma(A, B) \geq \vartheta_\sigma$
6. $\operatorname{esr}(A, B) \geq \vartheta_{\text{esr}}$

where $\operatorname{tsg}(A \Rightarrow B)$ is component-wise intersection of $\operatorname{tsg}(A)$ and $\operatorname{tsg}(B)$ and $\vartheta_{relv} = 1000$, $\vartheta_\sigma = 15$, $\vartheta_{\text{esr}} = 0.6$. We found these numbers by randomly sampling InfCands and inspecting their scores. Typing lets us set these thresholds higher, benefitting the quality of SherLIiC-InfCands.

Lastly, we apply Schoenmackers et al. (2010)'s heuristic to only accept the 100 best-scoring premises for each hypothesis. For each hypothesis B, we rank all possible premises A by the product of the three scores and filter out cases where A and B only differ in their types.

3 Crowdsourced Annotation

SherLIiC-InfCands contains ~960k InfCands. We take a random sample of size 5267 and annotate it using Amazon Mechanical Turk (MTurk).

3.1 Task Formulation

We are asking our workers the same kind of questions as Levy and Dagan (2016) did. We likewise form batches of sentence pairs to reduce annotation cost. Instead of framing the task as judging the appropriateness of answers, however, we state the premise as a fact and ask workers about its entailed consequences, i.e., we ask for each candidate hypothesis whether it is *certain* that it is also true. Fig. 1 shows the annotation interface schematically.

We use a morphological lexicon (XTAG Research Group, 2001) and form the present tense of a dependency path's predicate. If a sentence is flagged incomprehensible (e.g., due to a parse error), it is excluded from further evaluation.

Annotated Subset of SherLIiC-InfCands	
Validated InfCands	3985
Balance yes/no	33% / 67%
Pairs with unanimous gold label	53.0%
Pairs with 1 disagreeing annotation	27.4%
Pairs with 2 disagreeing annotations	19.6%
Individual label = gold label	86.7%

Table 2: Statistics for crowd-annotated InfCands. The gold label is the majority label.

We put premise and hypothesis in the present (progressive if suitable) based on the intuition that a pair is only to be considered an entailment if the premise makes it necessary for the hypothesis to be true *at the same time*. This condition of simultaneity follows the tradition of datasets such as SNLI (Bowman et al., 2015) – in contrast to more lenient evaluation schemes that consider a rule to be correct if the hypothesis is true at any time before or after the reference time of the premise (cf. Lin and Pantel, 2001; Lewis and Steedman, 2013).

3.2 Annotation Quality

We imposed several qualification criteria on crowd workers: number of previous jobs on MTurk, overall acceptance rate and a test that each worker had to pass. Some workers still had frequent low agreement with the majority. However, in most cases we obtained a clear majority annotation. These annotations were then used to automatically detect workers with low *trust*, where *trust* is defined as the ratio of submissions agreeing with the majority answer and a worker's total number of submissions. We excluded workers with a *trust* of less than 0.8 and collected replacement annotations until we had at least five annotations per InfCand.

Table 2 shows that workers agreed unanimously for 53% and that the maximal number of two disagreements only occurs for 19.6%. The high number of times an individual agrees with the gold label suggests that humans can do the task reliably. Interestingly, the number of disagreements is not evenly distributed among the two classes *entailment*/*non-entailment*. If the majority agrees on *entailment*, it is comparatively much more likely that at least one of the workers disagrees (cf. Fig. 2). This suggests that our workers were strict and keen on achieving high precision in their annotations.

4 Baselines

We split our annotated data 25:75 into SherLIiC-dev and SherLIiC-test, stratifying on annotated la-

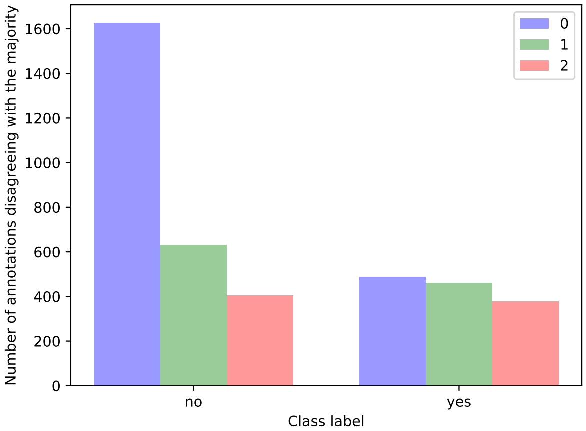

Figure 2: Number of disagreements per class label on all annotations.

bel (*yes*/*no*) and number of disagreements with the majority (0/1/2). If unanimity of annotations marks particularly clear cases, we figure they should be evenly distributed on dev and test.

To establish the state of the art for SherLIiC, we evaluate a number of baselines. Input to these baselines are either the dependency paths or the sentences that were presented to the crowd workers. Baselines that require a threshold to be used as binary classifiers are tuned on SherLIiC-dev.

Lemma baseline. Following Levy and Dagan (2016), this baseline classifies an InfCand as valid if the following holds true for the premise p and hypothesis h after lemmatization: (1) p contains all of h's content words,[3] (2) p's and h's predicates are identical, and (3) the relations' active/passive voice matches their arguments' alignment.

Rule collection baselines. `Berant I` (Berant et al., 2011) and `Berant II` (Berant, 2012)[4] are entailment graphs. `PPDB` is the largest collection (XXXL) of PPDB 2.0 (Pavlick et al., 2015).[5] `Patty` is a collection of relational patterns, consisting of ontological types, POS placeholders and words. We use the version extracted from Wikipedia with Freebase types (Nakashole et al., 2012). `Schoenmackers` is the rule collection released by Schoenmackers et al. (2010). `Chirps` is an ever-growing[6] predicate paraphrase database extracted via event coreference in news Tweets (Shwartz et al., 2017b). `All Rules` denotes the union of all of these rule bases. For rules with type (or POS) constraints, we ignore these constraints

[3] We use the stop word list of `nltk` (Loper and Bird, 2002).
[4] We use default threshold 0.
[5] We ignore stop words and punctuation for phrases.
[6] We use the version downloaded on May 28, 2019.

to boost recall. We will see that even with these recall-enhancing measures, the majority of our correct InfCands is not covered by existing rule bases.

Word2vec baselines. `word2vec` is based on Mikolov et al. (2013b)'s pre-trained word embeddings of size 300. We average them to obtain a vector representation of relations consisting of multiple words and use cosine similarity to judge a relation pair. `typed_rel_emb` (resp. `untyped_rel_emb`) is obtained by training word2vec skip-gram (Mikolov et al., 2013a) with vector size 300 and otherwise default parameters on a synthetic corpus representing the extensions of typed (resp. untyped) relations. The corpus is obtained by writing out one entity-relation-entity-triple per line, where each entity is prefixed with the argument slot it belongs to. `w2v+typed_rel` (resp. `w2v+untyped_rel`) produces its score by summing the scores of `word2vec` and `typed_rel_emb` (resp. `untyped_rel_emb`).

Some type signatures (tsgs) benefit more from type-informed methods than others. For example, the correct inference [INFLUENCER *is explaining in* WRITTEN_WORK ⇒ INFLUENCER *is writing in* WRITTEN_WORK] is detected by `w2v+typed_rel`, but not by `w2v+untyped_rel`. We therefore combine these two methods by using, for each tsg, the method that works better for that tsg on dev. (For tsgs not occurring in dev, we take the method that works better for the individual types occurring in the tsg. We use untyped embeddings if all else fails.) We refer to this combination as `w2v+tsg_rel_emb`.

Knowledge graph embedding baselines. As SherLIiC-TEG has the structure of a knowledge graph (KG), we also evaluate the two KG embedding methods `TransE` (Bordes et al., 2013) and `ComplEx` (Trouillon et al., 2016), as provided by the OpenKE framework (Han et al., 2018).

Asymmetric baselines. Entailment models built upon cosine similarity are symmetric whereas entailment is not. Therefore many asymmmetric measures based on the distributional inclusion hypothesis have been proposed (Kotlerman et al., 2010; Santus et al., 2014; Shwartz et al., 2017a; Roller et al., 2018). We consider `WeedsPrec` (Weeds et al., 2004) and `invCL` (Lenci and Benotto, 2012), which have strong empirical results on hypernym detection. We use cooccurrence counts with entity pairs as distributional representation of a relation.

Supervised NLI models. As LIiC is a special case of NLI, our dataset can also be used to evaluate the generalization capabilities of supervised models trained on large NLI datasets. We pick `ESIM` (Chen et al., 2017), a state-of-the-art supervised NLI model, trained on MultiNLI (Williams et al., 2018) as provided by the framework Jack the Reader (Weissenborn et al., 2018). Input to `ESIM` are the sentences from the annotation process with placeholders instantiated by entities randomly picked from the example lists that had also been shown to the crowd workers (cf. Fig. 1). As we want to measure `ESIM`'s capacity to detect entailment, we map its prediction of both *neutral* and *contradiction* to our *non-entailment* class.

Sherlock+ESR. We also evaluate the candidate scoring method inspired by Schoenmackers et al. (2010) that created the data in the first place. We again combine the three scores described in § 2 by multiplication. The low performance of `Sherlock+ESR` (cf. Table 3) is evidence that the dataset is not strongly biased in its favor and thus is promising as a general evaluation benchmark.

5 Experimental Results and Discussion

Quantitative observations. Table 3 summarizes the performance of our baselines on predicting the *entailment* class for SherLIiC-dev and -test.

Rule collections (lines 1–6) have recall between 0.119 and 0.308; the recall of their union (line 7) is only 0.483 on dev and 0.493 on test, showing that we found indeed new valid inferences missing from existing rule bases.

The state-of-the-art neural model `ESIM` does not generalize well from MultiNLI (its training set) to LIiC. In fact, it hardly improves on the baseline that always predicts *entailment* (`Always yes`). Our dataset was specifically designed to only contain good InfCands based on distributional features. So it poses a challenge to models that cannot make the fine semantic distinctions necessary for LIiC.

Turning to vector space models (lines 11–24), dense relation representations (lines 12, 13) predict entailment better than sparse models (lines 17–20) although they cannot use asymmetric measures.

KG embeddings (lines 21–24) do not seem at all appropriate for measuring the similarity of relations. First, their performance is very close to `Always yes`. Second, their F1-optimal thresholds are very low – even negative. This suggests that their relation vectors do not contain any helpful

	Baseline	θ^*	dev			test		
			P	R	F1	P	R	F1
1	`Berant I`	–	0.699	0.154	0.252	0.762	0.126	0.216
2	`Berant II`	–	0.800	0.181	0.296	0.774	0.186	0.300
3	`PPDB`	–	0.631	0.211	0.317	0.621	0.240	0.347
4	`Patty`	–	0.795	0.187	0.303	0.779	0.153	0.256
5	`Schoenmackers`	–	0.780	0.139	0.236	0.849	0.119	0.208
6	`Chirps`	–	0.370	0.308	0.336	0.341	0.295	0.316
7	`All Rules`	–	0.418	0.483	0.448	0.404	0.493	0.444
8	`Lemma`	–	**0.900**	0.109	0.194	**0.907**	0.089	0.161
9	`Always yes`	–	0.332	**1.000**	0.499	0.333	**1.000**	0.499
10	`ESIM`	–	0.391	0.831	0.532	0.390	0.833	0.531
11	`word2vec`	0.321	0.556	0.625	0.589	0.520	0.606	0.559
12	`typed_rel_emb`	0.864	0.561	0.568	0.565	0.532	0.486	0.508
13	`untyped_rel_emb`	0.613	0.511	0.740	0.605	0.499	0.672	0.572
14	`w2v+typed_rel`	1.106	0.549	0.710	0.619	0.523	0.688	0.594
15	`w2v+untyped_rel`	0.884	0.565	0.740	0.641	0.528	0.695	0.600
16	`w2v+tsg_rel_emb`	0.884	0.566	0.776	**0.655**	0.518	0.727	**0.605**
17	`WeedsPrec (typed)`	0.073	0.335	0.994	0.501	0.333	0.988	0.498
18	`WeedsPrec (untyped)`	0.057	0.403	0.807	0.538	0.386	0.783	0.517
19	`invCL (typed)`	0.000	0.332	**1.000**	0.499	0.333	**1.000**	0.499
20	`invCL (untyped)`	0.148	0.362	0.876	0.512	0.357	0.863	0.505
21	`TransE (typed)`	-0.922	0.336	**1.000**	0.503	0.333	0.991	0.498
22	`TransE (untyped)`	-0.476	0.340	0.964	0.503	0.332	0.942	0.491
23	`ComplEx (typed)`	-0.033	0.339	0.955	0.500	0.337	0.949	0.497
24	`ComplEx (untyped)`	-0.030	0.340	0.952	0.501	0.334	0.939	0.493
25	`Sherlock+ESR`	$9.460 \cdot 10^5$	0.504	0.592	0.544	0.491	0.526	0.508

Table 3: Precision, recall and F1 score on SherLIiC-dev and -test. All baselines run on top of `Lemma`. Thresholds (θ^*) are F1-optimized on dev. Best result per column is set in bold.

information for the task. These methods were not developed to compare relations; the lack of useful information is still surprising and thus is a promising direction for future work on KG embeddings.

General purpose dense representations (`word2vec`, line 11) perform comparatively well, showing that, in principle, they cover the information necessary for LIiC. Embeddings trained on our relation extensions SherLIiC-TEG (line 13), however, can already alone achieve better performance than word2vec embeddings alone.

In general, type-informed relation embeddings seem to have a disadvantage compared to unrestricted ones (e.g., cf. lines 12 and 13) – presumably because type-informed baselines have training sets that are smaller (due to filtering) and sparser (since relations are split up according to type signatures). The combination of general word2vec and specialized relation embeddings (lines 14–16), however, consistently brings gains. This indicates that distributional word properties are complementary to the relation extensions our method extracts. So using both sources of information is promising for future research on modeling relational semantics.

`w2v+tsg_rel_emb` is the best-performing method. It combines typed and untyped relation embeddings as well as general-purpose word2vec embeddings. Even though one cannot rely on typed extensions only, this shows that incorporating type information is beneficial for good performance.

We use `w2v+tsg_rel_emb` to provide a noisy annotation for SherLIiC-InfCands. This is a useful resource because learning from noisy labels has been well studied (Frénay and Verleysen, 2014; Hendrycks et al., 2018) and is often beneficial.

Qualitative observations. Although SherLIiC's creation is based on the same method that was used to create `Schoenmackers`, SherLIiC is fundamentally different for several reasons: (1) The rule sets are complementary (cf. the low recall of 0.139 and 0.119 in Table 3). (2) The majority of rules in `Schoenmackers` has more than one premise, leaving only ~13k InfCands in `Schoenmackers` compared to ~960k in SherLIiC-InfCands that fit the format of NLI. (3) `Schoenmackers` is filtered more aggressively with the goal of maximizing the number of correct rules. This, however, makes it inadequate as a challenging benchmark be-

(1)	PERSON[A] is REGION[B]'s ruler ⇒ PERSON[A] is dictator of REGION[B]
(2)	LOCATION[A] is fighting with ORGF[B] ⇒ LOCATION[A] is allied with ORGF[B]
(3)	ORGF[A] is coming into LOCATION[B] ⇒ ORGF[A] is remaining in LOCATION[B]
(4)	ORGF[A] is seeking from ORGF[B] ⇒ ORGF[B] is giving to ORGF[A]
(5)	LOCATION[A] is winning war against LOCATION[B] ⇒ LOCATION[A] is declaring war on LOCATION[B]

Table 4: False positives for each of the three best-performing baselines taken from SherLIiC-dev. ORGF=organization founder.

cause the performance of `Always yes` would be close to 100%. (4) SherLIiC is focused on events. When linking the relations from SherLIiC-TEG back to their surface forms in the corpus, 80% of them occur at least once in the progressive, which suggests that the large majority of our relations indeed represent events.

Taking a closer look at SherLIiC, we see that the data require a large variety of lexical knowledge even though their creation has been entirely automatic. Table 1 shows five positively labeled examples from SherLIiC-dev, each highlighting a different challenge for statistical models that is crucial for NLI. (1) is an instance of *troponymy*: "granting" is a manner or kind of "giving". This is the verbal equivalent to nominal hyponymy. (2) combines synonymy ("support" ⇔ "back") with morph. derivation. (3) can only be classified correctly if one knows that it is one of the typical actions of a president to represent their country. (4) requires knowledge about the typical course of events when interviewing someone. A typical interview involves asking questions. (5) can only be detected with common sense knowledge that goes even beyond that: you generally only claim something if you want it.

An error analysis of the three best-performing baselines (lines 14–16 in Table 3) reveals that none of them was able to detect the five correct InfCands from Table 1. Explicit modeling of one of the phenomena described above seems a promising direction for future research to improve recall. Table 4 shows five cases where InfCands were incorrectly labeled as *entailment*. (1) shows the importance of modeling directionality: every "dictator" is a "ruler" but not vice versa. (2) shows a well-known problem in representation learning from cooccurrence: antonyms tend to be close in the embedding space (Mohammad et al., 2008; Mrkšić et al., 2016). The other examples show other types of correlation that models relying entirely on distributional information will fall for: the outcome of events like "coming into a country" or "seeking something from someone" are in general uncertain although possible outcomes like "remaining in said country" (3) or "being given the object of desire" (4) will be highly correlated with them. Finally, better models will also take into account the simultaneity constraint: "winning a war" and "declaring a war" (5) rarely happen at the same time.

nsubj–X–prep–of–obj *A is an **ally** of B*	⇔	nsubj–X–poss *A is B's **ally***
nsubj–X–prep–in–obj *A is the **capital** in B*	⇔	nsubj–X–poss *A is B's **capital***
nsubjpass–X–prep–by–obj *A is **followed** by B*	⇔	obj–X–nsubj *B **follows** A*
nsubj–one–prep–of–obj–X–obj *A is one of the **countries in** B*	⇔	nsubj–X–obj *A is a **country in** B*
nsubj–capital–conj–X–obj *A is the capital and biggest **city in** B*	⇒	nsubj–X–obj *A is a **city in** B*
nsubj–Xer–prep–of–obj *A is a **teach**er of B*	⇔	nsubj–X–obj A **teaches** B
nsubj–co-Xer–prep–of–obj *A is a co-**found**er of B*	⇒	nsubj–X–obj *A **founds** B*
nsubj–reX–obj *A re**writes** B*	⇒	nsubj–X–obj *A **writes** B*
nsubj–overX–obj *A over**takes** B*	⇒	nsubj–X–obj *A **takes** B*
nsubj–agree–xcomp–X–obj *A agrees to **buy** B*	⇒	nsubj–X–obj *A **buys** B*
nsubjpass–force–xcomp–X–obj *A is forced to **leave** B*	⇒	nsubj–X–obj *A **leaves** B*
nsubjpass–elect–xcomp–X–obj *A is elected to be **governor** of B*	⇔	nsubj–X–obj *A is **governor** of B*
nsubj–go–xcomp–X–obj *A is going to **beat** B*	⇒	nsubj–X–obj *A **beats** B*
nsubj–try–xcomp–X–obj *A tries to **compete** with B*	⇒	nsubj–X–obj *A **competes** with B*
nsubj–decide–xcomp–X–obj *A decides to **move to** B*	⇒	nsubj–X–obj *A **moves to** B*
nsubjpass–expect–xcomp–X–obj *A is expected to **visit** B*	⇒	nsubj–X–obj *A **visits** B*

Table 5: Most frequent meta rules (top), character level meta rules (middle), and implicative verb meta rules (bottom). Bold: Words corresponding to X.

6 Meta Rules and Implicative Verbs

In addition to the annotated data, we also make available all ~960k SherLIiC-InfCands found by our unsupervised algorithm. SherLIiC-InfCands's distribution is similar enough to our labeled dataset to be useful for domain adaptation, representation learning and other techniques when working on LIiC. It can also be investigated on its own in a purely unsupervised fashion as we will show now.

We can find easily interpretable patterns by looking for cases where premise and hypothesis of an InfCand have common parts. By masking these parts (X), we can abstract away from concrete instances and interesting meta rules emerge (Table 5). The most common patterns represent reasonable equivalent formulations, e.g., active/passive voice or "be Y's $X \Leftrightarrow$ be X of Y" (the *in*-variant coming from a lot of location-typed rule instances). The fifth most frequent pattern could still be formulated in an even more abstract way but shows already that the general principle of a conjunction $Y \wedge X$ implying one of its components X can be learned.

If we search for meta rules whose X is part of a lemma (rather than a longer dependency path), we discover cases of derivational morphology such as agent nouns (e.g., *ruler*, *leader*) and sense preserving verb prefixes (e.g., *re-write*, *over-react*).

Finally, we observe several implicative verbs (verbs that entail their complement clauses) in their typical pattern V *to* $X \Rightarrow X$. A lot of these verbs are not traditional implicatives, but are called *de facto implicatives* by Pavlick and Callison-Burch (2016) – who argue for the importance of data-driven approaches to detecting de facto implicatives. The meta rule discovery method just described is such a data-driven approach.

7 Related Work

NLI challenge datasets. A lot of work exists that aims at uncovering weaknesses in state-of-the-art NLI models. Several approaches are based on modifications of popular datasets, such as SNLI or MultiNLI. These modifications range from simple rule-based transformations (Naik et al., 2018) to rewritings generated by genetic algorithms (Alzantot et al., 2018) or adversarial neural networks (Zhao et al., 2018). Lalor et al. (2016) constructed an NLI test set by judging the difficulty of the sentence pairs in a small SNLI subset based on crowd-sourced human responses via Item Response Theory. These works are related as they, too, challenge existing NLI models with new data but orthogonal to ours as their goal is not to measure a model's knowledge about lexical inference in context.

Glockner et al. (2018) modified SNLI by replacing one word from a given sentence by a synonym, (co-)hyponym, hypernym or antonym to build a test set that requires NLI systems to use lexical knowledge. They rely on WordNet's lexical taxonomy. This, however, is difficult for verbs because their semantics depends more on context. Finally, Glockner et al. (2018)'s dataset has a strong bias for *contradiction* whereas our dataset is specifically designed to contain cases of *entailment*.

Our work is more closely related to the dataset by Levy and Dagan (2016), who frame relation entailment as the task of judging the appropriateness of candidate answers. Their hypothesis is that an answer is only appropriate if it entails the predicate of the question. This is often but by no means always true; certain questions imply additional information. Consider: "Which country annexed country[B]?" The answer candidate "country[A] administers country[B]" might be considered valid, given that it is unlikely that one country annexes B and another country administers it. The inference *administer* $\Rightarrow$ *annex*, however, does not hold. Because of these difficulties, we follow the more traditional approach (Zeichner et al., 2012) of asking about consequences of a given fact (the premise).

Relation extraction. Some works (Schoenmackers et al., 2010; Berant, 2012; Zeichner et al., 2012) rely on the output from open information extraction systems (Banko et al., 2007; Fader et al., 2011). A more flexible approach is to represent relations as lexicalized paths in dependency graphs (Lin and Pantel, 2001; Szpektor et al., 2004), sometimes with semantic postprocessing (Shwartz et al., 2017b) and/or retransforming into textual patterns (Nakashole et al., 2012). We, too, choose the latter.

Relation typing. Typing relations has become standard in inference mining because of its usefulness for sense disambiguation (Schoenmackers et al., 2010; Nakashole et al., 2012; Yao et al., 2012; Lewis and Steedman, 2013). Still some resources only provide types for one argument slot of their binary relations (Levy and Dagan, 2016) or no types at all (Zeichner et al., 2012; Berant, 2012; Shwartz et al., 2017b). Our InfCands are typed in both argument slots, which both facilitates disambiguation and makes them more general.

Some works (Yao et al., 2012; Lewis and Steedman, 2013) learn distributions over latent type signatures for their relations via topic modeling. A large disadvantage of latent types is their lack of intuitive interpretability. By design, our KG types are meaningful and human-interpretable.

Schoenmackers et al. (2010) type common nouns based on cooccurrence with class nouns identified by Hearst patterns (Hearst, 1992) and later try to filter out unreasonable typings by using frequency thresholds and PMI. As KG entities are manually labeled with their correct types, we do not need this kind of heuristics. Furthermore, in contrast to this ad-hoc type system, KG types are the result of a KG design process. Notably, Freebase types function as interfaces, i.e., permit type-specific properties to be added, and are thus inherently motivated by relations between entities.

Lexical ontologies, such as WordNet (as used by Levy and Dagan, 2016) likewise lack this connection between relations and types. Moreover, relations between real-world entities are more often events than relations between common nouns. Thus, in contrast to existing resources that do not restrict relations to KG entities, SherLIiC contains more event-like relations.

Nakashole et al. (2012) also use KG types as context for their textual patterns. They simply create a new relation for each possible type combination for each entity occurring with a pattern and each possible type of this entity. It is unclear how the combinatorial explosion and the resulting sparsity affects pattern quality. Our approach of successively splitting a typewise heterogenous relation into its k largest homogenous subrelations aims at finding only the most typical types for an action and our definition of type signature as intersection of all common types avoids unnecessary redundancy.

Entailment candidate collection. Distributional features are a common choice for paraphrase detection and relation clustering (Lin and Pantel, 2001; Szpektor et al., 2004; Sekine, 2005; Yao et al., 2012; Lewis and Steedman, 2013).

The two most important alternatives are bilingual pivoting (Ganitkevitch et al., 2013) – which identifies identically translated phrases in bilingual corpora – and event coreference in the news (Xu et al., 2014; Zhang et al., 2015; Shwartz et al., 2017b) – which relies on lexical variability in two articles or headlines referring to the same event. We specifically focus on distributional information for our InfCand collection because current models of lexical semantics are also mainly based on that (e.g., Grave et al., 2017). Our goal is not to build a resource free of typical mistakes made by distributional approaches but to provide a benchmark to study the progress on overcoming them (cf. § 5).

Another difference to aforementioned works is that we explicitly model unidirectional entailment as opposed to bidirectional synonymy (cf. Table 4, (1)). Here one can distinguish a learning-based approach (Berant, 2012), where an SVM classifier with various features is trained on lexical ontologies like WordNet, followed by the application of global transitivity constrains to enhance consistency, and probabilistic models of noisy set inclusion in the tradition of the distributional inclusion hypothesis (Schoenmackers et al., 2010; Nakashole et al., 2012). We adapt Sherlock, an instance of the latter, for its simplicity and effectiveness.

8 Conclusion

We presented SherLIiC, a new challenging testbed for LIiC and NLI, based on typed textual relations between named entities (NEs) from a KG. The restriction to NEs (as opposed to common nouns) allowed us to harness more event-like relations than previous similar collections as these naturally occur more often with NEs. The distributional similarity of both positive and negative examples makes SherLIiC a promising benchmark to track future NLI models' ability to go beyond shallow semantics relying primarily on distributional evidence. We showed that existing rule bases are complementary to SherLIiC and that current semantic vector space models as well as SOTA neural NLI models cannot achieve at the same time high precision and high recall on SherLIiC. Although SherLIiC's creation is entirely data-driven, it shows a large variety of linguistic challenges for NLI, ranging from lexical relations like troponymy, synonymy or morph. derivation to typical actions and common sense knowledge (cf. Table 1). The large unlabeled resources, SherLIiC-InfCands and SherLIiC-TEG, are potentially useful for further linguistic analysis (as we showed in § 6), as well as for data-driven models of lexical semantics, e.g., techniques such as representation learning and domain adaptation. We hope that SherLIiC will foster better modeling of lexical inference in context as well as progress in NLI in general.

Acknowledgments

We gratefully acknowledge a Ph.D. scholarship awarded to the first author by the German Academic Scholarship Foundation (Studienstiftung des deutschen Volkes). This work was supported by the BMBF as part of the project MLWin (01IS18050).

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Procs. of IJCAI*, volume 7, pages 2670–2676.

Jonathan Berant. 2012. *Global Learning of Textual Entailment Graphs*. Ph.D. thesis, The Blavatnik School of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 610–619, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668. Association for Computational Linguistics.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing textual entailment: Models and applications*. Morgan & Claypool Publishers.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Christiane Fellbaum. 2005. Wordnet and wordnets. In Keith Brown et al., editor, *Encyclopedia of Language and Linguistics*, second edition, pages 665–670. Elsevier, Oxford.

Benoît Frénay and Michel Verleysen. 2014. Classification in the Presence of Label Noise: A Survey. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, 25(5):845–869.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. FACC1: Freebase annotation of ClueWeb corpora, Version 1 (Release date 2013-06-26, Format version 1, Correction level 0).

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

Edouard Grave, Tomas Mikolov, Armand Joulin, and Piotr Bojanowski. 2017. Bag of tricks for efficient text classification. In *EACL (2)*, pages 427–431. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*, COLING '92, pages 539–545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise.

In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10477–10486. Curran Associates, Inc.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-geffet. 2010. Directional distributional similarity for lexical inference. *Nat. Lang. Eng.*, 16(4):359–389.

John Lalor, Hao Wu, and hong yu. 2016. Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 648–657, Austin, Texas. Association for Computational Linguistics.

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79, Montréal, Canada. Association for Computational Linguistics.

Omer Levy and Ido Dagan. 2016. Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Mike Lewis and Mark Steedman. 2013. Combining distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Dekang Lin and Patrick Pantel. 2001. DIRT: Discovery of Inference Rules from Text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 323–328, New York, NY, USA. ACM Press.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Saif Mohammad, Bonnie Dorr, and Graeme Hirst. 2008. Computing word-pair antonymy. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 982–991, Honolulu, Hawaii. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. Patty: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1135–1145, Jeju Island, Korea. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Ellie Pavlick and Chris Callison-Burch. 2016. Tense manages to predict implicative behavior in verbs. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2225–2229, Austin, Texas. Association for Computational Linguistics.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China. Association for Computational Linguistics.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association* for Computational Linguistics (Volume 2: Short Papers), pages 358–363, Melbourne, Australia. Association for Computational Linguistics.

Wesley C Salmon. 1971. *Statistical explanation and statistical relevance*, volume 69. University of Pittsburgh Pre.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42, Gothenburg, Sweden. Association for Computational Linguistics.

Stefan Schoenmackers, Jesse Davis, Oren Etzioni, and Daniel Weld. 2010. Learning first-order horn clauses from web text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: 2010*, pages 1088–1098.

Satoshi Sekine. 2005. Automatic paraphrase discovery based on context and keywords between NE pairs. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017a. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 65–75, Valencia, Spain. Association for Computational Linguistics.

Vered Shwartz, Gabriel Stanovsky, and Ido Dagan. 2017b. Acquiring predicate paraphrases from news tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 155–160, Vancouver, Canada. Association for Computational Linguistics.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP 2004*, pages 41–48, Barcelona, Spain. Association for Computational Linguistics.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 2071–2080.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of Coling 2004*, pages 1015–1021, Geneva, Switzerland. COLING.

Dirk Weissenborn, Pasquale Minervini, Tim Dettmers, Isabelle Augenstein, Johannes Welbl, Tim Rocktäschel, Matko Bošnjak, Jeff Mitchell, Thomas Demeester, Pontus Stenetorp, and Sebastian Riedel. 2018. Jack the Reader – A Machine Reading Framework. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL) System Demonstrations*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

XTAG Research Group. 2001. A lexicalized tree adjoining grammar for english. Technical Report IRCS-01-03, IRCS, University of Pennsylvania.

Wei Xu, Alan Ritter, Chris Callison-Burch, William B. Dolan, and Yangfeng Ji. 2014. Extracting lexically divergent paraphrases from twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 712–720, Stroudsburg, PA, USA. Association for Computational Linguistics.

Naomi Zeichner, Jonathan Berant, and Ido Dagan. 2012. Crowdsourcing inference-rule evaluation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 156–160, Jeju Island, Korea. Association for Computational Linguistics.

Congle Zhang, Stephen Soderland, and Daniel S. Weld. 2015. Exploiting parallel news streams for unsupervised event extraction. *Transactions of the Association for Computational Linguistics*, 3:117–129.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *International Conference on Learning Representations*.

A Relation Filter Heuristics

In order to be kept as a relation, a dependency path must fulfill all of the following criteria:

1. It starts or ends with `nsubj` or `nsubjpass`.
2. It starts or ends with one of the following labels: `nsubj`, `nsubjpass`, `iobj`, `dobj`, `pobj`, `appos`, `poss`, `rcmod`, `infmod`, `partmod`.
3. It is not longer than 7 words and 8 dependency labels.

4. At least one of the presumable lemmas contains at least 3 letters.

5. It does not have the same dependency label at both ends.

6. It does not contain any of the following labels: `parataxis`, `pcomp`, `csubj`, `advcl`, `ccomp`.

7. It does not contain immediate repetitions of words or dependency labels.