Focused Entailment Graphs for Open IE Propositions

Omer Levy† **Ido Dagan**† **Jacob Goldberger**§
† Computer Science Department § Faculty of Engineering
Bar-Ilan University
Ramat-Gan, Israel
{omerlevy,dagan,goldbej}@{cs,cs,eng}.biu.ac.il

Abstract

Open IE methods extract structured propositions from text. However, these propositions are neither consolidated nor generalized, and querying them may lead to insufficient or redundant information. This work suggests an approach to organize open IE propositions using entailment graphs. The entailment relation unifies equivalent propositions and induces a specific-to-general structure. We create a large dataset of gold-standard proposition entailment graphs, and provide a novel algorithm for automatically constructing them. Our analysis shows that predicate entailment is extremely context-sensitive, and that current lexical-semantic resources do not capture many of the lexical inferences induced by proposition entailment.

1 Introduction

Open information extraction (open IE) extracts natural language propositions from text without pre-defined schemas as in supervised relation extraction (Etzioni et al., 2008). These propositions represent predicate-argument structures as tuples of natural language strings. Open IE enables knowledge search by aggregating billions of propositions from the web[1]. It may also be perceived as capturing an unsupervised knowledge representation schema, complementing supervised knowledge bases such as Freebase (Bollacker et al., 2008), as suggested by Riedel et al (2013).

However, language variability obstructs open IE from becoming a viable knowledge representation framework. As it does not consolidate natural language expressions, querying a database of open IE propositions may lead to either insufficient or redundant information. As an illustrative example, querying the demo (footnote 1) for the generally equivalent *relieves headache* or *treats headache* returns two different lists of entities; out of the top few results, the only answers these queries seem to agree on are *caffeine* and *sex*. This is a major drawback relative to supervised knowledge representations, which map natural language expressions to structured formal representations, such as *treatments* in Freebase.

In this work, we investigate an approach for organizing and consolidating open IE propositions using the novel notion of *proposition entailment graphs* (see Figure 1) – graphs in which each node represents a proposition and each directed edge reflects an entailment relation, in the spirit of textual entailment (Dagan et al., 2013). Entailment provides an effective structure for aggregating natural-language based information; it merges semantically equivalent propositions into cliques, and induces specification-generalization edges between them. For example, *(aspirin, eliminate, headache)* entails, and is more specific than, *(headache, respond to, painkiller)*.

We thus propose the task of constructing an entailment graph over a set of open IE propositions (Section 3), which is closely related to Berant et al's work (2012) who introduced *predicate* entailment graphs. In contrast, our work explores *propositions*, which are essentially predicates instantiated with arguments, and thus semantically richer. We provide a dataset of 30 such graphs, which represent *1.5 million* pairwise entailment decisions between propositions (Section 4).

To approach this task, we extend the state-of-the-art method for building entailment graphs (Berant et al., 2012) from predicates to complete propositions. Both Snow et al (2006) and Berant et al used WordNet as distant supervision when training a local pairwise model of lexical entailment. However, analyzing our data revealed that the lexical inferences captured in WordNet are quite dif-

[1] See demo: openie.cs.washington.edu

Proceedings of the Eighteenth Conference on Computational Language Learning, pages 87–97,
Baltimore, Maryland USA, June 26-27 2014.

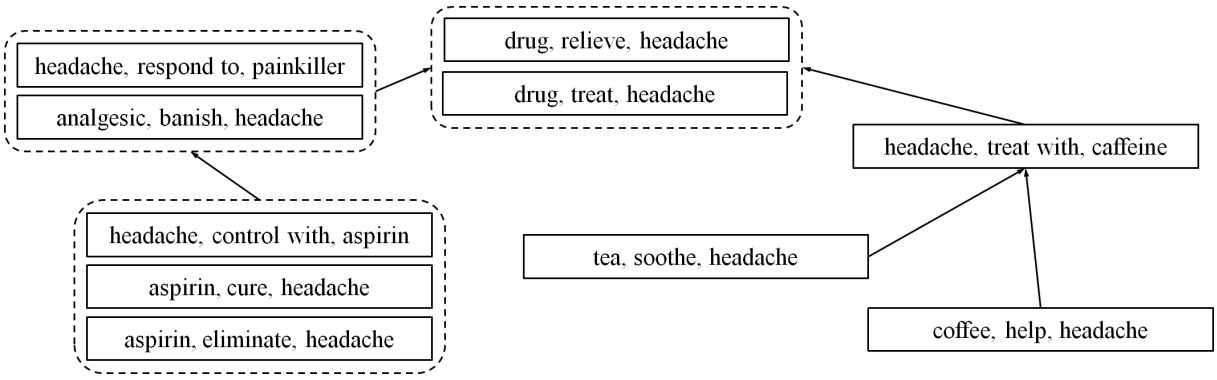

Figure 1: An excerpt from a proposition entailment graph focused on the topic *headache*. The dashed boundaries in the figure denote cliques, meaning that all propositions within them are equivalent.

ferent from the *real* lexical inferences induced by proposition entailment, making WordNet a misleading form of supervision. We therefore employ *direct* proposition-level supervision, and design a probabilistic model that captures the underlying lexical-component inferences (Section 5). We explore a variety of natural extensions to prior art as baselines (Section 6) and show that our model outperforms them (Section 7).

While our model increases performance on this task, there is still much room for improvement. A deeper analysis (Section 8) shows that common lexical-semantic resources, on which we rely as well, are either too noisy or provide inadequate recall regarding lexical entailment. In particular, we find that predicate inference within propositions often goes beyond inference between the predicates' linguistic meanings. While *pneumonia requires antibiotics* and *pneumonia is treated by antibiotics* mean the same, the inherent meanings of *require* and *treat* are different. These inferences pertain to specific world knowledge, and warrant future research.

Our work also contributes to textual entailment research. First, we extend entailment graphs to complete propositions. Secondly, we investigate an intermediate problem of recognizing entailment between language-based predicate-argument tuples. Though this problem is simpler than sentence-level entailment, it does capture entailment of complete statements, which proves to be quite challenging indeed.

2 Background

Our work builds upon two major research threads: open IE, and entailment graphs.

2.1 Open Information Extraction

Research in open IE (Etzioni et al., 2008) has focused on transforming text to predicate-argument tuples (propositions). The general approach is to learn proposition extraction patterns, and use them to create tuples while denoting extraction confidence. Various methods differ in the type of patterns they acquire. For instance, (Banko et al., 2007) and (Fader et al., 2011) used surface patterns, while (Mausam et al., 2012) and (Xu et al., 2013) used syntactic dependencies.

Yates and Etzioni (2009) tried to mitigate the issue of language variability (as exemplified in the introduction) by clustering synonymous predicates and arguments. While these clusters do contain semantically *related* items, they do not necessarily reflect *equivalence* or *implication*. For example, *coffee*, *tea*, and *caffeine* may all appear in one cluster, but *coffee* does not imply *tea*; on the other hand, separating any element from this cluster removes a valid implication. Entailment, however, can capture the fact that both beverages imply caffeine, but not one another. Also related, Riedel et al (2013) try to generalize over open IE extractions by combining knowledge from Freebase and globally predicting which unobserved propositions are true. In contrast, our work identifies inference relations between concrete *pairs* of *observed* propositions.

2.2 Entailment Graphs of Words and Phrases

Previous work focused on entailment graphs or similar structures at the sub-propositional level. In these graphs, each node represents a natural language word or phrase, and each directed edge an entailment (or generalization) relation. Snow et al (2006) created a taxonomy of sense-

disambiguated nouns and their hyponymy relations. Berant et al (2012) constructed entailment graphs of predicate templates. Recently, Mehdad et al (2013) built an entailment graph of noun phrases and partial sentences for topic labeling. The notion of *proposition* entailment graphs, however, is novel. This distinction is critical, because apparently, entailment in the context of specific propositions does not behave like context-oblivious lexical entailment (see Section 8).

Berant et al's work was implemented in Adler et al's (2012) text exploration demo, which instantiated manually-annotated predicate entailment graphs with arguments, and used an additional lexical resource to determine argument entailment. The combined graphs of predicate and argument entailments induced a proposition entailment graph, which could then be explored in a faceted-search scheme. Our work goes beyond, and attempts to build entailment graphs of propositions automatically.

2.2.1 Berant et al's Algorithm for Predicate Entailment Graph Construction

We present Berant et al's algorithm in detail, as we rely on it later on. Given a set of predicates $\{i\}_{1..n}$ as input (constituting the graph nodes), it returns a set of entailment decisions (i, j), which become the directed edges of the entailment graph. The method works in two phases: (1) *local estimation*, and (2) *global optimization*.

The **local estimation** model considers every potential edge (i, j) and estimates the probability p_{ij} that this edge indeed exists, i.e. that i entails j. Each predicate pair is represented with distributional similarity features, providing some indication of whether i entails j. The estimator then uses logistic regression (or a linear SVM) over those features to predict the probability of entailment. It is trained with distant supervision from WordNet, employing synonyms, hypernyms, and (WordNet) entailments as positive examples, and antonyms, hyponyms, and cohyponyms as negative.

The **global optimization** phase then searches for the most probable *transitive* entailment graph, given the local probability estimations. It does so with an integer linear program (ILP), where each pair of predicates is represented by a binary variable x_{ij}, denoting whether there is an entailment edge from i to j. The objective function corresponds to the log likelihood of the assignment: $\sum_{i \neq j} x_{ij} \left(\log \left(\frac{p_{ij}}{1-p_{ij}} \right) + \log \left(\frac{\pi}{1-\pi} \right) \right)$. The prior term π is the probability of a random pair of predicates to be in an entailment relation, and can be estimated in advance. The ILP solver searches for the optimal assignment that maximizes the objective function under transitivity constraints, expressed as linear constraints $\forall_{i,j,k} \;\; x_{ij} + x_{jk} - x_{ik} \leq 1$.

3 Task Definition

A *proposition entailment graph* is a directed graph where each node is a proposition s_i (s for *sentence*) and each edge (s_i, s_j) represents an entailment relation from s_i to s_j. A proposition s_i is a predicate-argument structure $s_i = \left(p_i, a_i^1, a_i^2, ..., a_i^{m_i}\right)$ with one predicate p_i and its arguments. A proposition-level entailment (s_i, s_j) holds if the verbalization of s_i implies s_j, according to the definition of textual entailment (Dagan et al., 2013); i.e. if humans reading s_i would typically infer that s_j is most likely true. Given a set of propositions (graph nodes), the task of constructing a proposition entailment graph is to recognize all the entailments among the propositions, i.e. deciding which directional edges connect which pairs of nodes.

In this paper, we consider the narrower task of constructing $focused$ proposition entailment graphs, following Berant et al's methodology in creating focused predicate entailment graphs. First, all predicates are binary (have two arguments) and are denoted $s_i = \left(a_i^1, p_i, a_i^2\right)$. Secondly, we assume that the propositions were retrieved by querying for a particular concept; out of the two arguments, one argument t (topic) is common to all the propositions in a single graph. We denote the non-topic argument as a_i. Figure 1 presents an example of an informative entailment graph focused on the topic *headache*.

Though confined, this setting still challenges the state-of-the-art in textual entailment (see Section 7). Moreover, these restrictions facilitate piece-wise investigation of the entailment problem (see Section 8).

4 Dataset

To construct our dataset of open IE extractions, we found Google's syntactic ngrams (Goldberg and Orwant, 2013) as a useful source of high-quality propositions. Based on a corpus of 3.5 million English books, it aggregates every *syntactic ngram*

– subtree of a dependency parse – with at most 4 dependency arcs. The resource contains only tree fragments that appeared at least 10 times in the corpus, filtering out many low-quality syntactic ngrams.

We extracted the syntactic ngrams that reflect propositions, i.e. *subject-verb-object* fragments where *object* modifies the verb with either *dobj* or *pobj*. Prepositions in *pobj* were concatenated to the verb (e.g. *use with*). In addition, both *subject* and *object* must each be a noun phrase containing two tokens at most, which are either nouns or adjectives. Each token in the extracted fragments was then lemmatized using WordNet. After lemmatization, we grouped all identical propositions and aggregated their counts. Approximately 68 million propositions were collected.

We chose 30 topics from the healthcare domain (such as *influenza*, *hiv*, and *penicillin*). For each topic, we collected the set of propositions containing it, and manually filtered noisy extractions. This yielded 30 high-quality sets of 5,714 propositions in total, where each set becomes the set of nodes in a separate focused entailment graph. The graphs range from 55 propositions (*scurvy*) to 562 (*headache*), with an average of over 190 propositions per graph. Summing the number of proposition pairs within each graph amounts to a total of *1.5 million potential entailment edges*, which makes it by far the largest annotated textual entailment dataset to date.

We used a semi-automatic annotation process, which dramatically narrows down the number of manual decisions, and hence, the required annotation time. In short, the annotators are given a series of small clustering tasks before annotating entailment between those clusters.[2]

The annotation process was carried out by two native English speakers, with the aid of encyclopedic knowledge for unfamiliar medical terms. The agreement on a subset of five randomly sampled graphs was $\kappa = 0.77$. Annotating a single graph took about an hour and a half on average.

Positive entailment judgements constituted only 8.4% of potential edges, and were found to be 100% transitive. We observe that in nearly all of those cases, a natural alignment between entailing components occurs: predicates align with each other, the topic is shared, and the remaining non-topic argument aligns with its counterpart. Consider the topic *arthritis* and the entailing proposition pair *(arthritis, cause, pain)→(symptom, associate with, arthritis)*; *cause→associate with*, while *pain→symptom*. Rarely, some misalignments do occur; for instance *(vaccine, protects, body)→(vaccine, provides, protection)*. However, it is almost always the case that propositions entail if and only if their aligned lexical components entail as well.

[2]The annotated dataset is publicly available on the first author's website.

5 Algorithm

In this section, we extend Berant et al's algorithm (2012) to construct entailment graphs of *propositions*. As described in Section 2.2.1, their method first performs local estimation of predicate entailment and then global optimization. We modify the local estimation phase to estimate *proposition* entailment instead, and then apply the same global optimization in the second phase.

In Section 4, we observed the alignment-based relationship between proposition and lexical entailment. We leverage this observation to predict proposition entailment with lexical entailment features (as Berant et al), using the Component Entailment Conjunction (CEC) model in Section 5.1.

Following Snow et al (2006) and Berant et al, we could train CEC using distant supervision from WordNet. In fact, we did try this approach (presented as baseline methods, Section 6) and found that it performed poorly. Furthermore, our analysis (Section 8) suggests that WordNet relations do not adequately capture the lexical inferences induced by proposition-level entailment. Instead, we use a more realistic signal to train CEC – direct supervision from the annotated dataset. Section 5.2 describes how we propagate proposition-level entailment annotations to the latent lexical components.

5.1 Component Entailment Conjunction

CEC assumes that proposition-level entailment is the result of entailment within each pair of aligned components, i.e. a pair of propositions entail if and only if both their predicate and argument pairs entail. This assumption stems from our observation of alignment in Section 4. Furthermore, CEC leverages this interdependence to learn separate predicate-entailment and argument-entailment features through proposition-level supervision.

Formally, for every ordered pair of propositions (i, j) we denote proposition entailment as a binary random variable x^s_{ij} and predicate and argument entailments as x^p_{ij} and x^a_{ij}, respectively. In our setting, proposition entailment (x^s_{ij}) is observed, but component entailments (x^p_{ij}, x^a_{ij}) are hidden. We use logistic regression, with features ϕ^p_{ij} and parameter w^p, as a probabilistic model of predicate entailment (and so for arguments with ϕ^a_{ij} and w^a):

$$p_{ij} = P\left(x^p_{ij} = 1 \middle| \phi^p_{ij}; w^p\right) = \sigma\left(\phi^p_{ij} \cdot w^p\right)$$

$$a_{ij} = P\left(x^a_{ij} = 1 \middle| \phi^a_{ij}; w^a\right) = \sigma\left(\phi^a_{ij} \cdot w^a\right) \quad (1)$$

where σ is the sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$. We then define proposition entailment as the conjunction of its binary components: $x^s_{ij} = x^p_{ij} \wedge x^a_{ij}$. Therefore, the probability of proposition entailment given the component features is:

$$s_{ij} = P\left(x^s_{ij} = 1 \middle| \phi^p_{ij}, \phi^a_{ij}; w^p, w^a\right)$$

$$= P\left(x^p_{ij} = 1, x^a_{ij} = 1 \middle| \phi^p_{ij}, \phi^a_{ij}; w^p, w^a\right)$$

$$= P\left(x^p_{ij} = 1 \middle| \phi^p_{ij}; w^p\right) \cdot P\left(x^a_{ij} = 1 \middle| \phi^a_{ij}; w^a\right)$$

$$= p_{ij} \cdot a_{ij}$$

The proposition entailment probability is thus the product of component entailment probabilities.

Given the proposition-level information $\left\{x^s_{ij}\right\}$, the log-likelihood is:

$$\ell\left(w^p, w^a\right) = \sum_{i \neq j} \log P\left(x^s_{ij} \middle| \phi^p_{ij}, \phi^a_{ij}; w^p, w^a\right) =$$

$$\sum_{i \neq j} \left(x^s_{ij} \log\left(p_{ij} a_{ij}\right) + \left(1 - x^s_{ij}\right) \log\left(1 - p_{ij} a_{ij}\right)\right)$$

5.2 Learning Component Models

We wish to learn the model's parameters (w^p, w^a). Our approach uses direct proposition-level supervision from our annotated dataset to train the component logistic regression models. Since component entailment (x^p_{ij}, x^a_{ij}) is not observed in the data, we apply the iterative EM algorithm (Dempster et al., 1977). In the E-step we estimate their probabilities from proposition-level labels (x^s_{ij}), and in the M-step we use those estimates as "soft" labels to learn the component-level model parameters (w^p, w^a).

E-Step During the E-step in iteration $t + 1$, we compute the probability of component entailments given the proposition entailment information, based on the parameters at iteration t (w^p_t, w^a_t). The predicate probabilities are given by:

$$c^p_{ij} = P\left(x^p_{ij} = 1 \middle| x^s_{ij}, \phi^p_{ij}, \phi^a_{ij}; w^p_t, w^a_t\right) \quad (2)$$

and are computed with Bayes' law:

$$c^p_{ij} = \begin{cases} 1 & \text{if } x^s_{ij} = 1 \\ \frac{p^t_{ij}\left(1 - a^t_{ij}\right)}{1 - p^t_{ij} a^t_{ij}} & \text{if } x^s_{ij} = 0 \end{cases} \quad (3)$$

where p^t_{ij} is computed as in Equations 1, with the parameters at iteration t (w^p_t). Argument entailment probabilities (c^a_{ij}) are computed analogously.

M-Step In the M-step, we compute new values for the parameters (w^p_{t+1}, w^a_{t+1}). In our case, there is no closed-form formula for updating the parameters. Instead, at each iteration, we solve a separate logistic regression for each component. While we have each component model's features (ϕ^p_{ij}, assuming predicates for notation), we do not observe the component-level entailment labels (x^p_{ij}); instead, we obtain their probabilities (c^p_{ij}) from the expectation step.

To learn the parameters (w^p_{t+1}, w^a_{t+1}) from the component entailment probabilities (c^p_{ij}), we employ a weighted variant of logistic regression, that can utilize "soft" class labels (i.e. a probability distribution over $\{0, 1\}$). To solve such a logistic regression (e.g. for w^p_{t+1}), we maximize the log-likelihood:

$$\ell\left(w^p_{t+1}\right) =$$

$$\sum_{ij} \left(c^p_{ij} \log\left(P\left(x^p_{ij} = 1 \middle| \phi^p_{ij}; w^p_{t+1}\right)\right) + \left(1 - c^p_{ij}\right) \log\left(P\left(x^p_{ij} = 0 \middle| \phi^p_{ij}; w^p_{t+1}\right)\right)\right)$$

For optimization, we calculate the derivative, and use gradient ascent to update w^p_{t+1}:

$$\Delta w^p_{t+1} = \frac{\partial \ell\left(w^p_{t+1}\right)}{\partial w^p_{t+1}} =$$

$$\sum_{ij} \left(c^p_{ij} - P\left(x^p_{ij} = 1 \middle| \phi^p_{ij}; w^p_{t+1}\right)\right) \phi^p_{ij}$$

This optimization is concave, and therefore the unique global maximum can be efficiently obtained.

5.3 Features

Similar to Berant et al, we used three types of features to describe both predicate pairs (ϕ^p_{ij}) and argument pairs (ϕ^a_{ij}): distributional similarities, lexical resources, and string distances.

We used the entire database of 68 million extracted propositions (see Section 4) to create a word-context matrix; context was defined as other words that appeared in the same proposition, and each word was represented as $(string, role)$, $role$ being the location within the proposition, either a_1, p, or a_2. The matrix was then normalized with pointwise mutual information (Church and Hanks, 1990). We used various metrics to measure different types of similarities between each component pair, including: cosine similarity, Lin's similarity (1998), inclusion (Weeds and Weir, 2003), average precision, and balanced average precision (Kotlerman et al., 2010). Weed's and Kotlerman's metrics are directional (asymmetric) and indicate the direction of a potential entailment relation. These features were used for both predicates and arguments. In addition, we used Melamud et al's (2013) method to learn a context-sensitive model of predicate entailment, which estimates predicate similarity in the context of the given arguments.

We leveraged the Unified Medical Language System (UMLS) to check argument entailment, using the *parent* and *synonym* relations. A single feature indicated whether such a connection exists. We also used WordNet relations as features, specifically: synonyms, hypernyms, entailments, hyponyms, cohyponyms, antonyms. Each WordNet relation constituted a different feature for both predicates and arguments.

Finally, we added a string equality feature and a Levenshtein distance feature (Levenshtein, 1966) for different spellings of the same word to both predicate and argument feature vectors.

6 Baseline Methods

We consider four algorithms that naturally extend the state-of-the-art to propositions, while using distant supervision (from WordNet). Since CEC uses direct supervision, we also examined another (simpler) directly-supervised algorithm. As a naive unsupervised baseline, we use *Argument Equality*, which returns "entailing" if the argument pair is identical. *Predicate Equality* is defined similarly for predicates.

Component-Level Distant Supervision The following methods use distant supervision from WordNet (as in Berant et al's work, Section 2.2.1) to explicitly train component-level entailment estimators. Specifically, we train a logistic regression model for each component as specified in Equations 1 in Section 5.1. We present four methods, which differ in the way they obtain global graph-level entailment decisions for propositions, based on the local component entailment estimates (p_{ij}, a_{ij} in Section 5.1).

The first method, $Opt(Arg \wedge Pred)$, uses the product of both component models to estimate local proposition-level entailment: $s_{ij} = p_{ij} \cdot a_{ij}$. The global set of proposition entailments is then determined using Berant et al's global optimization, according to the proposition-level scores s_{ij}. Note that this method is identical to CEC during inference, but differs in the way the local estimators are learned (with component-level supervision from WordNet).

An alternative is $Opt(Arg) \wedge Opt(Pred)$. It first obtains local probabilities (p_{ij}, a_{ij}) for each component as in $Opt(Arg \wedge Pred)$, but then employs component-level global optimization (transitivity enforcement), yielding two sets of entailment decisions, x^p_{ij} and x^a_{ij}. Proposition entailment is then determined by the conjunction $x^s_{ij} = x^p_{ij} \wedge x^a_{ij}$, as in (Adler et al., 2012).

Finally, $Opt(Arg)$ ignores the predicate component. Instead, it uses only the argument entailment graph (as produced by $Opt(Arg) \wedge Opt(Pred)$) to decide on proposition entailment; i.e. a pair of propositions entail if and only if their arguments entail. $Opt(Pred)$ is defined analogously.

Proposition-Level Direct Supervision A simpler alternative to CEC that also employs proposition-level supervision is *Joint Features*, which concatenates the component level features into a unified feature vector: $\phi^s_{ij} = \phi^p_{ij} \oplus \phi^a_{ij}$. We then couple them with the gold-standard annotations x^s_{ij} to create a training set for a single logistic regression. We use the trained logistic regression to estimate the local probability of proposition entailment, and then perform global optimization to construct the entailment graph.

7 Empirical Evaluation

We evaluate the models in Sections 5 & 6 on the 30 annotated entailment graphs presented in Section 4. During testing, each graph was evaluated separately. The results presented in this section are all micro-averages, though macro-averages were also computed and found to reflect the same trends. Models trained with distant supervision were evaluated on all graphs. For directly super-

vised methods, we used 2×6-fold cross validation (25 training graphs per fold). In this scenario, each graph induced a set of labeled examples – its edges being positive examples, and the missing potential edges being negative ones – and the union of these sets was used as the training set of that cross-validation fold.

7.1 Results

Table 1 compares the performance of CEC with that of the baseline methods.

While *Joint Features* and CEC share exactly the same features, CEC exploits the inherent conjunction between predicate and argument entailments (as observed in Section 4 and modeled in Section 5.1), and forces both components to decide on entailment *separately*. This differs from the simpler log-linear model (*Joint Features*) where, for example, a very strong predicate entailment feature might override the overall proposition-level decision, even if there was no strong indication of argument entailment. As a result, CEC dominates *Joint Features* in both precision and recall. The F_1 difference between these methods is statistically significant with McNemar's test (1947) with $p \ll 0.01$. Specifically, CEC corrected *Joint Features* 7621 times, while the opposite occurred only 4048 times.

CEC also yields relatively high precision and recall. While it has 2% less recall than $Opt(Arg)$ (the highest-recall baseline), it surpasses $Opt(Arg)$'s precision by 14%. Along with a similar comparison to *Argument Equality* (the highest precision baseline), CEC notably outperforms all baselines.

It is also evident that both directly supervised methods outperform the distantly supervised methods. Our analysis (Section 8.1) shows that WordNet lacks significant coverage, and may therefore be a problematic source of supervision.

Perhaps the most surprising result is the complete failure of WordNet-supervised methods that consider predicate information. A deeper analysis (Section 8.2) shows that predicate inference is highly context-sensitive, and deviates beyond the lexical inferences provided by WordNet.

7.2 Learning Curve

We measure the supervision needed to train the directly supervised models by their learning curves (Figure 2). Each point is the average F_1 score across 12 cross-validation folds; e.g. for 10 training graphs, we used 4×3-fold cross validation. Even 5 training graphs (a day's worth of annotation) are enough for CEC to perform on-par with the best distantly supervised method, and with 15 training graphs it outperforms every baseline, including *Joint Features* trained with 25 graphs.

Supervision	Method	Prec.	Rec.	F1
None	*Argument Equality*	**81.6**%	42.2%	55.6%
	Predicate Equality	9.3%	1.5%	2.6%
Component (WordNet)	$Opt(Arg \wedge Pred)$	73.8%	3.8%	7.2%
	$Opt(Arg) \wedge Opt(Pred)$	72.3%	3.2%	6.0%
	$Opt(Arg)$	64.6%	**55.4**%	59.7%
	$Opt(Pred)$	11.0%	6.2%	8.0%
Proposition (Annotated)	*Joint Features*	76.3%	51.7%	61.6%
	CEC	78.7%	53.5%	**63.7**%

Table 1: Performance on gold-standard (micro averaged).

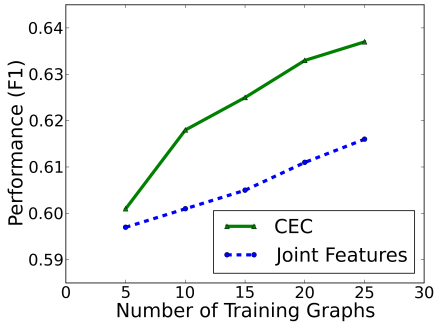

Figure 2: Learning curve of directly supervised methods.

7.3 Effects of Global Optimization

We evaluate the effects of enforcing transitivity by considering CEC with and without the global optimization phase. Table 2 shows how many entailment edges were added (and removed) by enforcing transitivity, and measures how many of those modifications were correct. Apparently, transitivity's greatest effect is the removal of incorrect entailment edges. The same phenomenon was also observed in the work on predicate entailment graphs (Berant et al., 2012). Overall, transitivity made 4,848 correct modifications out of 6,734 in total. A χ^2 test reveals that the positive contribution of enforcing transitivity is indeed statistically significant ($p \ll 0.01$).

Gold Standard	Global Opt Added Edge	Global Opt Removed Edge
Edge Exists	1150	482
No Edge	1404	3698

Table 2: The modifications made by enforcing transitivity w.r.t. the gold standard. 55% of the edges added by enforcing transitivity are incorrect, but it removed even more incorrect edges, improving the overall performance.

8 Analysis of Lexical Inference

Although CEC had a statistically-significant improvement upon the baselines, its absolute performance leaves much room for improvement. We hypothesize that the lexical entailment features we used, following state-of-the-art lexical entailment modeling, do *not* capture many of the actual lexical inferences induced by proposition entailment. We demonstrate that this is indeed the case.

8.1 Argument Entailment

To isolate the effect of different features on predicting argument entailment, we collected all proposition pairs that shared exactly the same predicate and topic, and thus differed in only their "free" argument. This yielded 20,336 aligned argument pairs, whose entailment annotations are equal to the corresponding proposition-entailment annotation in the dataset.

Using WordNet synonyms and hypernyms to predict entailment yielded a precision of about 88%, at 40% recall. Though relatively precise, WordNet's coverage is limited, and misses many inferences. We describe three typical types of inferences that were absent from WordNet.

The first type constitutes of widely-used paraphrases such as *people↔persons*, *woman↔female*, and *pain↔ache*. These may be seen as weaker types of synonyms, which may have nuances, but are typically interchangeable.

Another type is metonymy, in which a concept is not referred to by its own name, but by that of an associated concept. This is very common in our healthcare dataset, where a disease is often referred to by its underlying pathogen and vice-versa (e.g. *pneumonia↔pneumococcus*).

The third type of missing inferences is causality. Many instances of metonymy (such as the disease-pathogen example) may be seen as causality as well. Other examples can be drug and effect (*laxative→diarrhea*) or condition and symptom (*influenza→fever*).

WordNet's lack of such common-sense inferences, which are abundant in our proposition entailment dataset, might make WordNet a problematic source of distant supervision. The fact that 60% of the entailing examples in our dataset are labeled by WordNet as non-entailing, means that for each truly positive training example, there is a higher chance that it will have a negative label.

Distributional similarity is commonly used to capture such missing inferences and complement WordNet-like resources. On this dataset, however, it failed to do so. One of the more indicative similarity measures, inclusion (Weeds and Weir, 2003), yielded only 27% precision at 40% recall when tuning a threshold to optimize F_1. Increasing precision caused a dramatic drop in recall: 50% precision limited recall to 3.2%. Other similarity measures performed similarly or worse. It seems that current methods of distributional word similarity also capture relations quite different from inference, such as cohyponyms and domain relatedness, and might be less suitable for modeling lexical entailment on their own.

8.2 Context-Sensitive Predicate Entailment

The proposition-level entailment annotation induces an entailment relation between the predicates, which holds in the *particular context* of the proposition pair. We wish to understand the nature of this predicate-level entailment, and how it compares to classic lexical inference as portrayed in the lexical semantics literature. To that end, we collected all the entailing proposition pairs with equal arguments, and extracted the corresponding predicate pairs (which, assuming alignment, are necessarily entailing in that context). This list contains 52,560 predicate pairs.

In our first analysis, we explored which WordNet relations correlate with predicate entailment, by checking how well each relation covers the set of entailed predicate pairs. Synonyms and hypernyms, which are considered positive entailment indicators, covered only about 8% each. Surprisingly, the hyponym and cohyponym relations (which are considered negative entailment indicators) covered over 9% and 14%, respectively. Table 3 shows the exact details.

It seems that WordNet relations are hardly correlated with the context-sensitive predicate-level entailments in our dataset, and that the classic interpretation of WordNet relations with respect to entailment does not hold in practice, where en-

Interpretation	WordNet Relation	Coverage
Positive	Synonyms	7.85%
	Direct Hypernyms	5.62%
	Indirect Hypernyms	3.14%
	Entailment	0.33%
Negative	Antonyms	0.31%
	Direct Hyponyms	5.74%
	Indirect Hyponyms	3.51%
	Cohyponyms	14.30%

Table 3: The portion of positive predicate entailments covered by each WordNet relation. WordNet relations are divided according to their common interpretations with respect to lexical entailment.

tailments are judged in the context of concrete propositions. In fact, negative indicators in WordNet seem to cover more predicate entailments than positive ones. This explains the failure of WordNet-supervised methods with predicate entailment features (Section 7.1).

Since we do not expect WordNet to cover all shades of entailment, we conducted a manual analysis as well. 100 entailing predicate pairs were randomly sampled, and manually annotated for lexical-level entailment, without seeing their arguments. To compensate for the lack of context, we guided the annotators to assume a general healthcare scenario, and use a more lenient interpretation of textual entailment (biased towards positive entailment decisions). Nevertheless, only 56% of the predicate pairs were labeled as entailing, indicating that the context-sensitive predicate inferences captured in our dataset can be quite different from generic predicate inferences.

We suggest that this phenomenon goes one step beyond what the current literature considers as context-sensitive entailment, and that it is more specific than determining an appropriate lexical sense. To demonstrate, we present four such predicate-entailment phenomena.

First, there are cases in which an appropriate lexical sense could exist in principle, but it is too specific to be practically covered by a manual resource. For example, *cures cancer→kills cancer*, but the appropriate sense for *kill* (cause to cease existing) does not exist, and in turn, neither does the hypernymy relation from *cure* to *kill*. It is hard to expect these kinds of obscure senses or relationships to comprehensively appear in a manually-constructed resource.

In many cases, such a specific sense *does not* exist. For example, *(pneumonia, require, antibiotic)→(pneumonia, treated by, antibiotics)*, but *require* does not have a general sense which means *treat by*. The inference in this example does not stem from the linguistic meaning of each predicate, but rather from the real-world situation their encapsulating propositions describe.

Another aspect of predicate entailment that may change when considering propositional context is the direction of inference. For instance, *cause↛trigger*. While it may be the case that *trigger* entails *cause*, the converse is not necessarily true since *cause* is far more general. However, when considering *(caffeine, cause, headache)* and *(caffeine, trigger, headache)*, both propositions describe the same real-world situation, and thus both propositions are mutually entailing. In this context, *cause* does indeed entail *trigger* as well.

Finally, figures of speech (such as metaphors) are abundant and diverse. Though it may not be so common to read about a drug that "banishes" headaches, most readers would understand the underlying meaning. These phenomena exceed the current scope of lexical-semantic resources such as WordNet, and require world knowledge.

9 Conclusion

This paper proposes a novel approach, based on entailment graphs, for consolidating information extracted from large corpora. We define the problem of building proposition entailment graphs, and provide a large annotated dataset. We also present the CEC model, which models the connection between proposition entailment and lexical entailment. Although it outperforms the state-of-the-art, its performance is not ideal because it relies on inadequate lexical-semantic resources that do not capture the common-sense and context-sensitive inferences which are inherent in proposition entailment. In future work, we intend to further investigate lexical entailment as induced by proposition entailment, and hope to develop richer methods of lexical inference that address the phenomena exhibited in this setting.

Acknowledgements

This work has been supported by the Israeli Ministry of Science and Technology grant 3-8705, the Israel Science Foundation grant 880/12, and the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287923 (EXCITEMENT). We would like to thank our reviewers for their insightful comments.

References

Meni Adler, Jonathan Berant, and Ido Dagan. 2012. Entailment-based text exploration with application to the health-care domain. In *Proceedings of the System Demonstrations of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 79–84.

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

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2012. Learning entailment relations by global graph structure optimization. *Computational Linguistics*, 38(1):73–111.

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*, pages 1247–1250. ACM.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the Web. *Communications of the ACM*, 51(12):68–74.

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., July. Association for Computational Linguistics.

Yoav Goldberg and Jon Orwant. 2013. A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 241–247, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, page 707.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 768–774, Montreal, Quebec, Canada, August. Association for Computational Linguistics.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea, July. Association for Computational Linguistics.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Shafiq Joty. 2013. Towards topic labeling with phrase entailment and aggregation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 179–189, Atlanta, Georgia, June. Association for Computational Linguistics.

Oren Melamud, Jonathan Berant, Ido Dagan, Jacob Goldberger, and Idan Szpektor. 2013. A two level model for context sensitive inference rules. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1331–1340, Sofia, Bulgaria, August. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia, June. Association for Computational Linguistics.

Rion Snow, Dan Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL-COLING 2006)*, pages 801–808.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 81–88.

Ying Xu, Mi-Young Kim, Kevin Quinn, Randy Goebel, and Denilson Barbosa. 2013. Open information extraction with tree kernels. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 868–877, Atlanta, Georgia, June. Association for Computational Linguistics.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34(1):255.