ENTAILMENT GRAPH LEARNING WITH TEXTUAL ENTAILMENT AND SOFT TRANSITIVITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Typed entailment graphs try to learn the entailment relations between predicates from text and model them as edges between predicate nodes. The construction of entailment graphs usually suffers from severe sparsity and unreliability of distributional similarity. We propose a two-stage method, Entailment Graph with Textual Entailment and Transitivity (EGT2). EGT2 learns the local entailment relations by recognizing the textual entailment between template sentences formed by typed CCG-parsed predicates. Based on the generated local graph, EGT2 then uses three novel soft transitivity constraints to consider the logical transitivity in entailment structures. Experiments on benchmark datasets show that EGT2 can well model the transitivity in entailment graph to alleviate the sparsity, and leads to significant improvement over current state-of-the-art methods.

1 INTRODUCTION

Entailment, as an important relation in natural language processing (NLP), is critical to correct semantic understanding and natural language inference (NLI). Entailment relation has been widely applied in different NLP tasks such as Question Answering, Machine Translation and Knowledge Graph Completion. While coming across a question that *"Which medicine cures the infection?"*, one can recognize the information *"Griseofulvin is preferred for the infection,"* in the corpus and appropriately write down the answer with the knowledge that *"is preferred for"* entails *"cures"* when their arguments are medicines and diseases, although the surface form of predicate *"cures"* does not exactly appear in the corpus. There are many ways to present one question, and it is impossible to handle them without understanding the entailment relations behind the predicates. Previous works about entailment focus on Recognizing Textual Entailment (RTE), and recently reach relatively good performance in detecting entailment relations with the transformer-based language models (He et al., 2020; Raffel et al., 2020; Schmitt & Schütze, 2021).

By modeling typed predicates as nodes and entailment relations as directed edges, the **Entailment Graph** (**EG**) is a powerful and well-established form to contain the context-independent entailment relations between predicates and the global features of entailment inference, such as paraphrasing and transitivity. As EGs are able to help reasoning without additional contexts or resource, they can be seen as a special type of structural knowledge in natural language. Figure 1 shows a simple example of entailment graph about two types of arguments, *Medicine* and *Disease*. Generally speaking, the entailment graphs are built based on a three-step process: **extracting** predicate pairs from corpus, building **local** graphs with locally computed entailment scores, and modifying graphs with **global** methods.

However, existing methods of entailment graphs face different problems in both local and global stages. The Distributional Inclusion Hypothesis (DIH) about entailment assumes that given a predicate (relation) p, it can be replaced in any context by another predicate (relation) q if and only if p entails q (Geffet & Dagan, 2005). Most of local methods in previous works are guided by DIH and thus use the distributional co-occurrence in corpus, including named entities, entity pairs and contexts, as the features to compute the entailment scores as local models. By processing different entailment relations of predicate pairs independently, the locally built graphs suffer from severe **data sparsity**. The data sparsity means that many correct entailment relations between predicates are not indicated as edges in the graphs while the two predicates do not co-occur in the corpus. Furthermore,

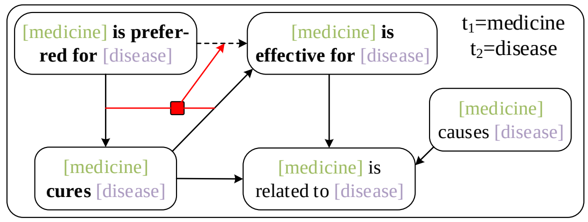

Figure 1: A simple example of entailment graph with types *medicine* and *disease*. The dashed line represents a missing entailment recovered by considering the transitivity constraint (red) based on the two premise entailment between three **boldfaced** predicates.

local models often have flaws for logical irrationality, which signifies the disobedience of predicates under some logical rules, especially transitivity.

To overcome the problem faced by local models, different global approaches are used to take the interactions and dependencies between entailment relations into consideration. The global dependency firstly implemented is the logical *transitivity*, which implies that predicate a entails predicate c if there is another predicate b making both "a entails b" and "b entails c" hold simultaneously. Berant et al. (2011) uses the Integer Linear Programming (ILP) to ensure the transitivity constraints on the entailment graphs, which is not scalable on large graphs with thousands of nodes. Hosseini et al. (2018) models the structural similarity across graphs and paraphrasing relations within graphs to learn the global consistence, but does not achieve high performance due to the lack of **high-quality local graphs** and the **transitivity** modeling.

In order to deal with the problems in local and global stage, we propose a novel entailment graph learning approach, **E**ntailment **G**raph with **T**extual Entailment and **T**ransitivity (**EGT2**). EGT2 builds high-quality local entailment graphs by inputting predicates as sentences into a transformer-based language model fine-tuned on RTE task to avoid the unreliability of distributional scores, and models the global transitivity on them by designed soft constraints losses, which alleviates the data sparsity and is available on large-scale local graphs. Our key insight is that the entailment relation $a \to c$ correctly implied by transitivity is based on two conditions: (1) the appropriate constraint scalable on large graphs containing rich information, and (2) the reliability of local graphs offering the premise $a \to b$ and $b \to c$, which is impractical in distributional approaches, but maybe available by the models well-behaved on RTE tasks. The inputting sentences are formed without contexts, which make our method accessible to those predicates not appearing in the corpus. The transitivity implication is confined to entailment relations with high confidence, which improves the quality of implied edges and cuts down the computational overheads. In a word, this paper makes the following contributions:

- It presents a new approach based on textual entailment to scoring the predicate pairs on local entailment graphs, which is reliable without distributional features and valid for arbitrary predicate pairs.
- It presents three meticulously designed global soft constraint loss functions to model the transitivity between entailment relations and alleviate the data sparsity of local approaches, which are available on large-scale entailment graphs.
- The results of extensive experiments on standard benchmarks show that our model, EGT2, significantly outperforms previous approaches of learning entailment graphs.

2 Related Work

Based on DIH, previous works extract feature vectors for typed predicates to compute the local distributional similarities. The set of entity argument pair strings, like *"Griseofulvin-infection"* in the example of Section 1, are used as the features weighted by Pointwise Mutual Information (Berant et al., 2015; Hosseini et al., 2018). Given two feature vectors of predicates, different local similarity scores, like cosine similarity, Lin (Lin, 1998), DIRT (Lin & Pantel, 2001), Weeds (Weeds & Weir, 2003) and Balanced Inclusion (Szpektor & Dagan, 2008), are calculated as the local similarities. Hosseini et al. (2019) and Hosseini et al. (2021) use Markov Chain on a entity-predicate bi-partial graph weighted by link prediction scores to calculate the transition probability between two predicates as the local score. They rely on the link predication model to calculate the features in fact. Guillou et al. (2020) adds temporal information by extracting the entity pairs within a limited temporal window as predicate features. McKenna et al. (2021) extends the graphs to include entailment relations between predicate with different numbers of arguments by splitting the features from argument pairs into independent entity slots, which impairs the representation ability of features.

As mentioned in Section 1, entailment graphs are generally learned by imposing global constraints on the local entailment relations about extracted predicates. The transitivity in entailment graph is modeled by the Integer Linear Programming (ILP) in Berant et al. (2011), which selects a transitive sub-graph of local weighted graph to maximize the summation over the weights of its edges. Their work is limited to a few hundreds of predicates due to the computational complexity of ILP. For better scalability, Berant et al. (2012) and Berant et al. (2015) propose a strong FRG-assumption that "if predicate a entails predicates b and c, b and c entail each other", and an approximation method, called Tree-Node-Fix (TNF). Obviously, the assumption is too strong to be satisfied by real cases.

Because the hard constraints show bad scalability on large-scale entailment graphs, Hosseini et al. (2018) proposes two global soft constraints that maintain the similarity between paraphrasing predicates within typed graphs and between predicates with the same names in graphs with different argument types. Their soft constraints are also used in Hosseini et al. (2019) and Hosseini et al. (2021). The first similarity implicitly takes the transitivity between paraphrasing predicates and third predicate into consideration, but ignores the transitivity in more common cases, and leads to a limited improvement on performance.

Meanwhile, the transformer-based Language Model (LM), although proved to be effective in RTE tasks (He et al., 2020; Raffel et al., 2020; Schmitt & Schütze, 2021), is not widely used in entailment graph learning. Hosseini et al. (2021) uses pretrained BERT to initialize the contextualized embeddings in their contextualized link prediction and entailment score calculation. High scores are assigned to the entailed predicates in the context of their premises, which is one implicit expression form of DIH and quite different from our direct utilization of LM on textual entailment.

3 OUR METHOD: EGT2

3.1 DEFINITION AND NOTATIONS

The target of entailment graph learning is to extract predicates, learn the entailment relations and build entailment graphs from raw text corpus. Following previous works (Hosseini et al., 2018; 2019), we use the binary relations from neo-Davisonian semantics as predicates, which is a type of first-order logic with event identifiers. For instance, the sentence *"Griseofulvin is preferred for the infection."* contains the predicate p =*(prefer.2,prefer.for.2,medicine,disease)*, and the sentence *"Griseofulvin cures the infection."* contains q =*(cure.1,cure.2,medicine,disease)*. The numbers after the predicate words are corresponding argument positions of entity *"Griseofulvin"* and *"infection"*, and the later two items are the types of arguments. Formally, a predicate with argument types t_1 and t_2 is represented as $p = (w_{p,1}.i_{p,1}, w_{p,2}.i_{p,2}, t_1, t_2)$. The predicate form is strong enough to describe most of the relations in real cases.

With T as the set of types and P as the set of all typed predicates, $V(t_1, t_2)$ contains typed predicates p with unordered argument type t_1 and t_2, where $p \in P$ and $t_1, t_2 \in T$. For predicate $p = (w_{p,1}.i_{p,1}, w_{p,2}.i_{p,2}, t_1, t_2)$, we denote that $\tau_1(p) = t_1$, $\tau_2(p) = t_2$ and $\pi(p) = (w_{p,1}.i_{p,1}, w_{p,2}.i_{p,2})$. In other words, $V(t_1, t_2) = \{p|(\tau_1(p) = t_1 \wedge \tau_2(p) = t_2) \vee (\tau_1(p) = t_2 \wedge \tau_2(p) = t_1)\}$.

A typed entailment graph $G(t_1, t_2) =< V(t_1, t_2), E(t_1, t_2) >$ is composed of the nodes of typed predicates $V(t_1, t_2)$ and the weighted edges $E(t_1, t_2)$. The edges can be also represented as sparse score matrix $W(t_1, t_2) \in [0, 1]^{|V(t_1,t_2)|\times|V(t_1,t_2)|}$, containing the entailment scores between predicates with type t_1 and t_2. As the different argument types can naturally determine whether two predicates have the same order of arguments, the order of argument type is not important while $t_1 \neq t_2$, and therefore we can ensure that $G(t_1, t_2) = G(t_2, t_1)$. For those predicates p with

Table 1: Examples of sentence generator S.

Predicates	**Sentences**
(be.1,be.capital.of.2,location$_1$,location$_2$)	Location A is capital of Location B.
(contain.1,contain.2,location$_2$,location$_1$)	Location B contains Location A.
(prefer.2,prefer.for.2,medicine,disease)	Medicine A is preferred for Disease B.
(give.2,give.3,person,thing)	Person A is given Thing B.
(aggrieved.by.2,aggrieved.felt.1,thing,person)	Person B feels aggrieved by Thing A.

$\tau_1(p) = \tau_2(p)$, the two argument types are labeled with orders, which allows the graph to contain the entailment relations with different argument orders, like *(be.1,be.capital.of.2,location$_1$,location$_2$)* $\rightarrow$ *(contain.1,contain.2,location$_2$,location$_1$)*.

3.2 Local Entailment based on Textual Entailment

Inspired by the outstanding performance of pretrained and fine-tuned LMs on RTE task, which is closely related to the entailment graphs, EGT2 uses fine-tuned transformer-based LM to calculate the local entailment scores of typed predicated pairs.

In order to utilize the knowledge about entailment relations in pretrained and fine-tuned LM, EGT2 firstly transfers the predicate pair (p, q) into corresponding sentence pair $(S(p), S(q))$ by sentence generator S, as the complicated predicates cannot be directly inputted into the LM. For typed predicate $p = (w_{p,1}.i_{p,1}, w_{p,2}.i_{p,2}, t_1, t_2)$, the generator deduces the positions of arguments about the predicate based on $i_{p,1}$ and $i_{p,2}$, generates the surface form of p based on $w_{p,1}$ and $w_{p,2}$, and finally concatenates the surface form with capitalized types as its arguments. Some generated examples are shown in Table 1, and the detailed algorithm of S is described in Appendix **??**.

After generating sentence pair $(S(p), S(q))$ for predicate pair (p, q), EGT2 inputs $(S(p), S(q))$ into a transformer-based LM to calculate the probability of the entailment relation $p \rightarrow q$ as the local entailment score in $G(t_1, t_2)$. In our experiments, the LM is implemented as DeBERTa (He et al., 2020). Generally, an entailment-oriented LM will output three scores for a sentence pair, representing the probability of relationship *entail*, *contradict* and *neutral* respectively. Formally, we denote the weighted matrix of local entailment graph with type t_1 and t_2 as W^{local}, and the weight of the edge between p and q in W^{local} is calculated as:

$$\begin{aligned} W^{local}_{p,q} &= P(p \rightarrow q) \in [0, 1], \\ P(p \rightarrow q) &= \frac{e^{LM(\textit{entail}|p,q)}}{\sum_{r \in \{\textit{entail,contradict,neutral}\}} e^{LM(r|p,q)}}, \end{aligned} \tag{1}$$

where $LM(r|p, q)$ is the output score of corresponding relationship by the LM. As the local entailment is based on the LM fine-tuned to perform textual entailment, the local graph can be built for any predicates in the parsed semantic form, or in any other forms by changing sentence generator S.

3.3 Global Entailment with Soft Transitivity Constraint

Existing approaches use global learning to find correct entailment relations which are missing or despised in local entailment graphs to overcome the data sparsity. Following Hosseini et al. (2018), the evidence from existing local edges with high confidence is used by EGT2 to predict missing edges in the entailment graphs.

The *transitivity* in entailment relation inference implies $a \rightarrow c$ while both $a \rightarrow b$ and $b \rightarrow c$ hold. For instance, in the example of Figure 1, the entailment *"is preferred for"* $\rightarrow$ *"is effective for"* is discovered because *"is preferred for"* $\rightarrow$ *"cures"* and *"cures"* $\rightarrow$ *"is effective for"* have been learned. The key challenge to incorporate the transitivity constraint into weighted graphs is discreteness of logical rules. Discreteness makes the rules impossible to be directly used in gradient-based learning methods without NP-hard complexity, as different predicate pairs are jointly involved in the calculation. To unify the discrete logical rules with gradient-based learning, inspired by Li et al. (2019), EGT2 uses the logical constraints in the form of differentiable triangular norms (Gupta

$$
\begin{aligned}
L_1 &= -log \prod_{\substack{a,b,c \in V(t_1,t_2),\\ W_{a,b},W_{b,c}>1-\epsilon}} \min(1, \frac{W_{a,c}}{W_{a,b}W_{b,c}}) \\
&= \sum_{a,b,c \in V(t_1,t_2)} I_{1-\epsilon}(W_{a,b}) I_{1-\epsilon}(W_{b,c}) ReLU(logW_{a,b} + logW_{b,c} - logW_{a,c}) \\
L_2 &= \sum_{a,b,c \in V(t_1,t_2)} -I_{1-\epsilon}(W_{a,b}) I_{1-\epsilon}(W_{b,c}) I_0(W_{a,b}W_{b,c} - W_{a,c}) logW_{a,c} \\
L_3 &= \sum_{a,b,c \in V(t_1,t_2)} -I_{1-\epsilon}(W_{a,b}) I_{1-\epsilon}(W_{b,c}) I_0(W_{a,b}W_{b,c} - W_{a,c}) W_{a,b}W_{b,c} logW_{a,c}
\end{aligned}
\tag{2}
$$

& Qi, 1991; Klement et al., 2013), or called t-norms, as the **soft constraints** so that the gradient-based learning methods can be applied.

Different t-norm methods transfer the discrete rules into different continuous loss functions. Traditional product t-norm maps $P(A \wedge B)$ into $P(A)P(B)$, $P(A \vee B)$ into $P(A)+P(B)-P(A)P(B)$, and $P(A \rightarrow B)$ into $\min(1, \frac{P(B)}{P(A)})$. For the entailment relations, the probability of transitivity to be satisfied is:

$$
\begin{aligned}
&P[(a \rightarrow b \wedge b \rightarrow c) \rightarrow (a \rightarrow c)] \\
=& \min(1, \frac{W_{a,c}}{W_{a,b}W_{b,c}}),
\end{aligned}
\tag{3}
$$

where the probability of the entailment relation $a \rightarrow b$ is represented by the local entailment scores $W_{a,b}$. To alleviate the noise from those edges assigned low confidence by local LM, EGT2 only takes the local edges whose scores are higher than $1-\epsilon$ into account (as $a \rightarrow b$ and $b \rightarrow c$), where ϵ is a small hyper-parameter because the local probability scores tend to be close to 0 or 1 in practice. Therefore, to maximize the probability of transitivity constraint satisfied over all predicates in the entailment graph $G(t_1, t_2)$, EGT2 tries to minimize the following minus-log-likelihood loss function L_1 in Eq. 2, where $I_y(x) = 1$ if $x > y$, or 0 otherwise.

Another important t-norm, called the Gödel t-norm, maps $P(A \rightarrow B)$ into 1 if $P(B) \geq P(A)$ or $P(B)$ otherwise. Therefore, the Gödel probability of transitivity to be satisfied is:

$$
\begin{aligned}
&P[(a \rightarrow b \wedge b \rightarrow c) \rightarrow (a \rightarrow c)] \\
=& \begin{cases} W_{a,c} & W_{a,b}W_{b,c} > W_{a,c} \\ 1 & otherwise \end{cases},
\end{aligned}
\tag{4}
$$

and EGT2 similarly tries to minimize the loss function L_2 in Eq. 2. It should be noted that transitivity constraints will be disobeyed not only by the missing edges, but also by the spurious edges in the local graphs. Therefore, we expect the soft constraints to take reducing the weights of premise edges into consideration. L_1 do this by the loss item $W_{a,b}$ and $W_{b,c}$, and we modify L_2 to L_3 in Eq. 2 so that the low confidence of $W_{a,c}$ will help to detect whether $W_{a,b}$ and $W_{b,c}$ are spurious.

Given the local entailment graph $G(t_1, t_2)$ with weighted edges W^{local}, in order to ensure that the global entailment graph W is not too far from W^{local}, EGT2 finally minimizes the following loss function L to trade off the distance from local graphs and the soft transitivity constraint:

$$
L = \sum_{a,b \in V} (W_{a,b} - W^{local}_{a,b})^2 + \lambda L_i, \; i = 1, 2, 3 \tag{5}
$$

where L_i is the specified implementation of soft transitivity constraint in Eq. 2, and λ is a non-negative hyper-parameter that controls the influence of two loss terms.

4 EXPERIMENTAL SETUP

4.1 PREDICATE EXTRACTION

Following Hosseini et al. (2018) and Hosseini et al. (2019), we use the multiple-source NewsSpike corpus (Zhang & Weld, 2013), which contains 550K news articles, to extract binary relations as generated predicates in EGT2. We make use of the triples released and filtered in Hosseini et al. (2019), which applies GraphParser (Reddy et al., 2014) based on Combinatorial Categorial Grammar (CCG) syntactic derivations to extracting binary relations between predicates and arguments. The argument entities are linked to Freebase (Bollacker et al., 2008) and mapped to the first level of the FIGER types (Ling & Weld, 2012) hierarchy. The type of a predicate is determined by its two corresponding argument entities. The triples are filtered by two rules to remove the noisy binary relations and arguments: (1) we only keep those argument-pairs appearing in at least 3 relations; (2) we only keep those relations with at least 3 different argument-pairs. The number of relations in the corpus is reduced from 26M to 3.9M, covering 304K typed predicates in 355 typed entailment graphs.

4.2 EVALUATION DATASETS AND METRICS

We use Levy/Holt Dataset (Levy & Dagan, 2016; Holt, 2018) and Berant Dataset (Berant et al., 2011) to evaluate the performance of entailment graph models.

In Levy's dataset, each example contains a pair of triple with the same entities but different predicates. Some questions with one predicate were shown to the annotating workers, like *"Which medicine cures the infection?"*. The label for each example are either *True* or *False*, indicating whether the first typed predicate entails the second one, by asking the workers whether the first predicates can answer the question with the second one. For example, if *"Griseofulvin is preferred for the infection"* is a correct answer of the above question, the dataset labels *"is preferred for"* $\rightarrow$ *"cures"*. Holt (2018) re-annotates Levy's dataset and forms the renewed dataset with 18,407 examples (3,916 positive and 14,491 negative), referred as Levy/Holt Dataset. The dataset is split into validation set (30%) and test set (70%) as Hosseini et al. (2018) in our experiments.

Berant et al. (2011) annotates all the entailment relations in their corpus, which generates 3,427 positive and 35,585 negative examples, referred as Berant Dataset. Their entity types do not exactly match with the first level of FIGER types hierarchy, and therefore a simple hand-mapping by Hosseini et al. (2018) is used to unify the predicate types.

To be comparable with previous works, we evaluate our methods on the test set of Levy/Holt Dataset and the whole Berant Dataset by calculating the area under the curves (AUC) with changing the classification threshold of global entailment scores. Hosseini et al. (2018) argues that the AUC of Precision-Recall Curve (PRC) for precisions in the range $[0.5, 1]$, as predictions with higher precision than *random* are more important for the downstream applications. Therefore, we report both the AUC of PRC for precisions in the range $[0.5, 1]$ and the traditional AUC of ROC, which is more widely used in evaluation of other tasks.

4.3 COMPARISON METHODS

We compare our model with existing entailment graph construction methods (Berant et al., 2011; Hosseini et al., 2018; 2019; 2021) and the best local distributional method, Balanced Inclusion (Szpektor & Dagan, 2008), referred as BInc. We also include ablation variants of our EGT2, including local models with or without fine-tuning.

4.4 IMPLEMENTATION DETAILS

For local transformer-based LM, EGT2 uses DeBERTa (He et al., 2020) implemented by the Hugging Face transformers library (Wolf et al., 2019)[1], which has been fine-tuned on MNLI (Williams et al., 2018) dataset. In order to adapt it to the special type-oriented sentence pattern generated by S,

[1]https://github.com/huggingface/transformers

Table 2: Model performance on Levy/Holt Dataset and Berant Dataset. The best performances on every metric are **boldfaced**. Results with $*$ are from original papers, as they did not share the codes or implementation details to reproduce the results.

Methods	Levy/Holt		Berant	
Metrics	PRC	ROC	PRC	ROC
BInc	.155	.632	.147	.677
Local-Sup	.161	.632	.129	.651
Hosseini18	.163	.637	.174	.682
Hosseini19*	.187	-	-	-
- Local	.167	.639	.118	.378
Hosseini21*	.195	-	-	-
EGT2-Local	.313	.712	.360	.857
- w/o Fine-tuning	.234	.673	.147	.732
EGT2-L_1	.345	**.761**	.437	**.880**
EGT2-L_2	.319	.755	.361	.879
EGT2-L_3	**.356**	.755	**.443**	.871

we expand the validation set by extracting all of the predicates, generating sentence pairs by generator S for every two predicates, and checking whether they are labeled as paraphrase or entailment in the Paraphrase Database collection (PPDB) (Pavlick et al., 2015). We split 80% of the generated corpus to fine-tune the DeBERTa with Cross-Entropy Loss, and the rest as the validation set of fine-tuning process. The fine-tuning learning rate $\alpha_f = 10^{-5}$, and the process is terminated while the F_1 score of *entail* on validation set does not increase in 10 epochs or training after 100 epochs.

For global soft transitivity constrains, we use SGD (Cun et al., 1998) to optimize the scores W in entailment graphs with loss function L in Eq. 5 for $e = 5$ epochs. The SGD learning rate $\alpha = 0.05$, the coefficient $\lambda = 1$, and the confidence threshold $\epsilon = 0.02$. The hyper-parameters are selected based on Levy/Holt validation dataset. More implementation details are given in Appendix **??**.

For testing, if one or both predicates of the example do not appear in the corresponding typed entailment graph, we handle the example as untyped one by resorting to its average score among all typed entailment graphs. This setting is used for all methods in the experiments for fair comparison.

5 EXPERIMENT RESULTS AND DISCUSSION

5.1 MAIN RESULTS

We summarize the model performances on both Levy/Holt and Berant datasets in Table 2. All global methods, including Hosseini et al. (2018), Hosseini et al. (2019) and EGT2, perform better than their corresponding local methods, which demonstrates the effect of global constraints in alleviating the data sparsity. Although using the same extracted entailment relations with Hosseini et al. (2019), our EGT2-Local significantly outperforms previous local methods because of the high-quality entailment scores generated by reliable fine-tuned textual entailment LM. On the whole, EGT2 with transitivity constraint L_3 outperforms all the other models on both Levy/Holt Dataset and Berant Dataset with AUC of PRC, while EGT2-L_1 performs best with AUC of ROC. All of three soft transitivity constraints boost the performance of local model on all evaluation metrics, which shows that making use of transitivity rule between entailment relations improves the local entailment graph. EGT2-L_1 or EGT2-L_3 performs better than EGT2-L_2, which indicates that involving the premises $a \to b$ and $b \to c$ into loss function is also important for using transitivity constraints.

The Precision-Recall Curves of different methods and the Precision-Recall Point of Berant et al. (2011) on the two evaluation datasets are shown in Figure 2(a) and 2(b) respectively. The local and global models of EGT2 consistently outperform previous state-of-the-art methods on all levels of precision and recall, which indicates the effect of our local model based on textual entailment and global soft constraints based on transitivity. The EGT2-Local achieves slightly higher precision than global models in the range $recall < 0.5$, but its precision drops quickly if we requires higher recall and therefore leads to worse performance than global models. The result indicates that global

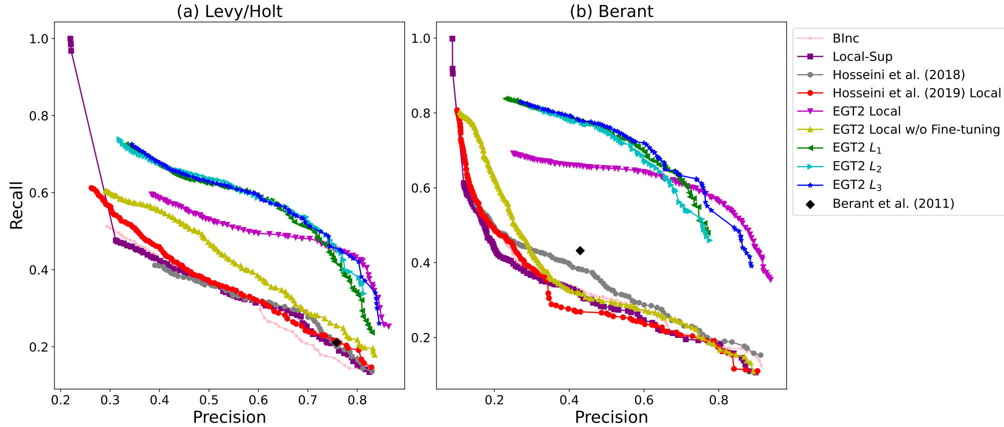

Figure 2: The Precision-Recall Curves of different methods on (a) Levy/Holt Dataset and (b) Berant Dataset. The result of Berant et al. (2011) is shown as a point, as they generate entailment graphs without weight.

Table 3: The number of testing examples appearing in entailment graphs learnt by corresponding models .

Methods	**Positive #**	**Negative #**
EGT2-Local	378	75
EGT2-L_1	642	174
EGT2-L_2	783	277
EGT2-L_3	685	190

models with transitivity constraints gain significant improvement on recall with far less expense on precision than EGT2-Local.

5.2 How the local model fine-tuning works?

As referred in Section 4.4, a new corpus is generated for fine-tuning the local model. We claim that the fine-tuning corpus helps to improve the performance of EGT2-Local by adapting it to the special sentence pattern by S, rather than offering additional data to fit the distribution of target datasets as traditional training datasets do. To prove this, we also test a simple supervised method, labelled as Local-Sup, which fits a 2-layers feedforward neural network on the fine-tuning corpus with cosine similarity, Weed, Lin and BInc scores as features. If the corpus acts as training dataset, the performance of Local-Sup should be obviously better than its unsupervised features.

As shown in Table 2, Local-Sup does not perform significantly better on Levy/Holt Dataset, and even worse on Berant Dataset than BInc, which is one of the inputting features of Local-Sup. The result illustrates the difference between the fine-tuning corpus and the evaluation datasets, and shows that the corpus plays a role as pattern adapting corpus rather than training dataset.

5.3 Why are global constraints helpful?

In Section 1, we expect that the improvement of soft transitivity constraints is attributed to the alleviation of data sparsity in corpus. To examine the sparsity before and after the applying of transitivity constraints, we count how many the positive and negative entailment relations in the Levy/Holt test set exactly appear in the local and global entailment graph respectively, and show the counting results in Table 3. All three soft transitivity constraints help to find more entailment relations than local entailment graph and therefore achieve better performance on the evaluation datasets. Although EGT2-L_2 finds the most entailment relations in the dataset in global stage, it finds more negative examples concurrently and thus performs worse than L_1 and L_3 as shown in Table 2. On the other hand, EGT2-L_1 and EGT2-L_3 obtain more proportions of positive examples by considering premise relations during the gradient calculation. The low confidence of hypothesis

Table 4: The major error types of false positive and false negative predictions by EGT2-L_3 in Levy/Holt test set, with predicted scores.

Error Types	**Examples**
False Negative	
Sparsity (46%)	Pain relieves by application of Chloroform. $\rightarrow$ Chloroform reduces pain. (0.0)
Under-weighted Relations (23%)	The Druids build the Stonehenge. $\rightarrow$ The Druids construct the Stonehenge. (0.558)
Dataset Wrong Labels (31%)	Salicylates reduces pain. $\rightarrow$ Salicylates is given for pain. (0.034)
False Positive	
Spurious Correlation (68%)	The cat sleeps on a fur. $\rightarrow$ The cat has a fur. (0.683)
Lemma-based Process (5%)	Lincoln comes to New York. $\rightarrow$ Lincoln comes from New York. (0.867)
Dataset Wrong Labels (27%)	The lamps are made of metal. $\rightarrow$ the lamps are made of metal. (1.0)

relationship $W_{a,c}$ should be helpful to detect spurious premises $W_{a,b}$ and $W_{b,c}$. Therefore, EGT2-L_3 slightly outperforms EGT2-L_1 as the gradients of $W_{a,b}$ and $W_{b,c}$ in L_3 are related to the hypothesis relationship $W_{a,c}$.

We have also applied the soft transitivity constraints on the local graph with BInc and Hosseini et al. (2019), but observed only slightly improvement of performance, as $.155 \rightarrow .157$ and $.167 \rightarrow .170$ for EGT2-L_3 on PRC of Levy/Holt Dataset respectively. Comparing it with the significant improvement based on EGT2-Local, we claim that the high-quality local entailment graphs are the basis of effective soft transitivity constraints.

5.4 Error Analysis

We randomly sample and analyze 100 false positive (FP) examples and 100 false negative (FN) examples from Levy/Holt test set according to predictions by EGT2-L_3. We manually setup the decision threshold as 0.574 to make the precision level close to 0.76, which is the same as Berant et al. (2011). The major error types are shown in Table 4. Although the global constraint is used, about half of FN errors are due to the data sparsity where the entailment relations are not found in the entailment graph. When compared with the results in Hosseini et al. (2018), EGT2-L_3 reduces the ratio of *Sparsity* in FN errors from 93% to 46% with stronger alleviation ability of data sparsity. About a quarter of FN are caused by the *Under-weighted Relations* in the graph, where EGT2 finds the entailment relations but gives them scores lower than the threshold.

Most of FP errors are caused by the *Spurious Correlation* as these relations are too fraudulent for EGT2 to see through their spurious relationships and consequently given high scores. A few FP errors are caused by *Lemma-based Processing in LM inevitably, but the ratio still reduces from 12% in Hosseini et al. (2018) to 5%. The result indicates that our fine-tuned LM can handle the predicates even with similar surface forms and contexts better than parsing-based distributional local features.*

6 Conclusions

In this paper, we propose a novel typed entailment graphs learning framework, EGT2, which utilizes fine-tuned textual entailment LM to calculate local entailment scores and applies soft transitivity constraints to learn global entailment graphs in gradient-based method. The transitivity constraints are achieved by carefully designed loss functions, and effectively boost the quality of local entailment graphsBy using the fine-tuned local LM and global soft constraints, EGT2 does not rely on distributional features, and can be easily applied to large-scale graphs. Experiments on standard benchmark datasets show that EGT2 achieves significantly better performance than existing state-of-the-art entailment graph methods.

REFERENCES

Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 610–619, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `https://aclanthology.org/P11-1062`.

Jonathan Berant, Ido Dagan, Meni Adler, and Jacob Goldberger. Efficient tree-based approximation for entailment graph learning. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 117–125, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://aclanthology.org/P12-1013`.

Jonathan Berant, Noga Alon, Ido Dagan, and Jacob Goldberger. Efficient global learning of entailment graphs. *Computational Linguistics*, 41(2):221–263, June 2015. doi: 10.1162/COLI_a_00220. URL `https://aclanthology.org/J15-2003`.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, 2008.

YL Cun, L Bottou, G Orr, and K Muller. Efficient backprop, neural networks: Tricks of the trade. *Lecture notes in computer sciences*, 1524:5–50, 1998.

Maayan Geffet and Ido Dagan. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 107–114, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219854. URL `https://aclanthology.org/P05-1014`.

Liane Guillou, Sander Bijl de Vroe, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. Incorporating temporal information in entailment graph mining. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pp. 60–71, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.textgraphs-1.7. URL `https://aclanthology.org/2020.textgraphs-1.7`.

Madan M Gupta and J11043360726 Qi. Theory of t-norms and fuzzy inference methods. *Fuzzy sets and systems*, 40(3):431–450, 1991.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Xavier Holt. Probabilistic models of relational implication. *arXiv preprint arXiv:1907.12048*, 2018.

Mohammad Javad Hosseini, Nathanael Chambers, Siva Reddy, Xavier R. Holt, Shay B. Cohen, Mark Johnson, and Mark Steedman. Learning typed entailment graphs with global soft constraints. *Transactions of the Association for Computational Linguistics*, 6:703–717, 2018. doi: 10.1162/tacl_a_00250. URL `https://aclanthology.org/Q18-1048`.

Mohammad Javad Hosseini, Shay B. Cohen, Mark Johnson, and Mark Steedman. Duality of link prediction and entailment graph induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4736–4746, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1468. URL `https://aclanthology.org/P19-1468`.

Mohammad Javad Hosseini, Shay B Cohen, Mark Johnson, and Mark Steedman. Open-domain contextual link prediction and its complementarity with entailment graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2790–2802, 2021.

Erich Peter Klement, Radko Mesiar, and Endre Pap. *Triangular norms*, volume 8. Springer Science & Business Media, 2013.

Omer Levy and Ido Dagan. Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 249–255, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2041. URL https://aclanthology.org/P16-2041.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. A logic-driven framework for consistency of neural models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3924–3935, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1405. URL https://aclanthology.org/D19-1405.

Dekang Lin. Automatic retrieval and clustering of similar words. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pp. 768–774, Montreal, Quebec, Canada, August 1998. Association for Computational Linguistics. doi: 10.3115/980691.980696. URL https://aclanthology.org/P98-2127.

Dekang Lin and Patrick Pantel. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360, 2001.

Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

Nick McKenna, Liane Guillou, Mohammad Javad Hosseini, Sander Bijl de Vroe, Mark Johnson, and Mark Steedman. Multivalent entailment graphs for question answering. *arXiv preprint arXiv:2104.07846*, 2021.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 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)*, pp. 425–430, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2070. URL https://aclanthology.org/P15-2070.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Siva Reddy, Mirella Lapata, and Mark Steedman. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392, 2014. doi: 10.1162/tacl_a_00190. URL https://aclanthology.org/Q14-1030.

Martin Schmitt and Hinrich Schütze. Language models for lexical inference in context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1267–1280, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.108. URL https://aclanthology.org/2021.eacl-main.108.

Idan Szpektor and Ido Dagan. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 849–856, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL https://aclanthology.org/C08-1107.

Julie Weeds and David Weir. A general framework for distributional similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp. 81–88, 2003. URL https://aclanthology.org/W03-1011.

Adina Williams, Nikita Nangia, and Samuel Bowman. 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)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Congle Zhang and Daniel S. Weld. Harvesting parallel news streams to generate paraphrases of event relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1776–1786, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1183.