Smoothing Entailment Graphs with Language Models

Anonymous ACL submission

Abstract

The diversity and Zipfian frequency distribution of natural language predicates in corpora leads to sparsity in Entailment Graphs (EGs) built by Open Relation Extraction (ORE). EGs are theoretically-founded and computationally efficient, but as symbolic models for natural language inference, they fail if a novel premise or hypothesis vertex is missing at test-time. We introduce a theory of optimal graph smoothing to overcome vertex sparsity by constructing transitive chains. We then demonstrate an efficient, open-domain smoothing method using an off-the-shelf Language Model to find approximations of missing premise predicates, improving recall by 25.1 and 16.3 percentage points on two difficult directional entailment datasets while raising average precision. Further, in a recent QA task, we show that EG smoothing is most useful for answering questions with lesser supporting text, where missing predicates are more costly. Finally, in controlled experiments with WordNet we show that hypothesis smoothing is difficult, but possible in principle. [1]

1 Introduction

An Entailment Graph (EG) is a learned structure for making natural language inferences of the form [premise] *entails* [hypothesis], such as "if Arsenal **defeated** Man United, then Arsenal **played** Man United." An EG consists of a set of vertices (typed natural language predicates), and a set of directed edges constituting entailments between predicates. They are constructed in an unsupervised manner using the Distributional Inclusion Hypothesis (Geffet and Dagan, 2005): a representation is generated for each predicate based on its distribution with arguments in a training corpus, and representation subsumption is used for learning directional entailments between predicates.

EGs are useful in tasks like Knowledge Graph link prediction (Hosseini et al., 2019, 2021) and question answering from text (Lewis and Steedman, 2013; McKenna et al., 2021). EG learning is unsupervised: building them only requires a parser and entity linker for a new language domain (Li et al., 2022b). Compared to Language Models, EGs are extremely data- and compute-efficient, requiring <2 days to train on 2GB of unlabeled text using a single GPU (Hosseini et al., 2021). Further, EGs are editable and also explainable, because decisions can be traced back to distinct sentences on a task.

However, EGs suffer from two kinds of sparsity. One is *edge sparsity*, when two predicates are not observed with co-occurring entities, so can't be connected together. Recent work improves on EG

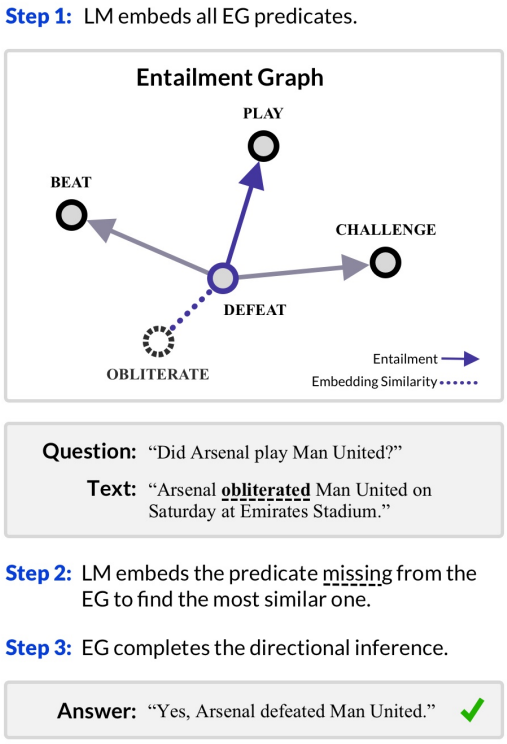

Figure 1: The question "Did Arsenal play Man United?" cannot be answered because the predicate "obliterate" from the text isn't in the Entailment Graph. An LM embeds "obliterate" so a nearest neighbor in the EG can be found, completing the directional inference.

[1] Code available at www.github.com/anonymous

connectivity (Berant et al., 2015; Hosseini, 2021; Chen et al., 2022) but little attention has been paid to *vertex sparsity*, arising when a predicate is not seen at all in training. Because EGs are learned structures of predicates, they cannot handle unseen queries: in an inference task, if *either* the premise or hypothesis predicate is not seen in training, no entailment edge can be learned. In fact, many EG demonstrations achieve just 50% of task recall.

Predicates occur in a Zipfian frequency distribution with an unbounded tail of rare predicates, so it is impractical to simply scale up distributional learning for predicate symbols.

Modern Language Models combine representations of subword tokens to solve a similar issue (Peters et al., 2018; Devlin et al., 2019), and recent scaling of LMs has lead to breakthrough performance on many tasks (Hoffmann et al., 2022; Wei et al., 2022), offering relief to sparsity problems via techniques like in-context learning (Brown et al., 2020). However, as LMs scale in size and compute they bring new problems: they require ballooning GPU resources to train or run, unavailable to most institutions; or are costly to query via API; and centralizing models under a few private companies opens challenges of data privacy. We are thus motivated to research lower-compute and more data-efficient methods which run on the scale of a single GPU. We offer three contributions toward improving an existing EG with the benefits of modern embeddings from a small pretrained LM:

(1) A theory for optimal smoothing of EG vertices by constructing transitive chains, taking account of a distinction between premise and hypothesis in this process.

(2) A novel, low-compute method for unsupervised smoothing of EG vertices using an LM to find approximations of missing premises. We improve recall by 25.1 and 16.3 percentage points on two directional entailment datasets while raising average precision. We also explain why hypothesis smoothing is possible, but more difficult.

(3) An application to Boolean Question Answering, where smoothing improves over two baselines on questions with sparse supporting context.

2 Background

Research on unsupervised Entailment Graph induction has mainly oriented toward edges: overcoming edge sparsity using graph properties like transitivity (Berant et al., 2015; Hosseini et al., 2018; Chen et al., 2022), incorporating contextual or extralinguistic information to improve edge precision (Hosseini et al., 2021; Guillou et al., 2020), and research into the underlying theory of the Distributional Inclusion Hypothesis (Kartsaklis and Sadrzadeh, 2016; McKenna et al., 2021). However, none of these address vertex sparsity.

Following innovations in tokenization like WordPiece (Devlin et al., 2019), we leverage subsymbolic encoding by an LM in this work as a means of smoothing, to generalize beyond a fixed vocabulary of predicates. Our most direct comparison is with Schmitt and Schütze (2021) who apply contemporary prompting techniques with the computationally tractable RoBERTa (Liu et al., 2019) to learn open-domain predicate entailment. They finetune on premise-hypothesis pairs and labels from the development split of the Levy/Holt NLI dataset (Holt, 2018), used in our experiments. They use templates like "[hypothesis], because [premise]" which are encoded by the LM, then classified true/false. They report high scores on datasets, but Li et al. (2022a) have shown that despite excelling at paraphrase detection, rather than learning directional inference (e.g. BUY $\models$ OWN and OWN $\not\models$ BUY), this technique picks up dataset artifacts spuriously correlated with the labels in Levy/Holt. In contrast, our approach combines the strengths of each: open-domain encoding using a computationally tractable LM with the directional capability of an EG.

3 Theory of Smoothing

We first present a theory for optimal smoothing of an EG which overcomes the problem of vertex sparsity, then discuss the theoretical intuition behind applying an LM as an open-domain smoother. We distinguish ways to **P-smooth** premises and **H-smooth** hypotheses.

We argue that it is most important when modifying EG predictions by smoothing to maintain the EG's strong directional inference capability. A **directional inference** is stricter than paraphrase or similarity, in that it is true only in one direction, but not both, e.g. DEFEAT $\models$ PLAY but PLAY $\not\models$ DEFEAT. Directional inferences are difficult, but crucial to language understanding.

3.1 Directionality by Transitive Chaining

We present a theory for optimal vertex smoothing of a symbolic inference model such as an EG,

which maintains directionality by constructing transitive chains, and distinguishing the *role* of the proposition as premise or hypothesis.

We start with a query entailment relation $Q : p \vDash h$, with unknown truth value to be verified by a model which is missing entries for at least p or h. We define *smoothing* as the process of generating a new relation Q_s suitable for the model by identifying a replacement predicate p' and/or h' within the model's vocabulary. We claim that to maintain directional precision, this must be done by identifying a p' (or h') related to p (or h) such that a transitive chain is constructed, as in the cases below. By this transitivity, confirmation of Q_s is leveraged to confirm Q.

1. **Generalize P**. Identify a more general premise p' in the EG such that $p \vDash p'$. This yields a new $Q_s : p' \vDash h$.

   (Q) "a obliterated b" $\vDash$ "a played b"
   ⫪

   (Q_s) "a beat b" $\vDash$ "a played b"

   $p \vDash p'$ is known, so if the EG confirms $p' \vDash h$, then $p \vDash h$ is confirmed by transitivity.

2. **Specialize H**. Identify a more specialized hypothesis h' in the EG such that $h' \vDash h$. This yields a new $Q_s : p \vDash h'$.

   (Q) "a bought b" $\vDash$ "a shopped for b"
   ⫫

   (Q_s) "a bought b" $\vDash$ "a paid for b"

   If the EG confirms $p \vDash h'$, then also knowing $h' \vDash h$ confirms $p \vDash h$ by transitivity.

3. **Generalize P and Specialize H**. This is a combination: identify new p' and h' as above, yielding a new $Q_s : p' \vDash h'$.

   Knowing $p \vDash p'$ and $h' \vDash h$, if a model confirms $p' \vDash h'$, then $p \vDash h$ is confirmed by transitivity.

Of course, the success of this smoothing depends on being able to find p' such that $p \vDash p'$, and h' such that $h' \vDash h$. However, when an additional inference is found, it is likely to be correct, aiding model precision. By definition we cannot use the EG for this, and we turn to Language Models to identify replacement predicates.

3.2 LM Embeddings and Specificity

We assume that p' and h' are respectively among the nearest neighbors of p and h in the embedding space of the LM, and in this paper propose a method to leverage LM embeddings in an unsupervised way to find them. As defined later in §4, we embed a target query predicate and EG predicates, then search for the K nearest neighbors to the target in embedding space. We predict that doing so for a premise predicate will build a transitive chain satisfying the conditions of §3.1. We identify two factors which, combined, make predictions that are likely more semantically general than the target, which enables P-smoothing, but not H-smoothing.

(A) The LM training objective. Li et al. (2020) show that the masked language modeling objective in BERT induces a particular structure in its latent embedding space: on average, corpus-frequent words are embedded near the origin and infrequent ones further out. This is because of statistical learning, which biases LMs toward high frequency words since they are trained on a corpus to predict the most probable tokens. This objective leads LSTM-based LMs to produce a beneficially Zipfian frequency distribution of words (Takahashi and Tanaka-Ishii, 2017), and similar biases are evident in Transformers for generation like GPT-2 and XL-Net (Shwartz and Choi, 2020).

(B) The natural anti-correlation of word frequency with specificity in text. Probabilistically, the more frequent a word, the lower its "semantic content." Caraballo and Charniak (1999) show this for nouns, and this assumption is even used in the "IDF" component of TF-IDF (Spärck Jones, 1972).

These factors imply that embedding a vocabulary of EG predicates using an LM will result in a space densely populated toward the origin by corpus-frequent predicates. KNN-search starting from a target predicate embedding will likely return neighbors toward this dense origin, thus selecting more corpus-frequent, semantically general words. We illustrate further in §3.3.

This effect has even been studied elsewhere: in Machine Translation, frequency bias causes a quantified semantic generalizing effect from translation input to output (Vanmassenhove et al., 2021), dubbed "Machine Translationese" due to the artificially non-specific tone.

3.3 The Specificity Taxonomy

To relate frequency and generality for our purposes, we illustrate a hierarchical taxonomy of predicates ordered by specificity, following from the theories of natural categories and prototype instances (Rosch and Mervis, 1975; Rosch et al., 1976). We

place very general predicate categories at the top of this taxonomy such as "act" and "move," with concrete subcategories beneath, and highly specific ones at the bottom, like "innoculate" and "perambulate." Rosch et al define their middle "basic level categories" for nouns, containing everyday concepts like "dog" and "table," which are learned early by humans and are used most commonly among all categories, even by adults (Mervis et al., 1976). We assume an analogous basic level in a predicate taxonomy, too, in Figure 2.

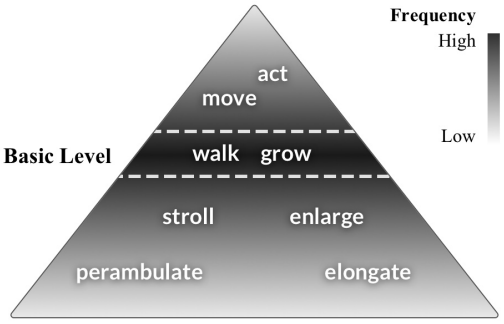

Figure 2: The specificity taxonomy. The basic level contains "everyday" predicates. Above becomes more general, and below becomes more concrete and specific. Usage frequency decreases away from the basic level.

There are few general categories at the top and many specific ones at the bottom (e.g., consider the many ways to "move," e.g. "walk," "sprint," "circumnavigate"). However, since basic level categories are the most frequently used, moving either up or down in the taxonomy accompanies a decrease in usage frequency. Above the basic level, predicates are fewer and more abstract, and can be infelicitous in daily use (e.g. calling a cat a "mammal" in Rosch's case or predicates like "actuate" in ours). Below, predicates are highly specialized for specific contexts, so there are more of them, and they are lower-frequency (e.g. "divebomb," "defenestrate").

This asymmetry motivates P-smoothing using an LM. A predicate z is likely to be missing from an EG if it is corpus-infrequent, thus likely specific. Randomly sampling another EG predicate z' neighboring z in embedding space, but sampled *proportional* to observed frequencies, is likely to return a predicate of higher frequency, toward the basic level, which is usually higher in the specificity taxonomy. Thus given z, a frequency-proportional sample z' is likely to be more general than z, usable for P-smoothing to construct a transitive chain.

4 Experimental Methods

In this work we consider Entailment Graphs of typed binary predicates. An EG is defined as $G = (V, E)$, consisting of a set of vertices V of natural language predicates (with argument types in the set $\mathcal{T}$), and directed edges E indicating entailments.

Binary predicates in V have two argument slots labeled with their types. For example, the predicate TRAVEL.TO(:person, :location) $\in V$, and the types :person, :location $\in \mathcal{T}$. An example entailment is TRAVEL.TO(:person, :location) $\models$ ARRIVE.AT(:person, :location) $\in E$.

Our smoothing method may be applied to any existing EG. In this work we show the complementary benefits of vertex-smoothing with existing methods in improving edge sparsity by comparing two related baseline models, described in §5. These EGs are learned from the same set of vertices, but are constructed differently so have different edges. The FIGER type system is used for these experiments (Ling and Weld, 2012), where $|\mathcal{T}| = 49$, and these models typically have up to $|\mathcal{T}|^2 = 49^2$ typed subgraphs $g \in \mathcal{G}$. Typing disambiguates senses of the same predicate, which improves precision of inferences. For example, KILL(:medicine, :disease) learned in the typed subgraph $g^{(\textit{medicine-disease})}$ has a different meaning and entailments than KILL(:person, :person).

4.1 Nearest Neighbors Search

Our method assumes that existing EGs contain enough information to enable discovery of suitable replacements for an unseen target predicate that are already present in the graph, using an LM. For example, in the sports domain, the EG may be missing a rare predicate OBLITERATE but contain similar predicates BEAT and DEFEAT which can be found as close neighbors in Language Model embedding space. These nearby predicates are expected to have similar semantics (and entailments) to the unseen target predicate, and will thus be suitable replacements. See Figure 1 for an illustration.

We define the smoothed retrieval function S, which replaces the typical method for retrieving a target predicate vertex x from a typed subgraph $g^{(t)} = (V^{(t)}, E^{(t)})$, with typing $t \in \{\mathcal{T} \times \mathcal{T}\}$.

Ahead of test-time, for each typed subgraph $g^{(t)}$ we encode the EG predicate vertices $V^{(t)}$ as a matrix $\mathbf{V}^{(t)}$. For each predicate $v_i^{(t)} \in V^{(t)}$, we encode $L(v_i^{(t)}) = \mathbf{v}_i^{(t)}$, a row vector $\mathbf{v}_i^{(t)} \in \mathbf{V}^{(t)}$.

At test-time we encode a corresponding vector

x : (join.1,join.2)#person#organization ⇒ "person **join** organization"
x : (give.2,give.to.2)#medicine#person ⇒ "**give** medicine **to** person"
x : (export.1,export.to.2)#location_1#location_2 ⇒ "location_1 **export to** location_2"

Table 1: A typed predicate x is converted to a sentence (shown), then encoded with an LM using $L(x)$, which outputs the average over **predicate** WordPiece vectors.

for the target predicate x, $L(x) = \mathbf{x}$. Then S retrieves the K-nearest neighbors of x in $g^{(t)}$:

$$S(x, g^{(t)}, K) =$$
$$\{v_i^{(t)} \mid v_i^{(t)} \in V^{(t)}, \text{ if } \mathbf{v}_i^{(t)} \in \mathit{KNN}(\mathbf{x}, \mathbf{V}^{(t)}, K)\}$$

$L(\cdot)$ is an encoder for a typed natural language predicate using a pretrained LM. First, a short sentence is constructed from the predicate using the types as generic arguments, and then the sentence is encoded by the LM (see Table 1 for examples). For these experiments we use RoBERTa (Liu et al., 2019) as the encoder. We extract the embeddings of WordPieces corresponding to the predicate, and average them into the resulting predicate vector.

For the KNN search metric we use Euclidean Distance (L^2 norm) from the target vector $\mathbf{x}$ to vectors in $\mathbf{V}^{(t)}$. We precompute a BallTree using scikit-learn (Pedregosa et al., 2011) which spatially organizes the EG vectors to speed up search from linear in the number of vertices $|V^{(t)}|$ to log $|V^{(t)}|$.

4.2 Datasets

We demonstrate our smoothing method on two explicitly directional datasets, which test both directions of these inferences (a 50% positive/50% negative class balance).

Levy/Holt. This dataset (Holt, 2018; Levy and Dagan, 2016) has been explored thoroughly in previous work (Hosseini, 2021; Guillou et al., 2021; Li et al., 2022b; Chen et al., 2022). Importantly, it includes inverses for all queries, allowing systematic investigation of directionality, although it contains a high proportion of paraphrases and selection bias artifacts that can be picked up by finetuning in supervised models (Li et al., 2022a). We test on the 1,784 questions forming the purely directional subset, which is more challenging.

ANT. This is a new, high-quality dataset improving on Levy/Holt, which tests predicate entailment in the general domain (Guillou and Bijl de Vroe, 2023). It was created by expert annotation of entailment relations between clusters of predicate paraphrases, expanded automatically using WordNet and other dictionary resources into thousands of test questions of the format "given [premise], is [hypothesis] true?" We test on the directional subset of 2,930 questions.

"The audience applauded the comedian" ⊨ "The audience observed the comedian"
"Apple supported Samsung" ⊨ "Apple had an opinion on Samsung"
"The laptop was assessed against the criteria" ⊭ "The laptop satisfied the criteria"

Table 2: Example queries, ANT (dev) directional subset.

See Table 2 for dataset examples. Each comes preprocessed with argument types from CoreNLP (Manning et al., 2014; Finkel et al., 2005), roughly aligning with EG FIGER types. We use the MoNTEE system (Bijl de Vroe et al., 2021) to extract the typed predicate relations (x) shown in Table 1, which are used as queries to Entailment Graphs.

4.3 Models

We smooth two recent Entailment Graphs which previously scored highly amongst unsupervised models on the full Levy/Holt dataset. Importantly, they are constructed from the same set of predicate vertices but have different edges, so we can observe how vertex- and edge-improvements combine.

GBL. The EG of Hosseini et al. (2018), which introduces a "globalizing" graph-based method to improve the edges after "local" EG learning.

CTX. The state-of-the-art contextualized EG of Hosseini et al. (2021), which improves over GBL edges by augmenting local learning with a contextual link-prediction objective, before globalizing.

GBL-P / GBL-H and **CTX-P / CTX-H**. We apply an LM separately for both P- and H-smoothing on GBL and CTX. For this we use RoBERTa (Liu et al., 2019), a well-tested, off-the-shelf Language Model of tractable size for running on a single GPU, which has pretrained on 160GB of unlabeled text. We use the LM to produce embeddings for smoothing the EG.

S&S. The finetuned RoBERTa model of Schmitt and Schütze (2021) (discussed in §2). We insert each premise/hypothesis pair into their 4 prompt templates, and take the maximum entailment score as the model prediction for the pair. Li et al. (2022a) find that this model has overfit to artifacts present in Levy/Holt, so we compare with it on a different

Figure 3: LM smoothing on ANT. Comparison of P- and H-smoothing on the CTX model. We explore $K \in \{2, 3, 4\}$ and show the best $K_{prem} = 4$ and $K_{hyp} = 2$.

question answering task in §6.

5 Experiment 1: Entailment Detection

We run two experiments on both Levy/Holt and ANT. (1) We apply our unsupervised smoothing to augment the ***Premise*** of each test entailment, generating K new target premise predicates. Separately, (2) we smooth the ***Hypothesis*** of each test entailment the same way. For both we try different values of the hyperparameter $K \in \{2, 3, 4\}$.

Plots for model performances are shown in Figure 3, in which we compare P-smoothing vs. H-smoothing of the CTX graph using $K_{prem} = 4$ and $K_{hyp} = 2$, chosen for producing the best AUC$_n$ (see Appendix A for all results). In Appendix B we also show P-smoothing in particular of the CTX graph vs. the GBL graph. For all models (best K selected) on both datasets we show summary statistics in Table 3, including *normalized* area under the precision-recall curve (AUC$_n$) and average precision (AP) across the recall range. A sample of model outputs is shown in Table 4.

Li et al. (2022a) introduce AUC$_n$, a fair way to compare models which may achieve different maximum recalls. It computes only the area under the precision-recall curve *above* the random-guess baseline for the dataset, so it is highly discerning compared to AUC, which can inflate performance when there is a high random baseline. In our case, the high 50% random baseline means that AUC$_n$ scores are systematically much lower than AUC.

As predicted, our method of selecting nearest-neighbors of a target predicate in an EG using their LM embedding distance has different behavior for P-smoothing than H-smoothing. We observe that P-smoothing is very beneficial to both the recall and precision of both Entailment Graphs it is applied to, with a slight advantage in AUC$_n$ to higher values of K. When applied to the SOTA model

	ANT		Levy/Holt	
Model	AUC$_n$	AP	AUC$_n$	AP
GBL	3.79	58.36	3.01	55.82
GBL-P$_{K=4}$	**13.91**	**64.71**	**9.95**	**60.70**
GBL-H$_{K=2}$	1.41	52.57	1.09	52.05
CTX	15.44	65.66	9.40	60.19
CTX-P$_{K=4}$	**25.86**	**67.47**	**13.45**	**60.80**
CTX-H$_{K=2}$	9.94	58.52	8.33	57.97

Table 3: P- and H-smoothing, compared to unsmoothed models. We report normalized area under the precision-recall curve (AUC$_n$) and average precision (AP).

Predicate Missing from EG	**Nearest Neighbors by Embedding Dist.**
DISCREDIT(:person, :thing)	PROBE, ACCUSE
CRACK.UP.AT(:person, :written_work)	MAKE.JOKE.AT, YELL.AT
MINIMIZE(:organization, :thing)	SOFTEN, EVADE
REBUKE(:person, :person)	OPPOSE, REMIND

Table 4: Sample of CTX outputs on ANT. A target PREDICATE(type1, type2) missing from CTX yields K=2 closest CTX predicates in LM embedding space.

CTX on the ANT dataset, our smoothing method increases maximum recall by 25.1 absolute percentage points (pp) to 74.3% while increasing average precision from 65.66% to 67.47%. On Levy/Holt we increase maximum recall by 16.3 absolute pp to 62.7% while slightly raising average precision. However, H-smoothing with the LM is highly detrimental: despite improving recall, average precision on ANT is cut to 58.52%, and the lowest confidence predictions are at chance (50% precision).

We also note that P-smoothing greatly improves recall and precision when applied to *both* GBL and CTX graphs. This shows the complementary nature of improving vertex sparsity with improving edge sparsity in EGs: these techniques improve different aspects, which can be applied together. Since effects are similar for both EGs, from now on we show results only for CTX, and report additional results for the weaker GBL in Appendix B.

6 Experiment 2: Question Answering

P-smoothing with an LM is effective in intrinsic tests, and we now experiment with LM smoothing in application on a "real" task. We test on the Boolean Open QA task (BoOQA) (Li et al., 2022a), in which models answer true/false questions about entities mentioned in news articles from multiple sources. BoOQA questions are chosen to be adver-

Context Size	CTX	CTX-P	CTX-H	S&S
$[2, 5)$	20.05	**20.66**	19.07	17.00
$[5, 10)$	29.13	**29.17**	29.01	23.05
$[10, 15)$	**32.32**	32.31	32.25	24.98
15+	**36.58**	36.57	36.51	26.13
All Questions	21.26	**21.74**	20.64	16.99

Table 5: Effect of P- and H-smoothing vs. baseline CTX and S&S across context sizes. AUC_n is reported.

sarial to simple similarity baselines, and EGs have proven useful by using directional reasoning.

6.1 Boolean Open-Domain QA

BoOQA is a task over open domain news articles, with questions formed by extracting triples of (entity, relation, entity), in the format "is it true that `<triple>`?" *Context statements* are other triples sourced from the articles concerning the same question entities, and the task is to compare each context statement with the question itself. If any context statement entails the question by means of its relation, the question can be labeled "true," otherwise "false." BoOQA also contains false questions derived from true ones, so models must decide carefully what is supported by evidence and what isn't.

We address vertex sparsity in a natural setting, so we relax the original entity restriction of Li et al. (2022a): instead of sampling questions about frequently-mentioned entities (which always have many context statements to decide from), we increase the challenge by sampling from the natural distribution of entities, regardless of popularity.

6.2 Results Across Context Sizes

Results corroborate the earlier tests: P-smoothing improves AUC_n from 21.26% to 21.74% over all questions, while H-smoothing worsens to 20.64% (as in §4, AUC_n is systematically lower than AUC). We also outperform Schmitt and Schütze (2021), our most direct competition which uses a tractable-size LM. Despite facility to encode any predicate, it lacks directional precision useful for this task.

To understand when smoothing an EG is most helpful, we further analyze the effect on different *context size bands*. For each question, we count the number of context sentences available to answer it; we then bucket the questions into size bands of $[2, 5)$, $[5, 10)$, $[10, 15)$, 15+. On these bands we compare an unsmoothed model with P-smoothing and H-smoothing, reported in Table 5.

The benefit of P-smoothing is greatest in the lowest band $f < 5$, and diminishes in higher bands. This is because in the lower bands there are fewer context statements which may be used to answer the question, increasing difficulty. Here the EGs are more prone to sparsity, because missing even a few context predicates devastates its chance to answer the question. In fact, the proportion of questions for which all context relations are missing from the EG is 1.5% for $f > 15$, but 32.7% for $f < 5$.

7 Experiment 3: P- and H-Smoothing with WordNet

P-smoothing with an LM achieves different performance than H-smoothing. We now confirm this is due to semantic generalization (in line with our theory in §3.1) by using controlled experiments with WordNet (Fellbaum, 1998). We show that directing the search for replacement predicates by constructing transitive chains provides a means for smoothing both premise and hypothesis.

7.1 Controlled Search with WordNet

We re-run the experiment of §4 by smoothing the CTX (Hosseini et al., 2021) model on the ANT directional dataset (GBL shown in Appendix B). However, in this design the target premise or hypothesis is approximated without the LM. Instead, we generate replacements using specific WordNet relations. We note that WordNet was partly used in ANT's construction, so this demonstration is meant to explain rather than claim a dataset high score.

In this test, we choose specific WordNet lexical relations as instances of entailment, then generate smoothing predictions from the WN database. We choose **hyponomy** for specialization and **hypernymy** for generalization, and compare both relations for both P- and H-smoothing. To illustrate, if smoothing by generalizing, given a predicate "elect," we retrieve WN hypernyms like "choose."

For each CCG-parsed predicate, we query WordNet for the head word and extract results from the first word sense, then insert into the predicate. For example, from `(receive.2,receive.from.2)` the WN query *hyponym*("receive") ⇒ "inherit" generates `(inherit.2,inherit.from.2)` which is used to query the EG.

7.2 Results

Results are shown in Figure 4. Importantly, from these plots we note a switch in performance of hypernyms and hyponyms between P- and H-smoothing on CTX (similar results for GBL, see ap-

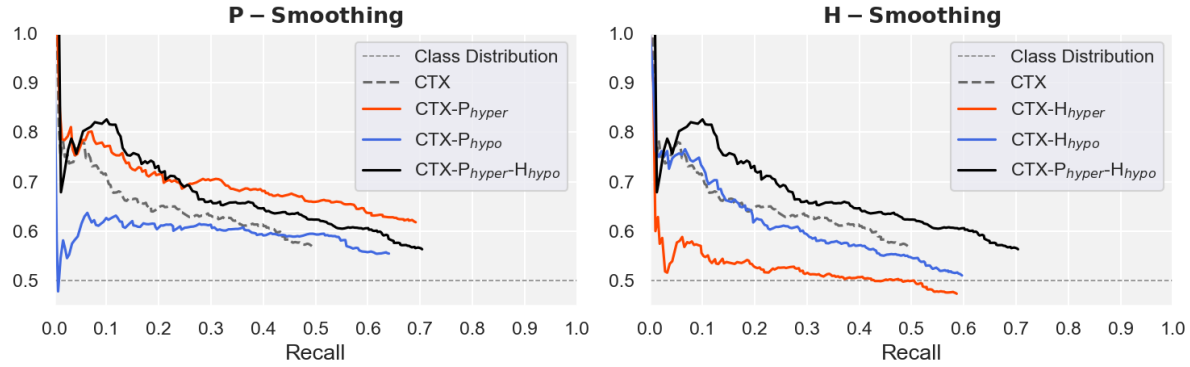

Figure 4: Comparison of WordNet relations used in smoothing P(remise), H(ypothesis), and P+H, with CTX graphs on the ANT dataset. Hypernyms are shown useful for P-smoothing, and hyponyms less so for H-smoothing.

pendix). It is clear that generalizing the premise using hypernyms is highly effective in terms of recall and precision, but specializing with hyponyms is extremely damaging to precision. For the hypothesis, the reverse is true: specializing with hyponyms can lead to some performance gains, while generalizing with hypernyms worsens performance.

These results nearly replicate the behavior of the LM-smoother in §4, verifying that nearest neighbor search in LM embedding space has a semantically generalizing effect suitable for P-smoothing. Table 4 shows examples of generalized predictions.

Finally, we note P-smoothing with WordNet performs similarly to the LM in this "laboratory" setting (see Appendix C), but an LM smoother is still preferable due to being fully automatic and open-domain, handling new words, misspellings, etc.

7.3 Discussion

We note two phenomena of interest. (1) For both CTX and GBL, performance is boosted in the low-recall/high-precision range when using both optimal smoothers ($P_{hyper} + H_{hypo}$), higher than using either smoother individually. (2) Additionally, H_{hypo} is the better H smoother tested, though it appears unreliable on its own without P smoothing: H_{hypo} is not useful for smoothing CTX (it does improve the weaker GBL, see Appendix B).

Both of these phenomena are likely related to data frequency. Generalized hypernyms such as BEAT and USE are quite common in training data, and therefore have more learned edges in the EG with high quality edge weights. However, specialized hyponyms like ELONGATE can be extremely sparse in training data, leading to poorer learned representations and fewer edges. Phenomenon (1) shows that using a frequently-occurring smoothed premise of high quality yields better odds of finding an edge to a smoothed hypothesis, leading to some performance gains over either smoother individually. Phenomenon (2) suggests that H-smoothing may be naturally more difficult than P-smoothing, and less stable due to sparsity of hyponyms (specializations) in corpora. If h is missing from the EG (meaning it wasn't seen in training) then deriving a candidate h' specialized from h will also be unlikely to occur in training, thus even if found in the EG it may have few or poorly learned edges. Though it can be beneficial to precision, data sparsity makes H-smoothing fundamentally harder.

8 Conclusion

We introduce a theory for optimal smoothing of an Entailment Graph by construction of transitive chains. Further, we show an unsupervised, open-domain method for P-smoothing an EG using Language Model embeddings, which improves both recall and precision on two difficult directional entailment datasets. We also test the method on a QA task, where we show its most useful benefit in difficult scenarios where limited context information is available. Our method is low-compute, combining an existing EG with a pretrained LM of tractable size for use with a single GPU, and it improves over two low-compute baselines: a SOTA EG and a finetuned RoBERTa-based prompting model.

We also demonstrate our theory of optimal smoothing by directing the search for predictions using WordNet relations. Our experiments replicate the behavior of the LM-based smoother, offering an explanation for why LM embeddings are useful for P-smoothing, but not H-smoothing, in terms of the semantic generalizing effect when searching a neighborhood in embedding space.

Limitations

In this work we present a simple "graph smoothing" method which leverages the natural structure in LM embedding space to find approximations of predicates missing from the EG, a major source of error.

This structure naturally benefits P-smoothing, because nearest neighbors search within LM embedding space is biased toward returning predicates of higher frequency (in LM training data), which are likely to be more semantically general than the starting predicate. This generality bias is helpful for P-smoothing. However, generalizing is detrimental to H-smoothing, which requires specialization. While we show a proof of specialization and empirical evidence using WordNet, solving H-smoothing in an open domain using an unsupervised model such as a Language Model is left open in this work. It is likely that H-smoothing is a more difficult task than P-smoothing due to natural data sparsity as discussed in the paper. If a hypothesis is missing from the EG, it is likely to be a corpus-infrequent predicate, and specializing it will yield other predicates of low frequency, yielding poor odds of recovery.

Further, while the use of a sub-symbolic LM encoder theoretically enables inference using any premise predicate, it is still restricted to choosing approximations from the pre-set predicate vocabulary learned by the EG. Hosseini et al. (2021) show that EG learning may be scaled up easily, which may provide a sufficiently scaled vocabulary for any application, but exploring this is left for future work.

References

Jonathan Berant, Noga Alon, Ido Dagan, and Jacob Goldberger. 2015. Efficient global learning of entailment graphs. *Computational Linguistics*, 41(2):221–263.

Sander Bijl de Vroe, Liane Guillou, Miloš Stanojevic, Nick McKenna, and Mark Steedman. 2021. Modality and negation in event extraction. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021), online. Association for Computational Linguistics (ACL)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sharon A. Caraballo and Eugene Charniak. 1999. Determining the specificity of nouns from text. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Zhibin Chen, Yansong Feng, and Dongyan Zhao. 2022. Entailment graph learning with textual entailment and soft transitivity.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, page 363–370, USA. Association for Computational Linguistics.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114, Ann Arbor, Michigan. Association for Computational Linguistics.

Liane Guillou and Sander Bijl de Vroe. 2023. Ant dataset.

Liane Guillou, Sander Bijl de Vroe, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2020. Incorporating temporal information in entailment graph mining. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 60–71, Barcelona, Spain (Online). Association for Computational Linguistics.

Liane Guillou, Sander Bijl de Vroe, Mark Johnson, and Mark Steedman. 2021. Blindness to modality helps entailment graph mining. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 110–116, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models.

Xavier Holt. 2018. Probabilistic models of relational implication. Master's thesis, Macquarie University.

Mohammad Javad Hosseini. 2021. *Unsupervised Learning of Relational Entailment Graphs from Text*. Ph.D. thesis, University of Edinburgh.

Mohammad Javad Hosseini, Nathanael Chambers, Siva Reddy, Xavier R. Holt, Shay B. Cohen, Mark Johnson, and Mark Steedman. 2018. Learning typed entailment graphs with global soft constraints. *Transactions of the Association for Computational Linguistics*, 6:703–717.

Mohammad Javad Hosseini, Shay B. Cohen, Mark Johnson, and Mark Steedman. 2019. Duality of link prediction and entailment graph induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4736–4746, Florence, Italy. Association for Computational Linguistics.

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

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. 2016. Distributional inclusion hypothesis for tensor-based composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2849–2860, Osaka, Japan. The COLING 2016 Organizing Committee.

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 (Volume 2: Short Papers)*, pages 249–255, Berlin, Germany. Association for Computational Linguistics.

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

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Tianyi Li, Javad Hosseini, Sabine Weber, and Mark Steedman. 2022a. Language models are poor learners of directional inference. In *Findings of the Conference on Empirical Methods in Natural Language Processing*, page to appear. ACL.

Tianyi Li, Sabine Weber, Mohammad Javad Hosseini, Liane Guillou, and Mark Steedman. 2022b. Cross-lingual inference with a chinese entailment graph.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 94–100. AAAI Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Nick McKenna, Liane Guillou, Mohammad Javad Hosseini, Sander Bijl de Vroe, Mark Johnson, and Mark Steedman. 2021. Multivalent entailment graphs for question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10758–10768, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Carolyn B Mervis, Jack Catlin, and Eleanor Rosch. 1976. Relationships among goodness-of-example, category norms, and word frequency. *Bulletin of the psychonomic society*, 7(3):283–284.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Eleanor Rosch and Carolyn B Mervis. 1975. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7(4):573–605.

Eleanor Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem. 1976. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439.

Martin Schmitt and Hinrich Schütze. 2021. 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*, pages 1267–1280, Online. Association for Computational Linguistics.

Vered Shwartz and Yejin Choi. 2020. Do neural language models overcome reporting bias? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6863–6870, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Shuntaro Takahashi and Kumiko Tanaka-Ishii. 2017. Do neural nets learn statistical laws behind natural language? *PLOS ONE*, 12(12):1–17.

Eva Vanmassenhove, Dimitar Shterionov, and Matthew Gwilliam. 2021. Machine translationese: Effects of algorithmic bias on linguistic complexity in machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2203–2213, Online. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

A Hyperparameter Search

In §5 we test three values for hyperparameters K_{prem} and K_{hyp}, each from choices {2, 3, 4}. Figure 5 shows all smoothing combinations. We select $K_{prem} = 4$ and $K_{hyp} = 2$ in the main experiments due to having the highest AUC$_n$ values for P- and H-smoothing, respectively. We highlight a few trends. (1) higher K_{prem} appears better (most notably, $K_{prem} = 4$ yields slightly better recall than $K_{prem} = 2$), though it has diminishing returns. (2) lower K_{hyp} is better, because H-smoothing using an LM is actively harmful ($K_{hyp} = 0$, an unsmoothed EG, would "perform" better in practice!).

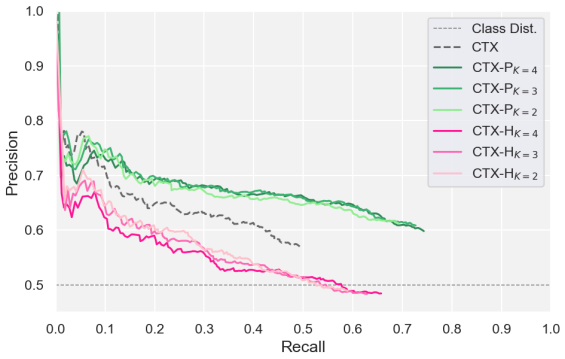

Figure 5: LM smoothing on the ANT dataset. Comparison of P- and H-smoothing CTX with different K_{prem} and K_{hyp} from choices {2, 3, 4}. *Higher values of K are shown more darkly.*

B The GBL Entailment Graph

We test the older GBL graph (Hosseini et al., 2018) on the ANT dataset. Results confirm findings on the newer CTX (Hosseini et al., 2021). Figure 6 shows results for the experiment in §4 but comparing P-smoothing with LM predictions for the CTX and GBL graphs. We note that base CTX performs much better than GBL, and that P-smoothing with an LM improves both GBL and CTX.

Figure 7 shows results for the experiment in §7 of smoothing an EG using WordNet relations, but we now show smoothing the older GBL graph. We observe similar results as with CTX: there is noticeable improvement over the base EG when smoothing either premises with hypernyms, hypotheses with hyponyms (stronger than when applied to CTX), or both combined.

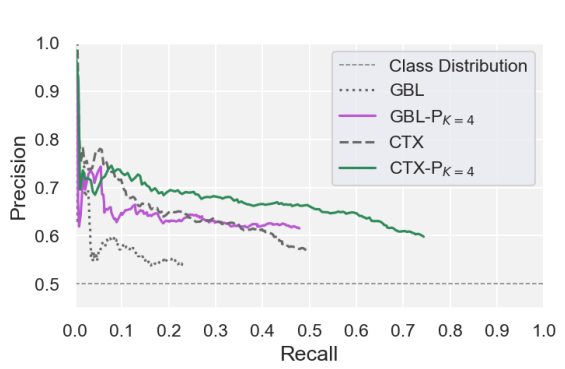

Figure 6: LM smoothing on the ANT dataset. Comparison of P-smoothing GBL and CTX with optimal K=4.

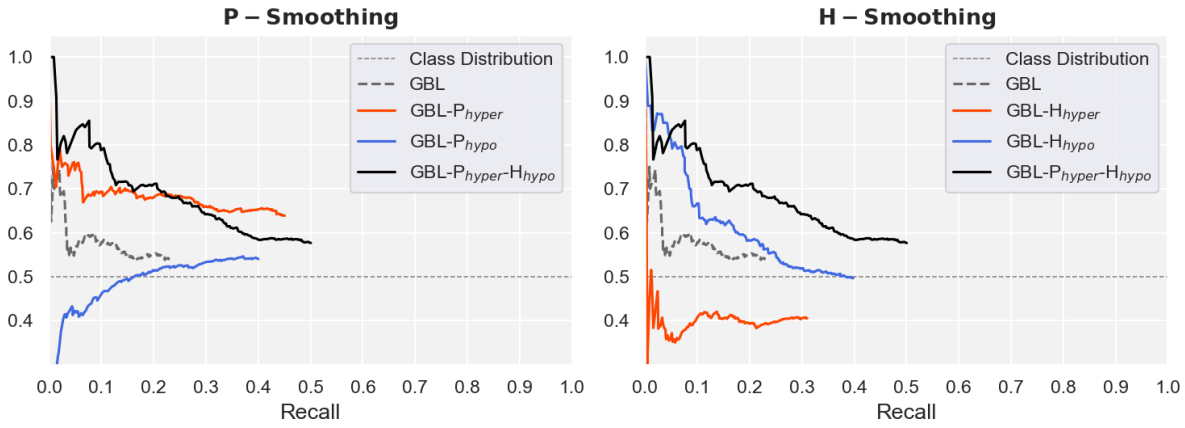

Figure 7: WordNet relations used to smooth P(remise), H(ypothesis), and P+H, with the Entailment Graph GBL on the ANT dataset. Hypernyms are useful for P-smoothing, and hyponyms for H-smoothing.

C P-Smoothing: LM vs. WordNet

In Figure 8 we show a comparison of P-smoothing between the LM (CTX-P$_{LM}$ AUC$_n$ = 25.86) and WordNet (CTX-P$_{hyper}$ AUC$_n$ = 27.39) on the ANT dataset. We note that although WordNet performs within about 1.5% of the LM smoother in this "laboratory" experiment, we believe the LM-smoother is preferable in use, because it is fully automatic to learn and apply, and because it encodes an open domain of predicates, which may include new words, misspellings, etc. that WordNet cannot handle.

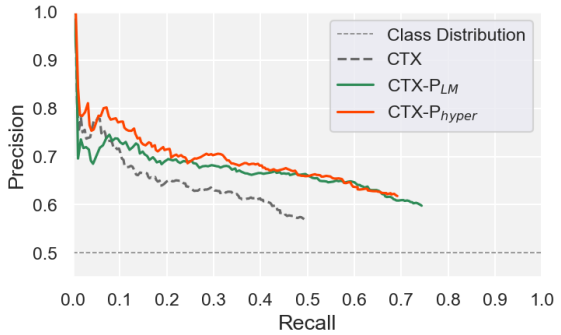

Figure 8: Comparison of P-smoothing methods on ANT: LM-based smoother and WordNet hypernym relations on the Entailment Graph CTX.