Smoothing Entailment Graphs with Language Models

Anonymous TACL submission

Abstract

The diversity and Zipfian frequency distribution of natural language predicates in corpora leads to sparsity when learning Entailment Graphs. As symbolic models for natural language inference, an EG cannot recover if missing a novel premise or hypothesis at test-time. In this paper we approach the problem of vertex sparsity by introducing a new method of graph smoothing, using a Language Model to find the nearest approximations of missing predicates. We improve recall by 25.1 and 16.3 absolute percentage points on two difficult directional entailment datasets while exceeding average precision, and show a complementarity with other improvements to edge sparsity. On an extrinsic QA task, we show that smoothing benefits the lower-resource questions, those with less available context. We further analyze language model embeddings and discuss why they are naturally suitable for premise-smoothing, but not hypothesis-smoothing. Finally, we formalize a theory for smoothing a symbolic inference method by constructing transitive chains to smooth both the premise and hypothesis.

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 edges (directional 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 these representations are used in learning directional entailments.

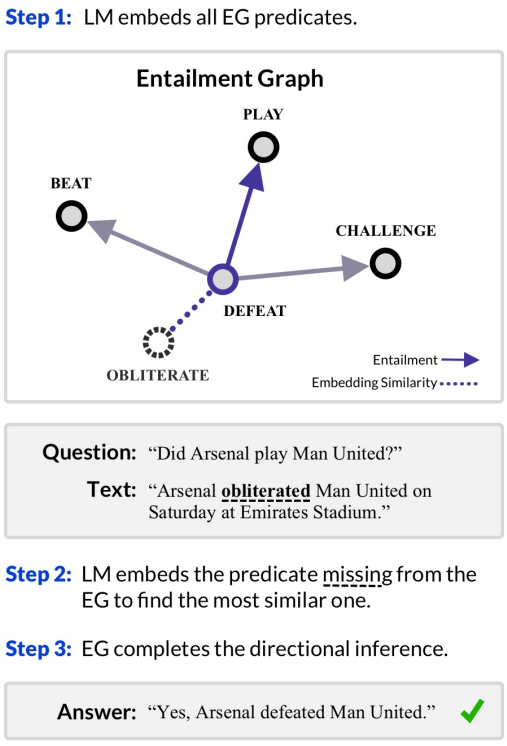

Figure 1: The question "Did Arsenal play Man United?" cannot be answered because the predicate "obliterate" from the text snippet isn't in the Entailment Graph. A Language Model embeds "obliterate" so a nearest neighbor in the EG can be found, completing the directional inference.

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); and as an unsupervised method, to build them only requires a parser and entity linker for a new language domain (Li et al., 2022b). Further, EGs are fully explainable, because model decisions can be traced back to sentences in training data.

However, EGs suffer from sparsity of two kinds. One kind is *edge sparsity*, arising from the fact that

authors usually omit facts that the reader can be expected to infer for themselves, making it hard to learn edges. Recent work has improved on EG connectivity (Berant et al., 2015; Hosseini, 2021; Chen et al., 2022) but little attention has been paid to the related problem of *vertex sparsity*, arising from predicates that are unseen at all in training. Because EGs are learned structures of predicates, they cannot reason about novel queries: in an inference task, if *either* the premise or hypothesis predicate has not been seen in training (thus is missing from the graph), there is no possibility to have learned an edge, and the model will have no chance to report an entailment. In fact, many EG demonstrations don't achieve more than 50% of task recall.

Like words, predicates occur in a Zipfian frequency distribution with an unboundedly long tail of rare predicates, so it is impractical to solve vertex sparsity by scaling up distributional learning.

Instead, we present a method for smoothing an Entailment Graph using a Language Model to search within the graph for approximations of a missing target predicate, completing otherwise impossible EG inferences. We illustrate the method in Figure 1. The paper offers three contributions:

1. A novel method for unsupervised smoothing of Entailment Graph vertices using a Language Model to find approximations of missing predicates.

2. An analysis of Language Model embedding space and a discussion of why this method is naturally suited to premise smoothing, but not hypothesis smoothing.

3. A theory for smoothing with high directional precision by constructing transitive inference chains, demonstrated on both premise and hypothesis.

2 Background

Unsupervised Entailment Graph research has mainly oriented toward edges: overcoming edge sparsity using graph properties like transitivity (Berant et al., 2010, 2015; Hosseini et al., 2018), 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). Recently, McKenna et al. (2021) interpret the DIH in terms of eventualities which may have variable argument numbers, learning edges between predicates of different valencies. Though this work expands the kinds of graph vertices, it does not address the problem of vertex sparsity, which is especially severe for binary predicates. To our knowledge, no other work in unsupervised entailment models has approached this issue of vertex sparsity.

Older language models like word2vec (Mikolov et al., 2013) learned representations for a fixed vocabulary of words, and couldn't be used to estimate probabilities for unseen words. Earlier methods like those based on n-grams smoothed the distribution using mathematical re-estimation. However, recent research in sub-symbolic character-based models like ELMo (Peters et al., 2018) and WordPiece models like BERT (Devlin et al., 2019), prove effective at generalizing from seen words to unseen. We leverage sub-symbolic encoding in this work as our means of smoothing, to generalize beyond a fixed vocabulary of predicates.

3 Smoothing an Entailment Graph using a Language Model

In this work we consider Entailment Graphs of typed binary predicates, as is the common mode of EG research. An Entailment Graph is defined $G = (V, E)$, 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 directional entailment TRAVEL.TO(:person, :location) $\models$ ARRIVE.AT(:person, :location) $\in E$.

Our smoothing method may be applied to any EG. In this work we show the complementary benefits of vertex-smoothing with existing methods in improving edge sparsity by comparing to two related baseline models, described in §3.3. 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 $|\mathcal{T}| = 49$. Typing aids EG precision by grouping predicates and their entailments by type-pair into $\mathcal{G}$ subgraphs: these models have up to $|\mathcal{T}|^2 = 49^2$ typed subgraphs $g \in \mathcal{G}$ in which learning is distributed. For example, the predicate KILL(:medicine, :disease) in the subgraph $g^{(\textit{medicine-disease})}$ has different learned

entailments than KILL(:person, :person).

3.1 Smoothing Method

Our method rests on the assumption that existing Entailment Graphs contain enough information to enable discovery of suitable replacements for an unseen target predicate that are already present in the graph, using a Language Model. 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 for the target predicate x, $L(x) = \mathbf{x}$. Then S retrieves the K-nearest neighbors of x in $g^{(t)}$:

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

We define $L(\cdot)$ and configure $\mathit{KNN}(\cdot)$ as follows.

$L(\cdot)$ is an unsupervised encoder for any typed natural language predicate using a pretrained Language Model. We first construct a short sentence from the typed predicate using each type as a standin argument in a CCG argument structure (Steedman, 2000), and then the sentence is encoded by the Language Model. For these experiments we use RoBERTa (Liu et al., 2019), a general-purpose contextual Language Model which shares a transformer architecture with other popular LMs but has robustly pretrained on 160GB of unlabeled text. We extract the embeddings of WordPieces corresponding to the predicate only, and average them to make the resulting predicate vector. See Table 1 for examples.

For the K-nearest neighbors search metric we use Euclidean Distance (l2 norm) from the target vector $\mathbf{x}$ in embedding space. We precompute a BallTree which spatially organizes the EG vectors to speed up search (Pedregosa et al., 2011). At best, this reduces search time from linear in the number of vertices $|V^{(t)}|$ to log $|V^{(t)}|$.

3.2 Testing Datasets

Several datasets now exist for testing general predicate paraphrase and entailment, but we argue that the most important consideration when modifying Entailment Graph predictions is maintaining the capability for strong directional inference. 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. Making these inferences is difficult, but crucial for nuanced language understanding. Therefore, we demonstrate our smoothing method on two fully directional datasets, which test both directions of these kinds of inferences, creating a 50% positive/50% negative class balance.

Levy/Holt Dataset. The Levy/Holt dataset has been explored thoroughly in previous work (Hosseini, 2021; Guillou et al., 2021; Li et al., 2022b; Chen et al., 2022). This dataset has the distinction of including inverses for all items, allowing systematic investigation of directionality, although it contains a high proportion of reversible entailments (paraphrases) and selection bias artifacts that can be picked up by fine tuning in supervised models, due to its construction method. We focus on the 1,784 questions forming the purely directional subset, which is more challenging.

ANT Dataset. ANT[1] is a new, high-quality dataset improving on Levy/Holt, which tests predicate entailment in the general domain. It was created by expert annotation of entailment relations between predicate clusters, 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 purely directional subset of 2,930 questions.

See Table 2 for dataset examples. Each dataset comes preprocessed to identify argument types using CoreNLP (Manning et al., 2014; Finkel et al., 2005) which roughly align with the EG's FIGER types. Typed relations are then extracted by the MoNTEE system (Bijl de Vroe et al., 2021), which are used as queries to our models.

[1]To be released soon in a separate paper.

Typed Predicate	Constructed Sentence
`(join.1,join.2)#person#organization`	"person **join** organization"
`(give.2,give.to.2)#medicine#person`	"**give** medicine **to** person"
`(export.1,export.to.2)#location_1#location_2`	"location_1 **export to** location_2"

Table 1: For an input typed predicate x, $L(x)$ constructs a pseudo-sentence and encodes it with a Language Model. The output representation is the average of the sentence vectors corresponding to the **predicate**.

"The audience applauded the comedian" $\models$ "The audience observed the comedian"
"Apple supported Samsung" $\models$ "Apple had an opinion on Samsung"
"The laptop was assessed against the criteria" $\not\models$ "The laptop satisfied the criteria"

Table 2: Example queries from the (development) directional subset of ANT.

3.3 Experiments with P and H smoothing

We experiment by smoothing two recent Entailment Graphs: the graph of Hosseini et al. (2018) (we refer to this model as **GBL** for short) and the state-of-the-art graph in Hosseini et al. (2021) (**CTX** for short). Importantly, these graphs are constructed from the same set of predicate vertices, but CTX improves upon the number of learned edges over GBL. GBL introduces a global edge-learning step after local learning, and CTX later improves on the local edge-learning step using a contextual link-prediction objective, then also globalizes. Both have previously scored highly amongst unsupervised models on the full Levy/Holt dataset.

We run two experiments on each dataset. (1) We apply our unsupervised smoothing method to augment the *premise* of each test entailment relation, generating K new target premises for each relation. Separately, (2) we smooth the *hypothesis* of each test relation the same way. For both we try different values of the hyperparameter $K \in \{2, 3, 4\}$.

3.4 Results

Plots for model performances are shown in Figure 2, in which we compare P-smoothing vs. H-smoothing of the CTX graph using the best $K_{premise} = 4$ and $K_{hypothesis} = 2$. In Appendix A 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 area under the precision-recall curve (AUC) and average precision (AP) across the range of recall achieved. A sample of model outputs is given in Table 4.

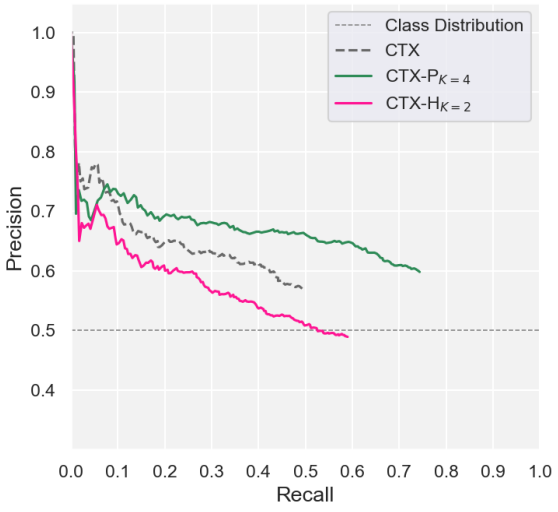

Figure 2: LM smoothing on ANT. Comparison of P(remise) and H(ypothesis) smoothing on the CTX model. We explored $K \in \{2, 3, 4\}$ and show the best $K_{premise} = 4$ and $K_{hypothesis} = 2$.

Our method selecting nearest-neighbors of a target predicate in an EG using their LM embedding distance has very different behavior for smoothing the premise vs. the hypothesis. We observe that P-smoothing is very effective at extending both the recall and precision of both Entailment Graphs it is applied to, with a slight advantage in AUC to higher values of K. When applied to the SOTA model CTX on the ANT dataset, our smoothing method increases maximum recall by 25.1 absolute percentage points to 74.3% while increasing average precision from 66% to 68%. On the Levy/Holt dataset we similarly increase maximum recall by 16.3 absolute pp to 62.7% while exceeding average precision. However, H-smoothing is actually detrimental: despite improving recall, average precision on ANT is severely cut to 59%, with the lowest

Model	ANT		Levy/Holt	
	AUC	AP	AUC	AP
GBL	0.134	0.584	0.158	0.558
GBL-Smooth-P$_{K=4}$	**0.310**	**0.647**	**0.289**	**0.607**
GBL-Smooth-H$_{K=2}$	0.16	0.526	0.173	0.521
CTX	0.324	0.657	0.279	0.602
CTX-Smooth-P$_{K=4}$	**0.501**	**0.675**	**0.381**	**0.608**
CTX-Smooth-H$_{K=2}$	0.345	0.585	0.303	0.580

Table 3: The results of our smoothing method on the premise and hypothesis of inference queries, as compared to unsmoothed models on the ANT and Levy/Holt directional datasets. We report both area under the precision-recall curve (AUC) and average precision (AP) across the recall range.

Predicate Missing from EG	Nearest Neighbors by Embed. 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. Given a target predicate shown as PREDICATE(type1, type2) where PREDICATE may be missing from the EG, we show the top K=2 closest EG predicates in LM embedding space. The missing PREDICATE may appear as either premise or hypothesis.

confidence predictions no better than 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 Entailment Graphs: these techniques improve different aspects of the graph and improvements can be applied together. Since effects are similar for both Entailment Graphs, from now on we show results only for CTX, and report additional results for the weaker GBL in Appendix A.

4 When Is It Helpful to Smooth and How? Analysis with QA

Premise-smoothing with an LM is effective in intrinsic tests, and we now experiment with vertex smoothing in application on a simulated "real-world" task. We use a QA task BoOQA (Li et al., 2022a), in which models must answer true/false questions about entities of interest by reading news articles from multiple sources. Entailment Graphs have proved useful for this task, since they can use directional reasoning to answer questions which are adversarial to simple similarity baselines. When we apply LM-based smoothing to an EG in these tests we again find it beneficial for premise-smoothing and detrimental to hypothesis-smoothing, and also that the amount of context information available to an EG determines the usefulness of premise-smoothing.

BoOQA is, in its original form, presented as an open-QA task, where statements are sampled from the common relations between popular entities, and the validity of each statement is decided according to a huge set of may-be-relevant documents.

Li et al. (2022a) tested the CTX entailment graphs on this task: they extracted open relations from the context documents, selected those concerning the same entities as the statement, and compared each with the statement itself. If any extracted context relation is found to confidently entail the statement, the statement is considered to be supported by context, thus valid.

We use their method as baseline, and apply the same smoothing technique as in §3, either on the statements at question (H-smoothing), or on each co-occurring relation extracted from context (P-smoothing). Surprisingly, we only observe a marginal change in AUC_{norm} values, less than 0.05% [2]. We further find that P-smoothing bears an effect on only 114 of the 58,528 statements; without smoothing, the CTX graphs already have a cut-off recall as high as 89.9%.

This suggests, in open QA, with essentially unbounded context, often we have dozens or hundreds of premises to answer each question. That means, in contrast to predicate entailment datasets where only one premise is available for each hypothesis, for BoOQA it is less hazardous when one premise is missing from the entailment graphs: when the hypothesis statement is valid, chances are there would be other context premises present in the graphs supporting that statement.

In order to verify the above hypothesis, and to

[2] See Li et al. (2022a) for discussion on the AUC_{norm} metric; notably, for the intrinsic experiments above, the AUC_{norm} values are the same as regular AUC.

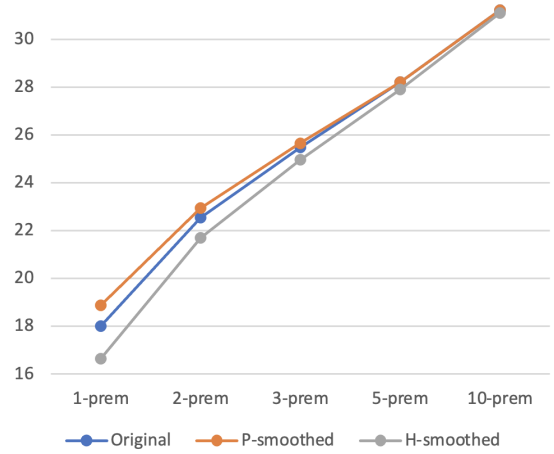

Figure 3: Effect of P-smoothing and H-smoothing when presented with various amounts of context; performance are in % of AUC_{NORM}.

understand the relation between the efficacy of smoothing and the availability of context, we re-evaluate EG smoothing with fixed numbers of context premises per statement. The intuition is, by restricting the amount of context, we push the entailment graphs to make the most of each potential premise; with decreasing amount of context information, P-smoothing should show increasing efficacy compared to unsmoothed models, where H-smoothing should show worsening harm.

In this follow-up experiment, we sample a maximum of K premises per statement to show to the CTX graphs. The graphs then produce a confidence score for each statement, based on the maximum entailment score from the K-sample to the statement itself.

We present the results in Figure 3. Aligned with our expectation, P-smoothing shows the best effect in the 1-premise condition, where H-smoothing brings the worst harm. As the number of available premises increases, performances under both smoothed models converge to that of the unsmoothed models.

From these results, it's shown that smoothing entailment graphs with LMs is most helpful when we have limited context information; on the other hand, when there are abundant context premises as each others' back-up, the entailment graphs themselves would have a good chance of identifying some supporting context and assigning the correct labels to each statement.

5 Discussion: The Asymmetry of LM Embeddings for Smoothing

When used in nearest-neighbor search, LM embeddings perform differently when searching for a premise vs. hypothesis. We attribute this performance difference to a Language Model's fundamental bias toward producing more frequent observations from training corpora, coupled with the natural correlation of frequency with semantic generality in text. Combined, these conditions result in predicted vertices which are semantically more generalized, which is good for P-smoothing, but bad for H-smoothing.

5.1 Language Model Frequency Bias

As statistical learners, Language Models are biased toward high frequency words, since they are trained on a corpus to return the most probable outputs. Frequency bias has been studied in detail: LSTM-based LMs produce a Zipfian frequency distribution of words (Takahashi and Tanaka-Ishii, 2017), and recent models for generation like GPT-2 and XLNet overfit to reporting bias (Shwartz and Choi, 2020). Overproduction of majority cases in training data cause known side-effects with ethical implications, like gender and racial bias (Mehrabi et al., 2021).

Research in Machine Translation has specifically studied this frequency bias as it relates to a semantic generalizing effect from translation input to output (Vanmassenhove et al., 2021). Across neural and phrase-based MT, systems produce translation outputs using words with higher training frequencies, which correlates with quantifiable lower lexical and syntactic richness than their inputs. This generalized output has long been colloquially called "Machine Translationese" due to its artificially non-specific tone.

5.2 Frequency and Generality in Language

Frequency has long been known to correlate with the semantic generality of a word (Caraballo and Charniak, 1999), and this property is used in fundamental algorithms like TF-IDF (Spärck Jones, 1972).

To relate frequency and generality for our purposes, we invoke for illustration 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 conceptualize very general predi-

cate categories at the top of this taxonomy such as "act" and "move," with more concrete subcategories underneath, and highly specific ones at the bottom, like "innoculate" and "perambulate." Rosch et al define a level of "basic level categories" which lie in the middle of their taxonomy, containing everyday concepts like "dog" and "table", which are learned early by humans and are used most commonly amongst all categories, even by adults (Mervis et al., 1976). We assume an analogous basic level in a predicate taxonomy, too, illustrated in Figure 4.

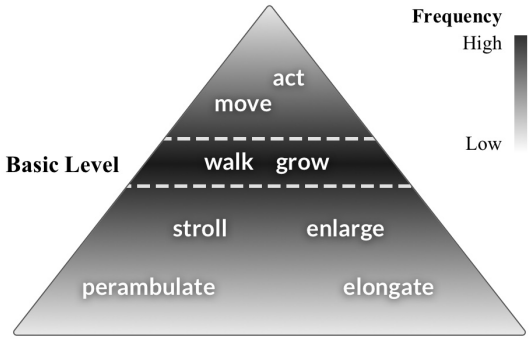

Figure 4: The specificity taxonomy. The basic level contains "everyday" predicates. Those above the basic level become more general, and below become more concrete and specific. Usage frequency decreases moving away from the basic level.

Critically, there are relatively few general categories at the top and very many specific ones at the bottom (consider for example, all the different ways you might "move" such as "walk," "run," "sprint," "circumnavigate"). However, with basic level categories being the most frequently used, moving in either direction in the taxonomy away from the basic level 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. saying "mammal" when discussing a "cat" in Rosch's case or predicates like "actuate" in ours). Below the basic level, predicates are highly specialized and are typically used in specific contexts, so they are both numerous and lower-frequency (e.g. "divebomb," "defenestrate").

This implies that a randomly sampled predicate z is likely to be highly specific as there are very many of them. Fixing z and randomly sampling another predicate z' neighboring z, 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 hierarchy. Thus given z, a frequency-proportional sample z' is likely to be more general than z.

We claim that this applies to Language Models, and that LM embedding space is learned in a way that makes high frequency, generalized predicates easiest to find "nearby" target inputs. When Entailment Graph vertices are embedded in LM space, the neighborhood structure of a predicate is based on similarity, with general, frequent predicates embedded more centrally so that they often appear as a neighbor to the many, more specific predicates. In effect, traversing this neighborhood structure moves *up* the specificity taxonomy.

We now test this claim by demonstrating a theory for vertex smoothing, showing how to smooth the premise and hypothesis by manipulating the specificity of smoothing predictions.

6 Directionality by Transitive Chaining

Applying the same nearest-neighbor search to the premise and hypothesis respectively yields drastically different results, because of a fundamental difference in the *role* of a proposition as a premise or hypothesis. An optimal smoothing algorithm can be formalized as follows for symbolic inference models such as Entailment Graphs, taking into account the role of the proposition we are smoothing by construction of transitive inference chains.

6.1 Constructing a Transitive Chain

We formalize vertex smoothing as a search for optimal replacements. Experiments in §3.3 show that recall may be improved by finding already-learned predicates to approximate missing target predicates. The problem is in maintaining high precision. We start with a target 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 claim that searching for replacement predicates p' and/or h' to build a Q_s suitable for the model must be done as follows:

1. **Generalize P**. Insert a more general premise p' such that $p \vDash p'$, yielding a $Q_s : p' \vDash h$.

$$(Q) \quad \text{“}a \underset{\Pi}{\text{obliterated}}\ b\text{”} \quad \vDash \quad \text{“}a \text{ played } b\text{”}$$

$$(Q_s) \quad \text{“}a \text{ beat } b\text{”} \quad \vDash \quad \text{“}a \text{ played } b\text{”}$$

Relation Category	Entailment Rules		WordNet Demo Relation	Wordnet Demo Example
x entails x'	x $\vDash$ x'	x' $\nvDash$ x	Hypernym	sprint $\Rightarrow$ move
x entailed-by x'	x $\nvDash$ x'	x' $\vDash$ x	Hyponym	play $\Rightarrow$ fumble
x paraphrases x'	x $\vDash$ x'	x' $\vDash$ x	Synonym	assault $\Rightarrow$ attack
x mutually non-entails x'	x $\nvDash$ x'	x' $\nvDash$ x	Antonym	win $\Rightarrow$ lose

Table 5: The four categorical relations $\mathcal{C}$ between a predicate x and its replacement x', defined in terms of entailment, such that $x' \in c(x), c \in \mathcal{C}$. We empirically demonstrate using a WordNet relation $r \subset c$.

2. **Specialize H**. Insert a more specialized hypothesis h' such that $h' \vDash h$, yielding a $Q_s : p \vDash h'$.

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

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

Because both Q and Q_s are test relations they each have unknown truth value. However, we construct Q_s by ensuring that p entails p' and h' entails h, for the purpose of completing a transitive inference chain from p to h. By insertion of p' and/or h' in the intermediary steps of the chain, we can thus leverage confirmation of Q_s to confirm Q.

Case 1. $p \vDash p'$ is known, so if a model confirms $p' \vDash h$, then $p \vDash h$ is confirmed by transitivity.

Case 2. If a model confirms $p \vDash h'$, already knowing $h' \vDash h$ confirms $p \vDash h$ by transitivity.

Case 3. This is a combination of the above. Knowing $p \vDash p'$ and $h' \vDash h$, if a model confirms $p' \vDash h'$, then $p \vDash h$ is confirmed by transitivity.

Restricting the generation of replacement predicates means that a model is not always guaranteed to find a suitable insertion leading to a transitive chain, therefore we cannot expect to attain perfect recall. However, when an additional inference is found this way, it is likely to be correct, aiding model precision.

Alternative smoothing methods which generate a replacement Q_s in a different way (such as with a Language Model) provide no such guarantee of transitivity or correctness. A model will thus generate false positives by mistakenly confirming Q_s when in fact Q is not true, harming overall precision. For instance, if we generalized h instead of specializing it, such that we know $h \vDash h'$ and construct $Q_s : p \vDash h'$. We cannot guarantee entailment between the original p and h, so confirming Q_s does not actually confirm Q.

6.2 Demonstration using WordNet Relations

We now demonstrate these ideas empirically using WordNet (Fellbaum, 1998), a handcrafted resource of English lexical relations such as synonymy and hypernymy. We aim to show that explicitly guiding the search for replacement predicates by constructing transitive chains provides a means for smoothing both premise and hypothesis. For completeness, we explore all possible entailment configurations between a predicate x and its smoothed replacement x'. The four relation categories $\mathcal{C}$ (shown in Table 5) are "entailment," "reverse entailment," "mutual entailment" (paraphrase), and "mutual non-entailment." We test all four categories to demonstrate the theory.

We re-run the experiment of §3.3 by smoothing the CTX (Hosseini et al., 2021) model on the ANT directional dataset (we also test GBL, see appendix). However, in this design the target premise or hypothesis is augmented without using the Language Model. Instead, we generate replacements from each category in $\mathcal{C}$ using WordNet. These entailment categories are broad, so we choose a specific WordNet lexical relation as an instance of each category, then at test-time generate smoothing predictions from the WN database. To illustrate, we choose x *has hypernym* x' as our instance of the "entails" category. At test-time if given a predicate such as "elect," we retrieve WN hypernyms like "choose." Besides *hypernymy*, entailment comprises many relations (often missing from WordNet) like *precondition* including "be a candidate," so enumerating all kinds of entailment for this experiment is not possible. We note that WordNet was used as part of ANT's construction, so this demonstration is meant to explain our model's behavior rather than claim a new dataset score.

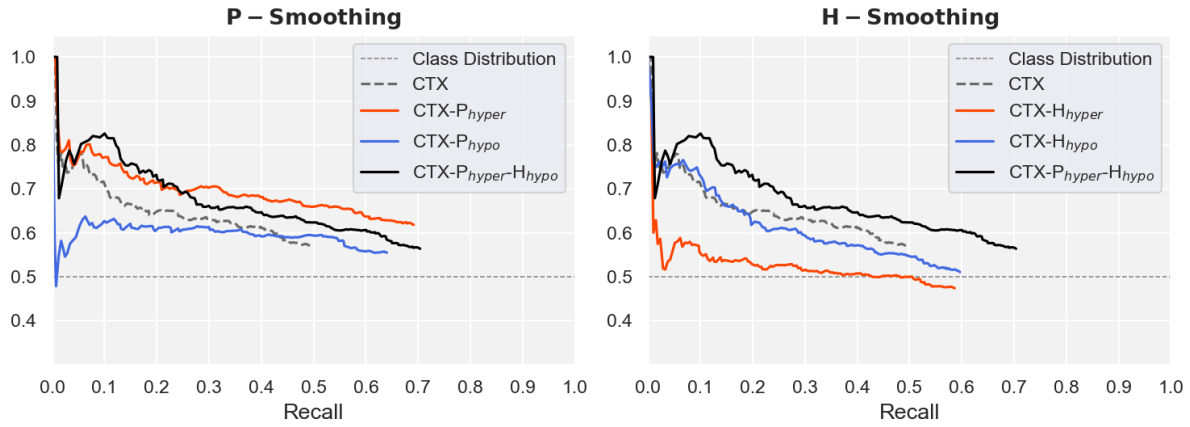

Figure 5: Comparison of WordNet demo relations used in smoothing P(remise), H(ypothesis), and P+H. We compare smoothing effects on the entailment graph CTX (Hosseini et al., 2021). Hypernyms are shown useful for P-smoothing, and hyponyms less so for H-smoothing.

To produce smoothing predictions for a predicate, we query WordNet for the predicate head with desired relation $c \in \mathcal{C}$ and extract all results from the first word sense, then insert each into the predicate. For example, given a target predicate `(receive.2,receive.from.2)` we use the WordNet relation *hyponym*("receive") $\Rightarrow$ "inherit" to form `(inherit.2,inherit.from.2)`. We test all four WordNet demo relations for P-smoothing and separately for H-smoothing in order to compare their effects.

6.3 Results

We show the results of this experiment in Figure 5. In analysis we noted that synonyms and antonyms always performed in between hyponyms and hypernyms (even sometimes outperforming the base EG). As extremes, it is most interesting to focus on hypernymy and hyponymy, so we omit synonyms and antonyms from the plots for clarity.

Importantly, from these plots we note a switch in performance of hypernyms and hyponyms between P- and H-smoothing on the CTX Entailment Graph (similar results for GBL, see appendix). It is clear that generalizing the premise using hypernyms is highly effective in terms of recall and precision, and that specializing the premise 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 it.

These results nearly replicate the behavior of our KNN model experiments discussed earlier in §3.3, verifying that nearest neighbor search in embedding space has a semantically generalizing effect. This result is reflected in Table 4, which shows examples of these generalized predictions.

We note two phenomena of interest. (1) In both models, 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 best of all four H smoothers tested, though it appears unreliable on its own without P smoothing: H_{hypo} is not useful for smoothing CTX (though it does improve the weaker Entailment Graph, GBL, see appendix).

We suggest that both of these phenomena are 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 Entailment Graph with higher quality edge weights. However, highly specialized hyponyms like ELONGATE can be extremely sparse in training data, leading to poorer representations with fewer edges. Phenomenon (1) shows that involving a frequently-occurring smoothed premise of high-quality makes it more likely to find an edge to a smoothed hypothesis, leading to some performance gains over either smoother individually. Phenomenon (2) shows that hypothesis smoothing may itself be more challenging than premise smoothing, and less stable due to relative sparsity of hyponyms (specializations) in corpora. If h is missing from the Entailment Graph (meaning that it wasn't seen in training) then deriving a candidate

h' specialized from h will also be unlikely to occur in training data, thus if found in the EG it may have few or poorly learned edges. Although beneficial in the low-recall setting, differences in data sparsity make hypothesis smoothing fundamentally harder.

7 Conclusions

It is clear from these experiments that smoothing target predicates at inference time calls for guiding the search for replacement predicates differently for premise and hypothesis. P-smoothing must be performed by generalizing, while H-smoothing requires specialization in order to maintain or improve directional precision.

We have shown an unsupervised method for P-smoothing an Entailment Graph using Language Model embeddings, which improves both recall and precision on two difficult directional entailment datasets. We improve over a SOTA Entailment Graph on Levy/Holt (directional) by 16.3 absolute percentage points in recall (to 62.7%), and on ANT (directional) by 25.1 absolute (to 74.3%) in recall, both while exceeding average precision. We have put the method to test on an extrinsic QA task, where we show its benefit in low-resource scenarios where limited context information is available.

Further, we developed a smoothing theory by controlling the search for smoothing predictions for both premise and hypothesis in order to build transitive inference chains, and demonstrated it using gold standard WordNet relations. Our experiments replicated the behavior of the unsupervised LM-based smoother, explaining that LM embeddings are useful for premise smoothing, but not hypothesis smoothing due to a semantic generalizing effect in embedding space neighborhood search.

References

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

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2010. Global learning of focused entailment graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1220–1229, Uppsala, Sweden. Association for Computational Linguistics.

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, 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.

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.

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

Tianyi Li, Mohammad Javad Hosseini, Sabine Weber, and Mark Steedman. 2022a. Language models are poor learners of directional inference.

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.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6).

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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

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.

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*.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

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.

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).*

A Experiments with the GBL Entailment Graph

We show additional testing of the older GBL graph (Hosseini et al., 2018) on the ANT dataset. Results confirm our findings from the newer CTX graph (Hosseini et al., 2021). Figure 6 shows results for the same experimental setup as in §6.2 but smoothing the GBL graph uisng WordNet relations. Figure 7 shows results for the same experimental setup as §3.3 comparing results of P-smoothing using LM predictions for the CTX and GBL graphs.

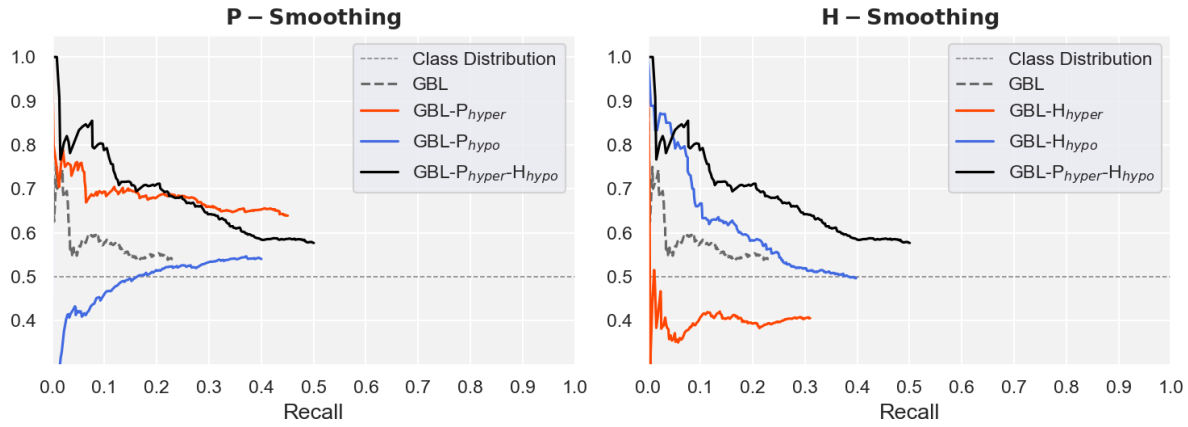

Figure 6: Comparison of WordNet demo relations used in smoothing P(remise), H(ypothesis), and P+H. We compare smoothing effects on the entailment graph GBL (Hosseini et al., 2018). Hypernyms are shown useful for P-smoothing, and hyponyms less so for H-smoothing.

Figure 7: LM smoothing on ANT. Comparison of GBL and CTX models and the P-smoothed versions with optimal K=4.