



THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e. g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

- This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.
- A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.
- This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.
- The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.
- When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Inference of Natural Language Predicates in the Open Domain

Nicholas McKenna



Doctor of Philosophy
Institute for Language, Cognition, and Computation
School of Informatics
University of Edinburgh
2023

Abstract

Inference of predicates in natural language is a common task for humans in everyday scenarios, and thus for natural language processing by machines, such as in question answering. The question *Did Arsenal beat Man United?* can be affirmed by a text *Arsenal obliterated Man United on Saturday* if an inference is drawn that the text predicate *obliterate* entails *beat* in the question. In a world of vast and varied text resources, automatic language inference is necessary for bridging this gap between records and queries.

A promising model of such inference between predicates is an Entailment Graph (EG), a structure of meaning postulates such as x *obliterates* y entails x *defeats* y . EGs are constructed using unsupervised distributional methods over a large corpus, learning representations of natural language predicates contained within. Entailment is directional, and correctly, EGs fail to confirm the opposite, that x *defeats* y entails x *obliterates* y ; these distinctions are important for language understanding applications. In an EG, postulates are typically defined for a predicate argument pair (x, y) over a fixed vocabulary of such binary valence predicates, which relate two arguments.

However, EG meaning postulates are limited in terms of their predicates in two ways. First, using the conventional approach, entailments may only be learned for predicates of the same valence, typically binary to binary entailment, ignoring entailments between valencies and their applications. For example, the binary relation *Arsenal defeats Man United* leads to an inference in humans that *Arsenal is the winner*, a unary relation applying to the subject *Arsenal*. Yet using conventional means, it is not possible to learn these in EGs.

Second, only a limited vocabulary of predicates may be learned in training. This is because of the natural Zipfian frequency distribution of predicates in text corpora, which includes an unbounded long tail of rarely-mentioned predicates like *obliterate*. This distribution simultaneously makes it impractical to learn entailments for every predicate in a language by reading corpora, and also very likely that many of these unlearned predicates may be involved in real queries.

This thesis explores inference in the open domain of natural language predicates beyond a fixed vocabulary of binary predicates. First, Entailment Graph valency is addressed. The distributional learning method is refined to enable learning entailments between predicates of different valencies. This improves recall in question answering by leveraging all available predicates in the reference text to answer questions. Second, the problem of overall predicate sparsity in EGs is explored, in which Language Model

encoding is applied unsupervised with an EG. This provides a means of approximating missing premise predicates at test-time, which improves both recall and precision. However, while approximating missing hypothesis predicates is shown to be possible in principle, it remains a challenge. Finally, a behavioral study is presented on Large Language Models (containing one billion parameters or more) which investigates their ability to perform language inference involving fully open-domain premise and hypothesis predicates. While superficially performant, this class of model is found to merely approximate language inference, utilizing unsound methods to mimic reasoning including memorized training data and proxies learned from corpus distributions, which have no direct relationship with meaning.

Lay Summary

In the Information Age, technologies have developed which record, organize, and synthesize data to assist in our daily lives and industries. The field of Natural Language Processing has emerged in pursuit of an interface between vast data-rich resources and humans, providing a means for us to manage and act on data using simple natural language. One of the important pursuits of the field thus far has been the creation of personal software agents, such as Apple’s Siri or ChatGPT, which in concept make accessing data as easy as speaking to another human being. Such an agent receives a natural language request from a human, manipulates it, and acts on it to synthesize a useful response. For example, one might ask a software agent, *Does Facebook own Instagram?* The agent must process the information request and consider available knowledge resources, such as the news text snippet, *Facebook bought Instagram for \$1 Billion*, in order to produce an answer, *Yes, Facebook owns Instagram*.

However, the process of using a software agent and stored knowledge to automatically answer a question like this must account for a major problem: natural language may express the same information in many different ways, and the stored information may answer the query, but not with an exact string match. Yet, humans are able to make “commonsense” inferences which bridge between expressions, such as that *buying* entails *owning*, enabling seamless person-to-person communication. In fact, this kind of inference isn’t just helpful, it is a necessity for both humans and software agents, since it is highly unlikely that an answer to a question can be found in an exact string match in even a very large knowledge store; the majority of information queries must be answered using language inference.

This thesis explores the topic of language inference in the open domain of natural language predicates, such as *buy*, *own*, *arrive at*, and *be a candidate for*. Command of such inference capability would enable software agents to perform useful tasks such as question answering, text summarization, and information retrieval.

This thesis begins by expanding on a theory for learning such inferences for the construction of an explicit Entailment Graph (EG). An EG is a structure of inference rules for predicates and their arguments, such as *x was elected to y* entails *x was a candidate for y*. They are useful because they can capture the variety of human expressions and are learnable without human annotation, as well as being interpretable by humans and editable after construction. The expansion in this thesis enables novel learning of inferences which involve a variable number of arguments, an essential property of human communication but not possible to learn in previous EGs built in this way. For

example, the question *Is Ted Chiang an author?* can be affirmed by the statement *Ted Chiang wrote “Story of Your Life”* by considering all statements involving *Ted Chiang*, which may involve other entities like his written works. This extension to EGs enables the learning of inferences in a theoretically open domain of natural language predicates, but a practical limit to their application remains. It is often the case that a query to an EG requires an inference rule involving a missing predicate, which was not seen in the original training corpus. This thesis contributes a method of approximating missing predicates and their rules “on demand” in application, by leveraging properties in the encoding process of text by Language Models. Following that, this thesis contributes a behavioral study on the class of Large Language Models alone (without an EG) for their capabilities on this predicate inference task. Though they present a promising impression of “understanding” language, it is shown that they provide only an approximation of language inference, and become unreliable when commonsense queries happen to be unattested by training data, or go against other simple heuristics.

Acknowledgements

This PhD thesis is a culmination of so much effort beyond my own, from many, many people who supported me along the way. My appreciation for them extends far beyond what can be written here.

I especially thank my supervisor Mark Steedman, who took an early interest in me and continues to believe in me with unwavering support. Mark gave me the vocabulary to ask the interesting questions, and the tools to go seek answers. I am a better scientist because of him.

Thank you to Mark Johnson for consistent advice and insight over the years. His contributions were invaluable to many projects, and shaped the way I see the field.

Thank you to Mirella Lapata and Ivan Titov for their thoughtful feedback and critique each year as disparate ideas progressed and matured into one thesis. And thank you to Shay Cohen and Chris Dyer for their dedicated and thorough examination of the final product of this process. Our conversations were deeply interesting and memorable.

Thank you to my coauthors, who I am lucky to count as friends. Their contributions to the work in this thesis were essential, but our time together is far more precious to me. Thank you, Sander Bijl de Vroe, Tianyi Li, Mohammad Javad Hosseini, Liane Guillou, and Liang Cheng.

Thank you to my two dear friends from my time at Amazon, Priyanka Sen and Sandeep Mavadia, who have mentored me and taught me many tricks of the trade. They showed me that research is serious business, but not *too* serious.

Thank you to my labmates, who turned out to be more than just that. They taught me many things, but were also friends and inspirations outside of the lab, too. Thank you to Elizabeth Nielsen, Sabine Weber, Nikita Moghe, Matthew Grenander, Katarzyna Pruś, Tianyang Liu, Thomas Kober, Haixia Man, Louis Mahon, and Miloš Stanojević.

Thank you to my dear friends who I had the pleasure of ~~goofing around with~~ working diligently alongside in the office. Thank you, Asif Khan, Eric Munday, Jesse Sigal, Andreas Grievass, Matthew Di Meglio, Resul Tugay, Lukas Schäfer, and Ibrahim Abu Farha.

A huge thank-you to my wonderful friends in Edinburgh, who had nothing to do with this thesis but everything to do with making me the person that I am. Thank you, Giulia Morrone, Tobias Seger, Thomas Sterling, Evan Loudon, Julie Næss Karlsen, Preben Rostgaard Kise, Katherine Ross Stewart, Joshua Wood, Alexandra Nash, Elissa

Webb, John Tranter, Aryan Kaveh, Grace Bailey, Maja Rimer, Morgan Wilson, and Stefan Baldacchino.

Another huge thank-you to my dear friends around the world, who do the same, but remotely via Zoom (that's a COVID joke). Thank you immensely for our time together, Benjamin Levin, Leonie Shulze, Roma Patel, Surbhi Madan, Quinn Li O'Shea, Sarah McNeil, Sarah Blunt, Hanna Chipman, Melisa Chuong, Vishalenee Thamboo, Thomas Bain, and Ho-Chun Herbert Chang.

Thank you to my family. They raised me, nourished my curiosity, and encouraged me to make big bets on myself. Thank you, Mom, Dad, Thea and Reiley, John, Rosemary and James, Lori and Mark, Brett and Sophia, Jenna and Dom, Karen and Will, Maureen and Larry.

Lastly, thank you to my loving partner Caoilinn. His support and impact on me cannot be overstated. But simply, he turns my rainy days into sunny ones, able to make me smile even in my toughest moments. His brilliance inspires me every day.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Nicholas McKenna)

Table of Contents

1	Introduction	1
1.1	Text, a.k.a. <i>Vast, Unusable Knowledge</i>	1
1.2	Thesis Statement	4
1.3	Thesis Outline and Contributions	4
2	Background	7
2.1	Form-Independent Semantics and Textual Entailment	7
2.2	The General Entailment Task	10
2.2.1	Linguistic Principles and Shotgun Coverage	10
2.2.2	Lexical Relation Resources	12
2.2.3	Lexical Inference in Context	13
2.3	The Directional Entailment Task, Its Difficulty, and Proxies	14
2.3.1	Levy/Holt	14
2.3.2	ANT	15
2.3.3	Artifacts and Heuristics	17
2.4	The Distributional Inclusion Hypothesis	18
2.5	Entailment Graphs	20
2.5.1	Unsupervised Construction from Parsed Text	20
2.5.2	The Problem of Edge Sparsity	23
2.5.3	The Problem of Vertex Sparsity	24
2.6	Language Models Applied for Entailment	25
3	Multivalent Entailment Graphs	29
3.1	Introduction	29
3.2	Background	30
3.3	Multivalent Distributional Inclusion Hypothesis	31
3.4	Methods: Learning Multivalent Entailment Graphs	34

3.4.1	Extraction of Predicate Relations	36
3.4.2	Learning Local Graphs	37
3.4.3	Learning Global Graphs	39
3.5	Methods: Constructing a Natural Multivalent QA Task	40
3.5.1	Question Generation	41
3.5.2	Question Answering Models	42
3.6	Experiment 1: All Questions from Text	45
3.6.1	Results	45
3.7	Experiment 2: Questions Within-Distribution	46
3.7.1	Results	46
3.8	Error Analysis	48
3.9	Conclusion	48
3.9.1	Limitations	49
3.9.2	Ethical Considerations	50
4	Smoothing Entailment Graphs with Language Models	51
4.1	Introduction	52
4.2	Background	53
4.3	Theory of Smoothing	55
4.3.1	Directionality by Transitive Chaining	55
4.3.2	LM Embeddings and Specificity	57
4.3.3	The Specificity Taxonomy	58
4.4	Methods: Approximating Missing EG Predicates	59
4.4.1	Nearest Neighbors Search	60
4.4.2	Datasets	61
4.4.3	Models	62
4.5	Experiment 1: Entailment Detection	63
4.5.1	Results	63
4.6	Experiment 2: Boolean Question Answering	65
4.6.1	Boolean Open QA Dataset	65
4.6.2	QA in the Natural Distribution of Contexts	67
4.6.3	Results	68
4.7	Experiment 3: Controlled Smoothing of P and H with WordNet Relations	69
4.7.1	Controlled Search with WordNet	69
4.7.2	Results	69

4.7.3	Discussion	71
4.8	Related Work	72
4.9	Conclusion	73
4.9.1	Limitations	73
4.9.2	Ethical Considerations	74
5	Large Language Models and Open-Domain Predicate Inference	75
5.1	Introduction	76
5.2	Background	78
5.3	Experimental Design	78
5.3.1	Two Biases in Inference Predictions	78
5.3.2	Datasets	81
5.3.3	Dataset Transformations	82
5.4	Methods: Querying Models with Prompts	84
5.4.1	Model Selection	84
5.4.2	Prompt Design	85
5.4.3	Scoring Model Output	91
5.5	Experiment 1: Attestation Bias	91
5.5.1	Results	92
5.5.2	Implications for Real Applications	93
5.6	Experiment 2: Entities are Indices to Memory	94
5.6.1	Results on Levy/Holt	94
5.6.2	Results on RTE-1	95
5.6.3	Instructing LLMs to Ignore Attestation	97
5.7	Experiment 3: Relative Frequency Bias	98
5.7.1	Results	99
5.8	Impact of Bias on Performance	100
5.8.1	Results	101
5.9	Conclusion	102
5.9.1	Limitations	103
5.9.2	Ethical Considerations	103
6	Conclusion	105
6.1	Summary of Findings	105
6.1.1	Multivalent Entailment Graphs	106
6.1.2	Combining Entailment Graphs and Language Models	107

6.1.3	Entailment by Large Language Models Alone	108
6.2	Directions for Future Work	109
6.2.1	Discovering Metarelations in Entailment Graphs	109
6.2.2	Improvement in Entailment Graph Coverage	109
6.2.3	Improvement in LLM Training Objectives	110
Bibliography		111

Chapter 1

Introduction

1.1 Text, a.k.a. *Vast, Unusable Knowledge*

The United States Library of Congress, the world’s largest library, recorded in 2021 that it possessed within its collection 118.6 million books, manuscripts, and other written works ([The Library of Congress, 2023](#)). It takes an impressive operation in order to maintain this collection and service information requests, costing nearly one billion dollars per year. Yet, personally searching through the wealth of knowledge in all this text, even using a system which indexes by volume, still takes dedicated time. Digital technologies like Google search have been developed to scan and index text for us, in order to speed up the answering of queries, for example, *Who played Captain Jean-Luc Picard?* Google considers all available text on the web in order to search for an answer in a tiny fraction of the time it would take to search a physical library. And yet, extremely fast search may still not be enough to actually answer the question.

Text is one of the most plentiful mediums of information we have, yet it confronts us with a severe problem of our own making: as humans, we’re too good at conveying information in multiple different ways. In fact, it is very often the case that a question cannot be answered by scanning for an exact string match in a large corpus of text, but it *can* be answered using the same corpus by drawing a simple natural language inference. For example, if Google is able to find a similar sentence such as *Patrick Stewart performed the role of Captain Jean-Luc Picard*, then a human may infer the answer is *Patrick Stewart*, by understanding that *x performed the role of y* entails that *x played y*. It is this inferential leap which is necessary to bridge the gap between vast text resources and queries; without such inference by human or machine, text is startlingly unusable for queries. This is a central concern of this thesis, which focuses

on the inference of textual predicates. Throughout, a **predicate** is defined as a descriptive word or phrase which applies to one or more arguments, such as a verb (x left y), verb phrase (x blow up y), or copula construction assigning a property (x is president).

Unfortunately, using a simple measure like the similarity of two statements to judge if they are paraphrastic is not enough for a software agent to be able to provide the answer itself. Unlike paraphrase, **directional predicate inference** holds in one direction, but not both, and they are commonly encountered when matching a statement to a query. Thus, it is risky to match statements using only paraphrase. For example, consider the following scenario:

TEXT SNIPPET: *Rishi Sunak was elected PM of the UK.*

QUESTION: *Who stood for PM of the UK?*

In this case, the answer *Rishi Sunak* may be inferred from the text snippet. However, now consider the opposite case where the propositions are reversed:

TEXT SNIPPET: *Rishi Sunak stood for PM of the UK.*

QUESTION: *Who was elected PM of the UK?*

In this case, a human reader and an ideal software agent would not infer that Sunak is the answer (indeed, before his win, Rishi Sunak stood for PM in the July-September 2023 election to replace Boris Johnson, and lost). Though very similar in usage, *stand for Prime Minister* does not entail *elected Prime Minister*. In the context of inference as in this example, a **premise** statement (the text snippet) is compared with a query **hypothesis** statement (the question), and it is up to a human or machine to determine if the premise entails the hypothesis.

Further, asymmetry in information content between a premise and hypothesis statement can take another form. **Predicate valency** refers to the number and types of arguments related by a predicate, such as binary $\langle 2 \rangle$ (x greets y) or unary $\langle 1 \rangle$ (y dies). The above *Sunak* examples consider only binary predicates which relate exactly two arguments. But natural language is more flexible than this, and predicates may relate other numbers of arguments, for example *Sunak speaks* $\langle 1 \rangle$ and *Sunak gave Hunt a position* $\langle 3 \rangle$. Human intuition easily extracts facets of information from higher-valency relations, such as that *Sunak was elected PM* $\langle 2 \rangle$ entails that *Sunak was elected* $\langle 1 \rangle$ and also that *Sunak won* $\langle 1 \rangle$.

Research in the process of learning inferences between natural language predicates has resulted in the *Distributional Inclusion Hypothesis* (DIH), which has been demonstrated in practice to learn directional entailments for a subset of predicates. This

theory states that for predicates p and q , p is said to entail q if the set of contexts that p appears in is included in the set of contexts that q appears in. For ease of computation, “context” has been operationalized to mean “argument pairs,” read in text with binary (2-argument) predicates as mentioned above; p is said to entail q if all argument pairs observed with p are also seen with q . For example, a machine may read about many people standing for a certain office, but only one of those people will be observed winning that same office. After reading about many such contests where one and only one candidate is observed winning, the machine may learn via the subset relation that x win y entails x stand for y , but not vice versa.

Entailment Graphs (EGs) are a family of unsupervised algorithms which typically implement the DIH for learning the set of rules of this form, using nothing more than a large text corpus, syntactic parser, and named entity linker. The DIH is useful because it theoretically enables learning entailments for any predicate mentioned in the training text, given enough textual mentions. The rules are formulated as graphs, composed of vertices representing binary natural language predicates like x *elected to* y . The graph’s directed edges represent entailment relations between predicates, such as x *elected to* y entails x *stand for* y . EGs are demonstrated to learn quality entailments, but they face an important challenge, which is that they can only *practically* discover entailments about a limited subset of predicates within the open domain of natural language. Though promising, two main obstacles prevent Entailment Graphs from performing inference in the open domain of natural language predicates:

(1) Entailment Graphs have previously only been learned for predicates of the same valency using the DIH. Usually, they are learned with binary predicates, containing edges from one binary (2-argument) predicate to another, though unary-to-unary (1-argument) predicate inference has also been demonstrated to an extent. But no Entailment Graph has been constructed based on the DIH which is learned with edges *between* predicates of different valencies, which would enable the use of any available inference in real tasks, such as the use of a binary predicate relation to answer a unary predicate question.

(2) Entailment Graphs are learned with a fixed vocabulary of predicate *symbols* observed in training, and cannot generalize to novel predicate symbols which may occur at test-time. This is often a problem since an edge will be unlearnable if either predicate involved in the inference was not seen in training. Further, it is impractical to simply scale up the same Entailment Graph learning process on a larger corpus of text to solve this problem of *vertex sparsity*, since predicates occur in corpora in a Zipfian

frequency distribution with an unbounded long tail. There will virtually always be predicates at test-time which were not observed in training.

1.2 Thesis Statement

This thesis explores inference in the open domain of natural language predicates beyond a fixed vocabulary of binary valence predicates. The Entailment Graph learning method is refined to enable learning entailments between predicates of different valencies, and general predicate sparsity in EGs is addressed to enable generalizing inference beyond the predicates seen in training. First, the *Multivalent Distributional Inclusion Hypothesis* is presented, which applies to linguistic eventualities, rather than the typical textual strings relating a fixed number of arguments. This theory enables the learning of edges in Entailment Graphs between predicates of different valencies by respecting the roles of arguments across eventualities. Second, the problem of vertex sparsity in Entailment Graphs is considered, in which a new technique called *graph smoothing* is demonstrated using an unsupervised Language Model to approximate missing predicates. While this provides a means for approximating missing premises, which improves both recall and precision, the process of approximating missing hypotheses is shown to be possible in principle, but remains a challenge. Finally, a study is presented of the claim that “Large” Language Models (one billion+ parameters) are capable of natural language inference involving fully open-domain premises and hypotheses. In contrast to explicit linguistic reasoning by applying an Entailment Graph, this class of model presents a useful *approximation* of linguistic reasoning, but is demonstrated to use unsound methods to achieve its performance. This includes the memorization of training data and use of biases learned from corpus distributions, which have no direct relationship with meaning.

1.3 Thesis Outline and Contributions

This thesis is structured as follows:

Chapter 2 Terminology and background information are presented for the material discussed throughout this thesis. The background ranges over topics including textual entailment; datasets for evaluation of entailment; a theory of textual entailment detection preceding methodologies, the *Distributional Inclusion Hypothesis*; and back-

ground on relevant past methodologies for entailment such as Entailment Graphs and Language Models.

Chapter 3 The problem of Entailment Graphs being constructed for a single valency is studied with an aim to learn entailments between natural language predicates of different valencies, e.g. *person x defeats person y* entails that *person x wins* and *person y loses*. The Distributional Inclusion Hypothesis is refined to track linguistic eventualities rather than textual strings, which enables *multivalent* entailment learning in Entailment Graphs by respecting the roles of arguments in eventualities, which may have varying numbers of arguments.

The lack of general resources for evaluating directional predicate inference is also addressed, and a novel method is presented for the automatic construction of an evaluation task, corpus-based Boolean Question Answering. Models must affirm or deny automatically generated questions by drawing inferences from a selection of real news text, which requires fine-grained semantic distinction. The benefit of Multivalent Entailment Graphs is shown in their ability to answer more questions than a typical EG by drawing on textual evidence consisting of predicate relations with multiple valencies, as well as improved precision over non-directional similarity baselines.

This chapter is based on [McKenna et al. \(2021\)](#), published in the Conference on Empirical Methods in Natural Language Processing (EMNLP). This work was completed with several coauthors: Liane Guillou, Mohammad Javad Hosseini, Sander Bijl de Vroe, Mark Johnson, and Mark Steedman.

Chapter 4 Next, the further problem of predicate sparsity due to symbolic learning in Entailment Graphs is explored. First, a theory is presented for *smoothing* EGs by approximating missing predicates using existing predicates, while maintaining directional precision in inference by choosing approximations which construct a transitive chain. An unsupervised smoothing method is then presented in which predicates are encoded into vectors with a sub-symbolic Language Model, and these are used in a nearest-neighbors search for approximations of missing predicates. This method enables inference with an open domain of *premise* predicates, even if querying out-of-vocabulary predicates, improving recall while maintaining strong directional precision. However, premise- and hypothesis-smoothing are shown to be fundamentally different operations with their own challenges. The same method cannot be applied for missing hypotheses, but hypothesis-smoothing is shown to be possible in principle following

the transitive chain theory by using a manually annotated resource such as WordNet, which is expansive, but still closed-domain.

This chapter is based on [McKenna et al. \(2023b\)](#), published in the Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (AACL), where it received the “Best Paper Award.” This work was completed with several coauthors: Tianyi Li, Mark Johnson, and Mark Steedman.

Chapter 5 Due to the limitations of Entailment Graphs and the utility of Language Models shown in the previous chapter, this chapter turns toward the Large Language Model as a system of textual entailment in its own right, without using an Entailment Graph. Language Models induce sub-symbolic encodings of both premises and hypotheses, enabling input from a completely open domain of language, and claims have been made about their capability for natural language inference. This chapter presents a series of behavioral studies using directional datasets and strong controls to probe this capability of directional inference in state of the art Large Language Models. Though superficially promising, they are shown to rely on several biases in inference decisions, including propositional memories learned in training, which are tied to specific entity IDs, and frequency effects analogous to those explored in the previous chapter. It is concluded that although they present a useful approximation of directional inference of open-domain predicates, Large Language Models use unsound methods which raise questions about generalization.

This chapter is based on [McKenna et al. \(2023a\)](#), published in the Findings of the Conference on Empirical Methods in Natural Language Processing (Findings of EMNLP). This work was completed with equal contribution from Tianyi Li (co-first authors), and with several other coauthors: Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman.

Chapter 6 Finally, this thesis concludes with a summary of findings and directions for future work.

Chapter 2

Background

This chapter introduces material which forms the basis of the contributions in this thesis. It begins with terminology and concepts useful for understanding the research area, and then a brief overview of the historical context is given which describes the development of models and evaluations in entailment, motivating the work of this thesis. The following chapters build on this material by contributing novel theories, experiments, and conclusions.

2.1 Form-Independent Semantics and Textual Entailment

The phenomenon introduced in Chapter 1, the gap between available text resources and queries, is a problem due to the lack of a form-independent semantic representation of natural language. The variability of natural language allows multiple surface forms to express the same meaning (such as interchanging *buy* and *purchase* in the same sentence), and it is up to the reader (human or machine) to make inferences between a text and a natural language query using a semantic representation which bridges between superficial forms.

Semantic parsing of natural language is a relevant technique which normalizes sentential content, including predicates, into a standard logical form suitable for computer manipulation such as logical deduction. However, it does not fully solve this problem of semantic representation because it cannot unify lexical relations, such as recognizing that the parsed logical propositions `BUY(John, apple)` and `PURCHASE(John, apple)` are paraphrases of the same event. Further, directional inferences are a more difficult challenge, because they require information beyond the mutual closeness of two predicates. For example, inferring that `BUY(John, apple)` entails `OWN(John, apple)`, but not

in the other direction: $\text{OWN}(\text{John}, \text{apple})$ does not necessarily entail $\text{BUY}(\text{John}, \text{apple})$.

Formal evaluation of entailment dates to [Tarski \(1935\)](#), with model-theoretic truth. In this formalism, a query statement is evaluated in the context of a structure of axiomatic statements (a model). By deduction it is determined if the query is true under the model, or not. Modern computational approaches to entailment must reckon with a broad variety of language and possible inferences in real use with human-generated text and queries, which is made possible using a softer, probabilistic interpretation of entailment detection. In NLP, the task of *Recognizing Textual Entailment* (RTE) ([Dagan et al., 2006](#)), also known as *Natural Language Inference* (NLI), requires a model to predict an inferential relationship between a text T and hypothesis H : “ T entails H if, typically, a human reading T would infer that H is most likely true,” and this is the working definition used throughout this thesis. Though broad, this definition—grounded in human intuition—has served as a basis for designing evaluation datasets ([Bowman et al., 2015](#)), steering toward a goal of “commonsense” inference capability similar to that of a human ([Pavlick and Kwiatkowski, 2019](#)). Thus in this sense, entailment approaches, but is less well-defined than logical implication, but this can also be seen as a useful benefit since entailment more broadly captures human commonsense like conversational implicature.

Throughout this thesis, the symbol \models (read as “entails”) denotes an entailment relation between two natural language statements or statements parsed into logical formulae: *premise* \models *hypothesis*. An entailment relation has an inherent binary truth-value depending on whether the hypothesis may actually be inferred from the premise, or not (where an explicit $\not\models$ may be used). Entailments may carry from many aspects of sentences, such as the lexical semantics of nouns (“monkey” \models “animal”) and verbs (“John buys an apple” \models “John owns an apple”); quantification (“all men fish” \models “one man fishes”); syntactic composition (“John and Mary walk” \models “John walks”); etc.

This thesis is concerned with models for the detection of entailments between natural language *predicates*, which includes verbs and other expressions that apply to arguments. These content-bearing words serve as the primitive events and states which describe *things* in the world around us ([Vendler, 1967](#)), and are thus of great interest in modeling language understanding by humans and machines. The truth-value of discussed relations will be decidable on the basis of contained predicates and their attributes, unless otherwise specified.

One of three relational cases may hold between two statements A and B , defined by entailment truth values:

1. **Paraphrase:** In the case that statement A entails B , and also that B mutually entails A , a relationship of paraphrase holds between the two statements. For example, “Google bought YouTube” is a paraphrase of “Google acquired YouTube.”
2. **No Relation:** In the case that neither A entails B , nor B entails A , then the two statements are unrelated by entailment. For example, “Google advertises on YouTube” and “Google acquired YouTube,” though sharing named entities and discussing two business-topical relations, do not entail in either direction, so are considered unrelated by entailment.
3. **Directional Entailment:** In the case that either A entails B , or B entails A , *but not both*, a directional entailment holds between the two statements. For example, “Google acquired YouTube” entails that “Google bid for YouTube,” but not vice versa.

Because paraphrase and unrelatedness are symmetric relations which hold between two predicates, they may be predicted using a crude measure like the similarity between the two statements, which can be estimated using many approaches. One such way is to use corpus distributional statistics, such as collocated context words. In other words, if distributionally, the statement predicates occur with many of the same context words, they may be considered “similar” to a quantifiable degree (according to the Distributional Hypothesis; [Harris 1954](#)). For instance, the predicates *sprint* and *dash* both usually occur with a person or animal, and some specified destination, and may be described using similar words like *speedy*. After calculating the overlap of co-occurring context words in terms of e.g. a percentage, a pre-set numerical threshold may be compared; *sprint* and *dash* should have a high percentage of overlaps, so they are classed as paraphrases. For predicates which do not meet this threshold, they may be considered unrelated. If a task requires language inference, but only to a degree where paraphrase detection is sufficient, then such a simple method can be very effective. However, if a computer model is only capable of distinguishing symmetric relatedness, i.e. paraphrase or no-relation, then on a task requiring directional inference it may make predictions no better than random chance.

Unfortunately, the capability of directional inference is crucial in language understanding in humans. It is often needed in real situations where even a slight information asymmetry between two statements requires directional reasoning. A simple example of this is entailment between predicates of different valencies, e.g. $\text{STAND.FOR}(\text{Rishi Sunak, Prime Minister}) \models \text{BE.CANDIDATE}(\text{Rishi Sunak})$. An entailment holds in this

direction, but cannot hold in the reverse direction, since `BE.CANDIDATE` only applies to Sunak, and it cannot be inferred strictly from this statement which office he is a candidate for. Dropping down in valency such as in this example is no guarantee of entailment, however, as is clear from the counterexample `STAND.FOR(Rishi Sunak, Prime Minister) $\not\models$ BE.WINNER(Rishi Sunak)`. Further, information asymmetries can occur between valencies or within-valency, such as between one binary predicate and another.

Often, detecting a directional entailment between two statements A and B is more difficult than paraphrase, because it immediately requires a more nuanced representation of meaning, wherein switching which statement acts as a premise may result in different valid entailments. Any model demonstrating a capability for directional entailment is also demonstrating a representation of meaning at least *operationally* similar to human intuition.

This thesis addresses broad natural language inference including detection of paraphrase and unrelatedness, but is focused specifically on directional entailment, the most challenging relational case to detect between two natural language predicates, but which is also the most useful, due to the breadth and nuance of natural language expressions which may be inferred.

2.2 The General Entailment Task

Entailment is a broad class of inferences, and it is most easily understood by cataloguing classes of evaluation methods, which characterize the task, rather than individual models. Resources and evaluations may be generally grouped as below, which identify the coverage achieved thus far and the shortcomings in entailment detection.

2.2.1 Linguistic Principles and Shotgun Coverage

The task of language inference began with small, hand-crafted datasets, such as `FraCaS` ([The Fracas Consortium et al., 1996](#)) (340 test samples), a test of widely varied entailments involving logic and linguistic concepts beyond lexical inference of predicates, such as quantification. The PASCAL Recognizing Textual Entailment (RTE) series of challenges followed ([Dagan et al., 2006](#)), starting with RTE-1 (800 test samples), which focus on entailment of various lexical items, syntactic patterns, and other features in naturalistic sentences. The task format is simple, requiring a model to con-

sider one premise statement (as in RTE, see Table 2.1) or several premises (FraCaS) plus one additional hypothesis statement, and infer whether the hypothesis is entailed from the premise(s), with labels True and False.

Label	Premise		Hypothesis
True	Increased storage isn't all Microsoft will be offering its Hotmail users — they can also look forward to free anti-virus protection.	\models	Microsoft will provide free anti-virus protection.
False	Vodafone's share of net new subscribers in Japan has dwindled in recent months.	$\not\models$	There have been many new subscribers to Vodafone in Japan in the past few months.
True	Comdex — once among the world's largest trade shows, the launching pad for new computer and software products, and a Las Vegas fixture for 20 years — has been canceled for this year.	\models	Los Vegas hosted the Comdex trade show for 20 years.

Table 2.1: Dev set examples from the RTE-1 dataset.

However, these early datasets can be very difficult due to the variety of inference required to solve them. In RTE-1, besides lexical entailments, these can range from world knowledge (such as knowing that Berlin is in Germany) to abduction by combining information from multiple clauses and inferring a pragmatic implication (as in the first example of Table 2.1).

The recent success of data-driven neural network approaches creates a need for large, supervised datasets. A new class of datasets and evaluations has emerged for this purpose (Bowman et al., 2015; Williams et al., 2018; Nie et al., 2020). These are created by crowd-annotating many thousands of samples for a task like image captioning (SNLI). However, constructing datasets for a task can result in biases of one kind or another, for example, in SNLI this can lead to relatively simple and definite texts of the form “subject(s) X do task Y” (as might be observed in an image), and do not contain a variety of linguistic features.

There is great variety of linguistic inference in human communication, and the lack of focus in many datasets on a defined subset of inferences leaves many gaps in entailment evaluation. For instance, it is possible for a model to do very well on SNLI or MNLI but have poor competence with negation (Geiger et al., 2020; Luo et al., 2022). Further, Talman and Chatzikyriakidis (2019) show that many neural network models which train on one NLI benchmark may perform well on the respective test portion,

but fail to generalize to other NLI benchmark datasets, even if they have a similar task definition. This is another indicator of over-reliance on dataset idiosyncracies, indicating that what is learned may not be true understanding after all. The Glue/SuperGLUE benchmarks (Wang et al., 2019b,a) aim to alleviate this by aggregating datasets into one benchmark, where models report scores on all subsets; this may help the development of models which generalize to a degree, but the problem of competency gaps remains clear (Bijl de Vroe, 2023).

In many cases, deeper analysis of model performance indicates overfitting to dataset artifacts during training of neural models. Poliak et al. (2018) shows that in many NLI datasets, sample hypotheses can contain cues which are sufficient for predicting correct labels, without the need for models to consider a premise at all. This reveals a deeper difficulty in designing supervised models which cannot “game” the dataset by learning to detect artifacts which are unrelated to entailment, but correlated with correct labels.

In particular, and in relation to this thesis, the above datasets lack focus on lexical entailment (inferences drawn about individual words), specifically the directional inference of predicates, which constitute a very important kind of inference needed across NLP tasks.

2.2.2 Lexical Relation Resources

Human-built resources of lexical relations can be useful for the task of entailment itself, or even evaluation of models. For instance, WordNet Fellbaum (1998) contains relations like hypernymy and hyponymy of nouns, and even has analogous troponym relations for verbs. This is a kind of entailment relation between verbs by adding/removing manner. For example, *stroll* is a more specific kind of *walk* with an added manner of leisure, and it follows that *Mary strolled in the park* entails *Mary walked in the park*. However, lexical entailment is not limited to troponymy and hypernymy, and WordNet has extremely poor annotation coverage of general entailment relations between verbs, such as that *inherit* entails *own*. Other similar projects aim to create human-annotated datasets of lexical relations, such as FrameNet (Baker et al., 1998), in which entries are contextualized in sentences, highlighting the semantic roles of events and their relations, like causal results. Yet manual annotation is expensive, and the more rich the resource, the harder it is to complete.

PPDB is an automatically generated database containing similar relations between lexical items but at a much greater scale (Pavlick et al., 2015). It is constructed using

bilingual pivoting, involving translating from language L_1 to L_2 then back to L_1 , which produces many near-paraphrase candidates. Despite this it, too, faces the challenge of sparseness. Because most generated pairs describe the same event paraphrastically, it lacks most commonsense inferences between two related events, such as that “getting elected” entails first “being a candidate.”

Further, these resources are removed from any sentential context. This is critical, because words may take one of many senses in actual use, and the sense of a word determines its entailments. For example, if *person X writes software Y* then *person X is a programmer*. However a different sense of *write* leads to different entailments, like how *person X writes book Y* entails *person X is an author*.

2.2.3 Lexical Inference in Context

In summary, the previous approaches address general entailment competence, but include evaluation gaps of specific capabilities including predicate inference, or they make it possible to test predicate inference, but are very sparse and removed from natural context.

Another trend in evaluation design, sometimes called “Lexical Inference in Context” (LiC) is a relatively recent class of datasets aiming at exactly this niche. The Levy/Holt dataset (Holt, 2018), discussed later, and similarly constructed SherLlic dataset (Schmitt and Schütze, 2019) are two commonly used benchmarks for predicate entailment evaluation. These have been very useful for evaluating models of predicate entailment (Hosseini, 2021; Schmitt and Schütze, 2021; Chen et al., 2023).

Modeling approaches for these datasets mostly involve Entailment Graphs or fine-tuning and application of a Language Model. However, similar to the supervised neural models in § 2.2.1, Li et al. (2022a) shows that the RoBERTa-based model of (Schmitt and Schütze, 2021) also overfits to hidden artifacts in Levy/Holt, casting some doubt on the viability of learning entailment, specifically of predicates, from supervised examples with Language Models, while avoiding artifacts.

While useful, these datasets contain mostly paraphrase/unrelated relations between predicates, and sometimes contain a specifically directional subset. Directional entailments are the most important to this thesis.

Another gap in these datasets for entailment evaluation is that between predicates of different valencies. This thesis also contributes a method for the automatic generation of a question answering task for the evaluation of entailment models. This method

has the benefit of filling gaps such as multivalent entailment evaluation, while being configurable in the kinds of questions and supporting texts that are selected.

2.3 The Directional Entailment Task, Its Difficulty, and Proxies

Several datasets have been created to test model performance on the task of predicate inference. Two datasets of interest are described here, which are used for evaluating models in this thesis.

2.3.1 Levy/Holt

Levy and Dagan (2016) present an entailment dataset, which was re-annotated via crowd-sourcing by Holt (2018), consisting of 18,407 test samples of the form “given that [premise], is it true that [hypothesis]?” Each sample premise (and hypothesis) is a simple sentence which expresses a (subject, relation, object) triple in natural language. This dataset contains a subset of 1,784 samples which hold in only one direction, but not the reverse, with 892 True samples and their reversals, 892 False samples. Several dev set examples are shown in Table 2.2.

Label	Premise		Hypothesis
True	Ephedrine, is widely used in, Medicine	\models	Ephedrine, is used in, Medicine
False	Ephedrine, is used in, Medicine	$\not\models$	Ephedrine, is widely used in, Medicine
True	Crockett, was killed at, the alamo	\models	Crockett, died at, the alamo
False	Crockett, died at, the alamo	$\not\models$	Crockett, was killed at, the alamo
True	Russia, expanded to, the Pacific	\models	The Pacific, borders, Russia
False	The Pacific, borders, Russia	$\not\models$	Russia, expanded to, the Pacific

Table 2.2: Dev set examples from the Levy/Holt directional subset.

The full dataset has been used to compare the progress of many entailment models (Hosseini, 2021; Guillou et al., 2021; Schmitt and Schütze, 2021; Li et al., 2022b; Chen et al., 2022), and the directional subset is of particular interest in this thesis due to its challenging nature.

2.3.1.1 The Length Artifact

However, as shown in Li et al. (2022a), the Levy/Holt dataset suffers from several artifacts which are exploitable by models, so training or finetuning on this dataset risks learning simple (though often effective) heuristics instead of entailment. One is the “length artifact,” which is the phenomenon that often a sample hypothesis predicate is simply a shortening of the premise by way of eliding detail (even if rephrased in different words). One very simple way to measure this is by comparing the relative lengths of the premise and hypothesis in terms of their number of characters. Indeed, in cases of elided detail, there is often a measurably lesser number of characters in the hypothesis than the premise. In these cases the sample label is naturally True, because a statement entails vaguer generalizations of itself. The label is False in the reverse case, because a semantically general premise cannot guarantee entailment of a specific hypothesis. For example, a Levy/Holt dev sample with True label, *Ephedrine is widely used in medicine* entails *Ephedrine is used in medicine* (and in the reverse case, *is used in* does not necessarily entail *is widely used in*).

Due to the distribution of samples in Levy/Holt skewing toward this artifact, by simply comparing the relative lengths of premise vs. hypothesis predicates a model can make guesses much better than chance instead of learning or demonstrating capability for semantic understanding. This is demonstrable. The directional subset contains a class balance of 50% positive labels and 50% negative, so random-chance guessing respecting this distribution achieves 50.0% precision. This thesis also contributes a performance evaluation of the simple heuristics discussed, shown in Table 2.4. Notably, predictions made on the basis of the relative number of characters in predicates alone achieves 71.1% overall precision, which is a vast overestimation of the skill used.

2.3.2 ANT

Guillou and Bijl de Vroe (2023) present a newer dataset, ANT, in the same format as Levy/Holt, but which attempts to solve the problem of artifacts by construction. ANT contains several subsets including a directional portion of 2,930 samples (1,465 positive and their reverses, the 1,465 negative), and was generated by a two step process: first, expert manual annotation of seed entailment relations, followed by automatic expansion of predicates into predicate clusters using crafted resources such as WordNet Fellbaum (1998). This process instantiates many-to-many comparisons from a single manual labeling. Several examples are shown in Table 2.3.

Label	Premise		Hypothesis
True	Medicine, subdued, the patient	$\not\models$	Medicine, was given to, the patient
False	Medicine, was given to, the patient	$\not\models$	Medicine, subdued, the patient
True	The stockbroker, broadened, the fund	\models	The stockbroker, changed, the fund
False	The stockbroker, changed, the fund	$\not\models$	The stockbroker, broadened, the fund
True	Singapore, permitted, smoking	\models	Singapore, ruled on, smoking
False	Singapore, ruled on, smoking	$\not\models$	Singapore, permitted, smoking

Table 2.3: Dev set examples from the ANT directional subset.

2.3.2.1 The Relative Frequency Artifact

As shown in Table 2.4, the same length heuristic of predicting “entail” if the premise is simply longer than the hypothesis applied to ANT achieves 45.5% precision (below random chance of 50.0%). This makes the dataset actually *adversarial* to the same strategy useful on Levy/Holt, meaning that models optimized for this strategy on the Levy/Holt dataset will fail if directly transferred to ANT. However, a different artifact is detectable within ANT which may be similarly exploited by models.

Similar to the length heuristic, the relative frequencies between two predicates correlates with differences in specificity, where it is expected that infrequent words will be very specific, and frequent words will be more general (having more senses or applicable in more contexts) (Caraballo and Charniak, 1999). This has a strong implication for the direction of entailment between such statements. For example, in the ANT sample *medicine subdued the patient* entails *medicine was given to the patient*, the specific word *subdue* is likely to be much less corpus-frequent than a very general word like *give*. Once again, a specific statement can entail a semantically general one, but the reverse direction cannot hold: a general statement cannot entail a specific one. So, by estimating the relative corpus frequencies of the premise and hypothesis predicates, a model can make predictions about the direction of entailment better than chance, without understanding what the predicates mean.

The predicate frequency heuristic in Table 2.4 is calculated using a simple measure. For a given predicate, the average is taken over the unigram counts of all included words, ignoring 127 stopwords from NLTK (Loper and Bird, 2002) such as “in,” “some,” etc. Ungigram token counts are estimated from the WikiText-103 corpus of popular Wikipedia articles (Merity et al., 2017). The estimated frequency of a premise is then compared to that of the hypothesis, and True is predicted if the premise

is less frequent than the hypothesis: it is assumed that in this case, the premise will be more specific than the hypothesis. Indeed, though the ANT dataset is adversarial to the length heuristic, the relative frequency heuristic achieves precision of 69.2%, well above the random baseline of 50.0%.

2.3.3 Artifacts and Heuristics

The artifacts discussed above make Levy/Holt and ANT vulnerable to simple heuristics. These approaches obtain surprising performance above random chance in precision (on each dataset), and recall (for the length heuristic on Levy/Holt). These results are summarized in Table 2.4.

Levy/Holt (directional)	Precision	Recall
Random Choice	50.0	50.0
Length Heuristic	71.1	67.3
Frequency Heuristic	52.2	27.9
ANT (directional)	Precision	Recall
Random Choice	50.0	50.0
Length Heuristic	45.5	40.2
Frequency Heuristic	69.2	45.9

Table 2.4: Performance of simple heuristics on two directional predicate entailment datasets. Values significantly above random chance are **bolded**. While the ANT dataset fixes the length artifact present in the Levy/Holt dataset, it is vulnerable to a different heuristic based on estimated unigram frequency in common corpora. This comparison highlights the difficulty of producing an artifact-free dataset, but also the potential possibility of doing so. Training on either dataset risks learning the respective bias instead of entailment itself.

Relative frequency as an artifact and heuristic is explored more in Chapters 4 and 5. It is important to note that these are not exhaustive of the kinds of artifacts present in these datasets, and it is possible that there are more which are undetected. It is also possible that there are more kinds of heuristics which may be applied to directional datasets to obtain better-than-random performance without modeling entailment itself. The opacity of neural models, and in particular, modern neural Language Models which contain many billions of parameters, make model introspection much more

difficult. Thus, it is not always obvious which factors are responsible for model outputs, especially in Language Models. In Chapter 5, another heuristic of simple memory recall is introduced, and several strategies are proposed with which to control for known artifacts in evaluation.

There is danger in training supervised models on these datasets, which will detect the simple correlation of artifacts with labels before any possible learning of true entailment. Yet, the prevalence of such artifacts betrays the difficulty of designing an evaluation of directional predicate inference without accidentally including easier correlated tasks. Designing an entailment evaluation may even be described as a task in itself.

Indeed, the problems discussed in this chapter of (a) the scarcity of resources for supervised training of lexical inference and (b) the ease with which artifacts may accidentally enter the few evaluation datasets available, which are correlated with correct labels, bears some significance. Learning entailments may better be accomplished by unsupervised means, wherein contamination by selection artifacts is impossible by construction. This entire thesis is focused on such unsupervised methods and their capability to fully capture predicate inference in the open domain of natural language.

2.4 The Distributional Inclusion Hypothesis

Learning directional entailments in the open domain of natural language is a challenge, because (a) acquiring a single entailment relation between two expressions requires either a sophisticated algorithm or human annotation, and (b) natural language is capable of limitless recombination, and any method for acquiring single entailments must also work at this scale.

The Distributional Inclusion Hypothesis (DIH) is an abstract theory for acquiring directional entailments using unsupervised signals apparent in natural text, given only a textual resource for a target domain and sufficient computational resources for training. It has no theoretical scaling limit on the number of learnable entailments, since they can be learned one at a time, and do not require joint learning. In other words, learning the first entailment requires the same computational cost as the millionth entailment. However, practical limitations do exist, which are addressed later, and throughout this thesis.

The DIH is a refinement of the Distributional Hypothesis (Harris, 1954), and seeks to infer directional relationships between two target words by comparing the typical

contexts in which they both occur (Geffet and Dagan, 2005). The DIH states that for some words p and q , if the contextual features of p are included in those of q , then p entails q . As an abstract theory, the DIH requires concrete interpretation in order to implement it. It has been operationalized in previous work for the learning of predicate entailments by using the observed predicate *arguments* as these contextual features (Kartsaklis and Sadrzadeh, 2016; Hosseini, 2021).

For example, collecting predicate argument pairs seen with both the predicates “elected to” and “candidate for” in some large corpus may yield these observations:

Arguments of “elected to”		Arguments of “candidate for”
(Biden, US President)	—	(Biden, US President)
		(Harris, US President)
(Harris, US Vice President)	—	(Harris, US Vice President)
		(Warren, US President)
(Pelosi, US House)	—	(Pelosi, US House)
		(Mehmet Oz, US Senate)

Table 2.5: The argument pairs observed with *elected to* are included in those seen with *candidate for*, implying that *elected to* entails *candidate for*.

With these argument pairs provided as the contextual features, it can be inferred using the DIH that since the features of *elected for* are a subset of *candidate for*, then *elected for* entails *candidate for*. Since there are several pairs seen with *candidate for* that are not seen with *elected for*, the reverse entailment likely does not hold. These observations can be noisy, but the underlying mechanism is not coincidental; in the larger view of an election cycle, it is intuitive that many persons run in an election, giving them candidacy status, but only one will win election. So it is the case that getting elected implies being a candidate, but being a candidate does not guarantee winning election. The DIH is a powerful signal useful for acquiring directional correlations between predicates, such as this one.

An election is one kind of “episode,” in which events and states are causally related in a typical series (Schank, 1975; Tulving, 1972). In an election, a person announces their run for office, becomes a candidate, and *possibly* gets elected, after which they assume their office. By tracking a particular argument tuple as it appears with predicates across a large corpus of text, one such episode may be aggregated. Collections of similar episodes may be obtained when predicates overlap between them. These col-

lections of episodes will overlap greatly in their predicates, but necessarily have some different inclusions, and may also be noisily reported in the text, as well. But collections of episodes yield a distribution from which a probability can be estimated for the relation of two particular contained eventualities such as *elected to* and *candidate for*, and whether by forwards entailment or backwards.

2.5 Entailment Graphs

The challenge of representing a set of directional entailments, in which a unique natural language predicate may entail a variable number of other natural language predicates, lends itself to a graph structure. **Entailment Graphs** (EGs) have been developed for the purpose of learning, representing, and refining directional entailments between natural language predicates.

In this thesis, a standard Entailment Graph is defined as a directed graph of predicates and their entailments, $G = (V, E)$.

- The vertices V are the set of natural language predicates for which entailments are learned, and predicate arguments have a type from the set \mathcal{T} , containing the 48 FIGER base types (+ 1 type `:thing` used in failure cases) (Ling and Weld, 2012). For example, $\text{FLY.TO}(:\text{person}, :\text{location}) \in V$, and $:\text{person}, :\text{location} \in \mathcal{T}$.
- The directed edges are $E = \{(v_1, v_2) \mid v_1, v_2 \in V \text{ if } v_1 \models v_2\}$, or all learned entailments between vertices in V . For example, $\text{FLY.TO}(:\text{person}, :\text{location}) \models \text{ARRIVE.AT}(:\text{person}, :\text{location})$

2.5.1 Unsupervised Construction from Parsed Text

Entailment Graphs may be constructed using unsupervised means. The basic construction method only requires a syntactic parser for extracting natural language relations between arguments, and a named entity linker to standardize differently-phrased mentions of the same entity reference (otherwise, predicate arguments are unlikely to match up between predicates, even if they refer to the same entity). This can be applied to English, or just as easily to another language with these tools available plus a large enough corpus of text for training (Li et al., 2022b).

The standard construction process begins by applying two steps of data pre-processing in order to extract the concrete predicate-argument triples used to learn an Entailment

Graph with typical binary predicates. Following this is the local learning phase, by applying the DIH to learn initial entailment rules between predicates. This results in a graph structure, which can be used as-is for explicit lexical semantic reasoning, or refined further.

2.5.1.1 Entity Recognition, Linking, and Typing

The first step is to pre-process the natural text corpus by identifying named entities. Past work has used the AIDA-Light system (Nguyen et al., 2014) which links a textual mention to an ID, the entity’s Wikipedia URL, which is useful for standardizing different mentions of the same entity, and may also be used to gather more information about the entity from other sources, such as a typing. Using this, the entity ID is mapped to its Freebase entry (Bollacker et al., 2008), which provides the entity’s FIGER type (Ling and Weld, 2012).

Typing is essential for Entailment Graph learning (Berant et al., 2010; Lewis and Steedman, 2013; Hosseini et al., 2018). Inducing a type for each entity such as “person,” “location,” etc. is useful for disambiguating word sense, e.g. “running a company” (:organization type) has different entailments than “running code” (:software type). This is established in NLP since “One Sense Per Collocation” (Yarowsky, 1993). Typing of entities also enables the aggregation of predicate mentions of the same types, in order to estimate the distributional overlaps required for learning robust entailments.

After identifying named entities and their types, the training corpus is ready for relation extraction.

2.5.1.2 Syntactic Parsing and Relation Extraction

Past work in Entailment Graph construction has made use of Combinatory Categorical Grammar (CCG) parsing (Steedman, 2000), such as the Lewis and Steedman (2014) or later Stanojević and Steedman (2019) parsers, which has benefits for this process such as syntactic analyses, like the graceful handling of conjunctions. The CCG-parsed sentences are then mined for relations. Hosseini et al. (2018) extract relations using the words along the CCG syntactic path between two nominals. Further pre-processing may be done to clean and normalize the relations, such as conversion of passives to actives, lemmatization of predicates, and stripping of tense, aspect, modality, and other auxiliaries.

Examples of this process are shown in Table 2.6. The CCG argument positions

are identified in the predicate, where roughly, 1 is the subject, 2 is the object, and 3 is the indirect object. Finally, the predicate is appended with the mentioned entity argument types, which enable predicates to be disambiguated and quickly sorted by typing. The argument tuple of entities is logged as one instance of this relation. There may be many instances of a particular normalized relation, and the distributions of entity tuples enable learning of entailments in the next step.

Natural Language Sentence	Extracted Typed Relation	Argument Tuple
“Obama flew this week to the Hawaii base.”	(fly.1,fly.to.2)#person#location	(Obama, Hawaii)
“Obama landed in Hawaii and met the general.”	(land.1,land.in.2)#person#location (meet.1,meet.2)#person#thing	(Obama, Hawaii) (Obama, general)

Table 2.6: First, an entity type is induced for each nominal. Then, the sentences are CCG parsed. Finally, typed relations are extracted between nominals, with corresponding argument tuples.

2.5.1.3 Learning with the Distributional Inclusion Hypothesis

Research in Entailment Graphs begun by assembling rules learned immediately from text (Geffet and Dagan, 2005; Szpektor and Dagan, 2008), dubbed “local” learning because individual or “local” graphs are learned for each type-pairing. For instance, the (:person, :location) graph is learned separately from the (:person, :organization) graph, in which entailments are learned between predicates with arguments of these types.

Typically, local Entailment Graphs are learned by implementing a model of the Distributional Inclusion Hypothesis. This is done by comparing the distributions of “context features” for two given predicates, as described in §2.4. The features vary by implementation, as does the scoring function which estimates the strength of directional entailment between two predicates based on the overlap of features. In most implementations, the context features are operationalized as counts of unique tuples of argument entities, or alternatively the positive pointwise mutual information (PPMI) between each unique argument tuple and the predicate. The score is chosen to be Weeds Precision for a purely directional measure (Weeds and Weir, 2003), or BInc for better performance (Szpektor and Dagan, 2008) by down-weighting infrequent pred-

icates, which often align with frequent ones by chance, resulting in spurious entailments. An example Entailment Graph is shown in Figure 2.1.

The incidence of such spurious entailments highlights an important fact. Though the DIH is built upon meaningful intuition, this is an *approximate* process for learning linguistic inference of predicates, which increases in precision over larger distributions in training data, or using other added signals. This process is distinguished from the later use of a constructed EG for explicit linguistic reasoning on tasks.

Recent work has improved the precision of local learning by adding temporal signals in the feature set used for DIH calculations (Guillou et al., 2020) and targeting the use of modalized relations to specific domains (Guillou et al., 2021). Other work has modernized the typical DIH by using Language Model embeddings as the contextual features for better predicate disambiguation (Hosseini et al., 2021).

However, due to methodological simplification, only predicates of the same valency have been considered in previous methods, e.g. binary predicates entail binary, or unary entail unary (Szpektor and Dagan, 2008). This overlooks the learning of crucial entailments that cross valencies, which are easy for humans and necessary for open domain inference of natural language predicates. For example, given the statement *Biden defeated Trump* a human would also understand that *Biden won* and *Trump lost*. This gap is not trivial to fill, however, since lower-valency entailments apply to a particular subset of arguments in the premise, e.g. a unary predicate applying to the subject *vs.* the object of a binary predicate, and a model must learn this additional information. In Chapter 3 the application of the DIH is refined to learn entailments both within and across predicate valencies.

Further, Entailment Graphs are constructed from predicate relationships which are grounded to original observations from real corpora, so they are limited in two crucial ways, as follows.

2.5.2 The Problem of Edge Sparsity

Edge Sparsity is the phenomenon where EGs capture an imperfect subset of “true” entailment relations that may exist between a set of predicates. This is a problem because at test-time, it may be the case that the EG contains both premise and hypothesis predicates, but has no textual evidence for their entailment, so it predicts a false negative. This issue is due to the fact that the DIH, as typically operationalized, tracks occurrences of entity pairs across the training corpus and uses these to connect men-

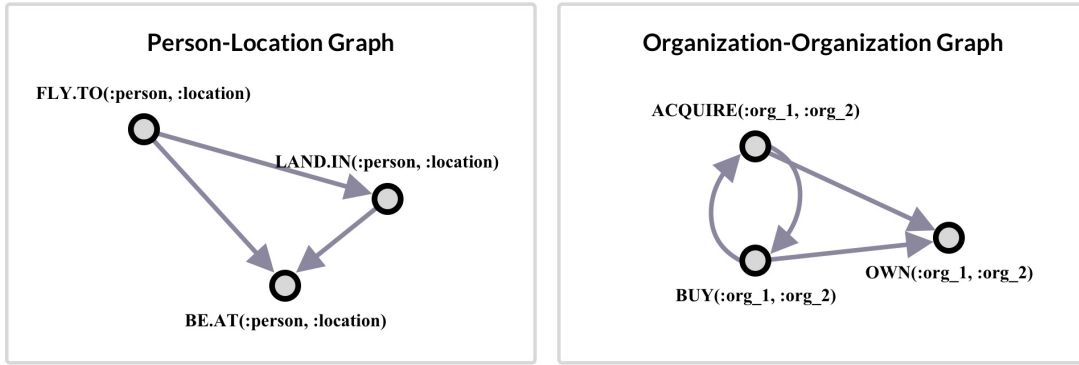


Figure 2.1: An example Entailment Graph containing two typed subgraphs: person-location, and organization-organization. Paraphrases *buy* and *acquire* mutually entail each other, and other entailments are shown which hold in only one direction.

tioned predicates, so edge learning is dependent on observing these predicates with the same entity pairs, which does not always occur due to natural noisiness in reported text.

There is a growing body of research into the problem of Edge Sparsity. After local EGs are constructed, the graphical structure may be leveraged in a process called “globalization,” where graph structures may be shared between local graphs, and also within them (Berant et al., 2010; Hosseini et al., 2018). The transitive property of entailment (if $a \models b$, and $b \models c$, then $a \models c$) is another signal which has been used to infer additional edges in already-learned graphs (Berant et al., 2015; Chen et al., 2022). Further, the complementary nature of Entailment Graphs and Knowledge Graphs constructed from the same corpus may be leveraged for iterative joint-improvement of both EG and KG edges (Hosseini et al., 2019).

2.5.3 The Problem of Vertex Sparsity

Even after applying scaling techniques such as “global soft constraints” (Hosseini et al., 2018) which enable optimization and learning much larger EGs than first possible, the methodology hits another practical limit. EGs are symbolic models, in which are learned representations for a fixed “vocabulary” of predicate symbols observed in the training corpus. However, predicates occur in a Zipfian frequency distribution with an unbounded long tail of rarely-mentioned predicates, so there will virtually always be predicates which appear at test-time which were not seen in training.

This thesis is the first to explicitly recognize the problem of **Vertex Sparsity**. Due to this phenomenon, even if EGs maximize the predicates learned from the training corpus and learn a perfect representation of their entailments, this will still only represent a subset of the possible natural language predicates that will be queried. Thus, the graphs will still be insufficient for use in the open domain of natural language. This is a problem because the long tail of the Zipfian frequency distribution makes it impractical to learn entailments for all possible predicate symbols by reading corpora by this method, yet very likely that many of these unlearned predicates will occur at test-time in corpora for use in real tasks, like question answering from text. Further, at test-time if *either* the premise or hypothesis predicate is missing from the EG, there is no way to learn an entailment involving them, and the EG will predict a false negative. A subsymbolic approach to “smooth” over missing graph predicates by approximating them on-demand is presented in Chapter 4, which seeks to alleviate the problem of vertex sparsity.

2.6 Language Models Applied for Entailment

The recent class of neural Language Models (LMs) tackle the problem of limited vocabulary by a process of encoding subword tokens as vectors and then performing vector space recombination to represent larger word- or sentence-units (Kudo and Richardson, 2018), which enables the encoding of words at test-time which may not have been seen in training. This development solves a problem analogous to vertex sparsity in Entailment Graphs, and enables the encoding and processing of a truly open domain of natural language. Caveats must necessarily be stated that an open encoder does not escape the need for target-domain training data, or the bigger question of whether a meaningful semantics is learned by such models.

Nevertheless, the opportunities presented by embedding techniques and the promising performance of progressively-larger Language Models (Brown et al., 2020; Chowdhery et al., 2022) demands investigation into the possibilities of applying these modern NLP methods to predicate entailment.

Neural Language models with Transformer architectures may typically include an encoder module which produces fixed-dimension vector embeddings corresponding to tokens in the input sequence. These can be used either as-is in downstream applications, or by further finetuning the LM with a small neural module appended (e.g. a multi-layer perceptron) for tasks like classification (Liu et al., 2019). They may also

include a text-generating, auto-regressive decoder module (Raffel et al., 2020), or consist *only* of a decoder module (Radford and Wu, 2019; Brown et al., 2020). The exact architecture is dependent on the target downstream application.

The definition of what is a “small” vs. “large” Language Model has been a moving target in the past several years, because increases in the model parameter count and pre-training data size have consistently achieved better and better results on downstream tasks (Hoffmann et al., 2022), which to many is more appealing than investment in major changes to algorithmic design. By the time you (the reader, hello!) read this thesis, the distinction between small and large may have changed even further. But at the time of writing, it is generally understood that a small neural Language Model (typically a transformer-based model) contains fewer than one billion parameters, which may be arranged in varying stacks of interspersed self-attention and feed-forward layers. A BERT model, for instance, was once considered to be outrageously large, but is actually a “mere” 340 million parameters in its largest configuration, and is now considered to be small (Devlin et al., 2019). A large model is often simply a scaled-up version of the same architecture with additional layers, exceeding one billion or even one trillion parameters (Fedus et al., 2022). These large models are empirically more fluent in text generations than smaller models equipped with decoders, because they have trained on larger and more diverse natural language corpora. Because of this, larger models are usually intended to be used as agents which can converse with a user or even complete tasks if prompted with data formatted as an instruction, as if to another human being (Chung et al., 2022).

While large models achieve impressive performance on a variety of tasks, they require massive resources to train, and even so just for running inference, so they are inaccessible not just to institutions with limited budgets, but also in the kinds of hardware settings in which they may be deployed. For instance, the largest, most performant models are too large to run on current mobile phone hardware. However, small models can train locally on many desktop-class GPUs, and can run inference in real-time, even models deployed to high-end phones. This is possible while still providing many benefits of modern techniques like open-domain text encoding into vectors.

This thesis explores both routes for predicate entailment: the application of small Language Models to assist Entailment Graphs in processing an open domain of natural language predicates in the “personal” hardware setting (Chapter 4), and the use of the most cutting-edge large Language Models as conversational agents, running in the

server setting, which are tested for their ability to “understand” entailment as queried in natural language requests (Chapter 5).

Chapter 3

Multivalent Entailment Graphs

This chapter addresses the restriction of single-valency entailment learning in Entailment Graphs, which is a major barrier to learning an open domain of predicate inferences in natural language. It also addresses the lack of evaluation resources for general predicate entailment models.

First, a theory is presented for distributional learning of entailments between predicates of different valencies, by refining the Distributional Inclusion Hypothesis. This enables learning entailments such as $\text{DEFEAT}(\text{Biden}, \text{Trump}) \models \text{WIN}(\text{Biden})$, which is natural for humans but impossible to capture in previous Entailment Graphs. Further, this theory is actualized by learning unsupervised *Multivalent Entailment Graphs* of open-domain predicates. Finally, the capabilities of these graphs are demonstrated on a novel question answering task. Directional entailment is shown to be more helpful for inference than non-directional similarity on questions of fine-grained semantics. In addition, drawing on evidence across valencies answers more questions than by using only the same valency evidence.

3.1 Introduction

Say that we are reading a murder mystery, and a question comes to mind: *is Mr. Boddy dead?*¹ The passage might say *Colonel Mustard killed Mr. Boddy*, or *Mr. Boddy was murdered in the kitchen with a candlestick*, either of which answers the question, but only via natural language inference.

An *Entailment Graph* (EG) is a structure of meaning postulates supporting these inferences such as “if *x murdered y*, then *x killed y*.” Entailment Graphs contain natu-

¹The murder mystery board game *Clue* (also known as *Cluedo*) lends inspiration to this work.

ral language predicates (represented by vertices) and their entailments (directed edges connecting the vertices). Previous EGs are learned with predicates of a single *valency*, the number of arguments related by the predicate. Commonly, these graphs contain binary predicates of two arguments, and cannot model single-argument predicates like the entity states *x is dead* or *x is an author*. This means they miss a variety of entailments in text that could be useful for answering questions such as *is Mr. Boddy dead?* The Distributional Inclusion Hypothesis (DIH) (Dagan et al., 1999; Kartsaklis and Sadrzadeh, 2016) is a theory which has been used effectively in unsupervised learning of these same-valency entailment graphs, as discussed in Chapter 2 (Geffet and Dagan, 2005), and forms the basis of this work.

This chapter presents 3 contributions:

1. The Multivalent Distributional Inclusion Hypothesis, a refinement of the DIH, is presented, which supports learning entailments between predicates of different valencies such as $\text{KILL}(\text{Mustard}, \text{Boddy}) \models \text{DIE}(\text{Boddy})$ by respecting the roles of arguments in eventualities.
2. A new Multivalent Entailment Graph is developed, where vertices may be predicates of different valencies, which results in new kinds of entailments that answer a broader range of questions including about the individual (unary) properties of entities.
3. Further, a true-false question answering task is posed, which is generated automatically from news text. The Multivalent EG draws inferences across propositions of different valencies to answer more questions than using same-valence entailment graphs. Several baselines are also compared, including unsupervised pretrained language models, and it is shown that directional entailment graphs succeed over non-directional similarity measures in answering questions of fine-grained semantics.

This research is conducted in English, but as an unsupervised algorithm, EG construction may be applied in other languages given a parser and named entity linker (Li et al., 2022b).

3.2 Background

The original task of *recognizing textual entailment* (Dagan et al., 2006) requires models to predict a relation between a text T and hypothesis H ; “ T entails H if, typically, a

human reading T would infer that H is most likely true.” Within RTE, this work is a specific study on the entailment of predicates, including verbs and phrases that apply to arguments.

Research in predicate Entailment Graphs started with “local” learning of entailment rules of the form “if $p(x,y)$, then $q(x,y)$ ” for binary predicates p , q , and entities x , y (Geffet and Dagan, 2005), or “if $r(x)$, then $s(x)$ ” for unary predicates r and s (Szpektor and Dagan, 2008). As discussed in Chapter 2, these methods frequently rely on the DIH for the local learning step to learn initial predicate entailments. The DIH states that for some predicates p and q , if the contextual features of p are included in those of q , then p entails q (Geffet and Dagan, 2005). In previous work predicate arguments are operationalized as these contextual features, but only predicates of the same valency are involved in learning an entailment, e.g. binary predicates entail binary; unary entail unary. However, this leaves out the crucial inferences which cross valencies such as that x kills y entails y is dead, which are easy for humans. Thus, implementations of the DIH thus far cannot be said to model entailment in the open domain of natural language predicates. This work refines the DIH in a way which supports principled learning of entailments within and across valencies.

Later work on joint learning of “globalized” rules leverages the inherent graph structure output by local learning to further improve on the problem of edge sparsity (Berant et al., 2010; Hosseini et al., 2018). These techniques often leverage information such as graph transitivity, so they do not necessarily require linguistic features such as predicates being binary relations, though nearly all work on Entailment Graphs has focused on these binary predicates.

This work compares the Multivalent Entailment Graph to several baselines, including strong pretrained language models in an unsupervised setting using similarity. BERT (Devlin et al., 2019) generates impressive word representations, even unsupervised (Petroni et al., 2019), which is compared with on a task of predicate inference. Further, RoBERTa (Liu et al., 2019) is tested to show the impact of robust in-domain pretraining on the same architecture. These non-directional similarity models provide a strong baseline for evaluating directional Entailment Graphs.

3.3 Multivalent Distributional Inclusion Hypothesis

The Distributional Inclusion Hypothesis is formalized as two statements in Geffet and Dagan (2005), for the word senses v_i and w_j of the words v and w :

Hypothesis I: If $v_i \models w_j$ then all the characteristic (syntactic-based) features of v_i are expected to appear with w_j .

Hypothesis II: If all the characteristic (syntactic-based) features of v_i appear with w_j then we expect that $v_i \models w_j$.

When applied to learn predicate entailments, a “feature” of a predicate is usually operationalized as a tuple of concrete entities related by the predicate that is observed in collocation in a textual corpus (typically, 2-tuples of entites related by binary predicates) (Berant et al., 2010; Hosseini et al., 2018). For example, the tuple (Obama, Hawaii) is a feature of the parsed predicate $\text{ARRIVE.AT}(x,y)$ because it was observed with it in the training corpus. Aligned with this definition, these implementations are designed to search for (potential) matches of premise tuples amongst hypothesis tuples. The confidence of directional entailment between premise p and hypothesis h is estimated based on how many of p ’s tuples are found amongst h ’s tuples. However, this theory (and implementations) require that features (tuples) take the same form between p and h . It does not define entailment if the individual features (tuples) of h are themselves a systematic subset of features of p (these may be called “subtuples”). For example, the DIH does not specify how to compare 2-tuples of the (potential) premise x kills y with 1-tuples of the hypothesis y dies, even though an alignment may be possible if a transformation is applied between premise and hypothesis features.

A new extension of the DIH is posed within the context of predicate entailment, the Multivalent DIH, which models the entailment of predicates both within and across valencies. According to the established DIH, if p entails h , the distribution of p ’s features should form a subset of h ’s features. Essential to this work, in addition, the contextual features of h need only match a systematic transformation on features of p , if h has lower valence than p .

The intuition comes from observing eventualities (Vendler, 1967) which occur in the world. Neo-Davidsonian semantics (Davidson, 1967; Maienborn, 2011) explains that a textual predicate, its arguments, and adjuncts, are all properties of an underlying eventuality, being an event or state. Entailments about one or more of the arguments arise from their roles in this eventuality. In the running example, it may be inferred that *Mr. Boddy died* due to his role as a direct object in the killing event. No other information is needed for a human or ideal learning algorithm to draw this inference, including who murdered Mr. Boddy, where, or with what instrument. Boddy is dead simply because he was murdered. This insight is key to developing the MDIH.

As in earlier EG work, a predicate is represented by features which are the argument tuples it appears with. A tuple is recognized as a proxy for a world event, e.g. `VISIT(Obama, Hawaii)` identifies one instance of a real `VISIT` event. The MDIH acquires entailments by tracking entity tuples across events in the world, recognizing that a lower-valency argument tuple may reference the same eventuality as a higher-valency one, if the entities form a subtuple. The MDIH defines an entailment from a premise p to hypothesis h if, distributionally, subtuples of p are always found amongst tuples of h . Crucially, h is allowed to drop in valency so that entailments may be learned about subsets of p 's arguments.

The MDIH is now formalized and then illustrated with an example. The argument tuple structures for a premise and hypothesis predicate are defined:

$$P = \{(a_{k,1}, \dots, a_{k,I}) \mid k \in \{1, \dots, M\}\}$$

$$H = \{(b_{k,1}, \dots, b_{k,J}) \mid k \in \{1, \dots, N\}\}$$

P is a set of M argument tuples (each of size I) which correspond to instances of a premise predicate p . H is a set of N argument tuples (each of size J) representing the same for hypothesis h . J is limited such that $J \leq I$. This is because entailments are learned for realized entities only, so they cannot be learned from lower to higher valencies (such as a unary entailing a binary). A hypothesis cannot be inferred about real arguments that are not present in the premise; such inferred hypotheses must necessarily contain existential arguments, which are not observable in text. For example, it cannot be learned that “Boddy was murdered” entails “Body was murdered with something” because such kinds of hypothesis statements are virtually never written by rational authors following the Gricean cooperative principle (Grice, 1975), thus they are not observable in text. The special case of linguistic analysis of existentials is left to future work. Finally, for $J = I$, this theory is equivalent to the DIH.

To select argument subtuples from tuples in P , a vector of indices \mathbf{j} is defined with length J , which selects arguments by position. For example, with $\mathbf{j} = [2, 3]$, perform $P[:, \mathbf{j}]$. For each argument tuple in P , select just the 2nd and 3rd arguments, forming a new set of 2-tuples. The Multivalent Distributional Inclusion Hypothesis is defined:

$$\text{If } P[:, \mathbf{j}] \subseteq H[:, m(\mathbf{j})], \text{ then } p \models h$$

Here, $m : \mathbb{N}^J \rightarrow \mathbb{N}^J$ is a simple bijective mapping from argument indices of p to h . For example, m is needed for argument mapping in “ x bought y for z ” entails “ y sold to x .”

The kill/die example is now illustrated on a hypothetical corpus. It might be found that $\text{KILL}(x, y) \models \text{DIE}(y)$ by trying $\mathbf{j} = [2]$ and $m([2]) = [1]$. Starting with P , all 2-tuples

of *killings*, and H , all 1-tuples of *dyings* and apply \mathbf{j} and m . It may be found that selecting arg 2 from all tuples in P forms a subset of the selection of arg 1 from tuples in H . Though *dyings* may happen in many ways, it may be observed that arg 2 of a *killings* often occurs elsewhere in the corpus with a *dying*, and thus the entailment between predicates can be inferred. Intuitively this is true for arbitrarily large valencies: MURDER(Mustard, Boddy, kitchen, candlestick) entails KILL(Mustard, Boddy) and both entail DIE(Boddy).

Though premise arguments may be dropped in the hypothesis, they still influence entailments. This is because the MDIH tracks the underlying *eventualities*. “Person writing a book” is a different kind of event than “person writing software” with a different distribution of argument tuples, so it may be learned that the former entails “person is an author” while the latter entails “person is a programmer.”

3.4 Methods: Learning Multivalent Entailment Graphs

The theory is demonstrated by learning Multivalent Entailment Graphs which contain entailments between predicates of 1- and 2-valency.

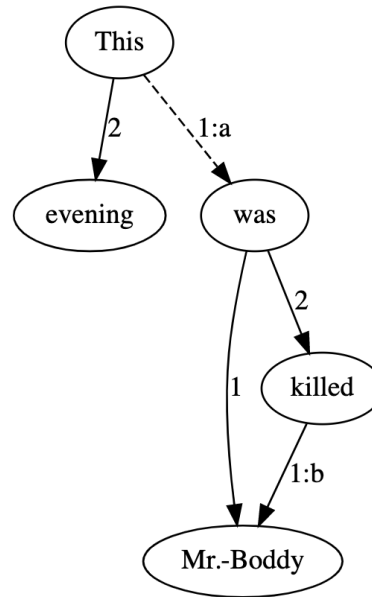
An Entailment Graph is defined as a directed graph of predicates and their entailments, $G = (V, E)$. The vertices V are the set of predicates, where each argument has a type from the set of 49 FIGER base types \mathcal{T} , e.g. $\text{TRAVEL.TO}(\text{:person}, \text{:location}) \in V$, and $\text{:person}, \text{:location} \in \mathcal{T}$. The directed edges are $E = \{(v_1, v_2) \mid v_1, v_2 \in V \text{ if } v_1 \models v_2\}$, or all entailments between vertices in V .

In Multivalent Entailment Graphs V is expanded to contain predicates of both 1- and 2-valency, and E to edges between these vertices, described as follows. Let $b_i, b_j \in V$ be distinct binary predicates and $u_i, u_j \in V$ be distinct unary predicates. Define \mathcal{E} as the set of all entities in the world, and some particular entities $x, y \in \mathcal{E}$ to illustrate argument slots. E contains these patterns of entailment:

1. $b_i(x, y) \models b_j(x, y)$ or $b_i(x, y) \models b_j(y, x)$
Binary entails binary ($\mathbf{B} \rightarrow \mathbf{B}$ entailments)
2. $b_i(x, y) \models u_i(x)$ or $b_i(x, y) \models u_i(y)$
Binary entails unary of one argument ($\mathbf{B} \rightarrow \mathbf{U}$ entailments)
3. $u_i(x) \models u_j(x)$
Unary entails unary ($\mathbf{U} \rightarrow \mathbf{U}$ entailments)

Predicates with valence > 2 are sparse in the text, but are also included in the Multivalent EG by decomposing them into binary relations between pairs of entities. This is another application of the Multivalent DIH. Argument roles are maintained, so each binary is a window into its higher-valency predicate, allowing higher-valency predicates to entail lower binaries and unaries.

To learn these new kinds of connections, a method of local entailment rule learning is developed using the MDIH. As in §3.2, in the local step are learned the initial directed edges of the entailment graph, which are further improved with global learning. This step learns entailments by machine-reading the NewsSpike corpus (2.3GB in size), which contains 550K news articles, or over 20M sentences (Zhang and Weld, 2013). NewsSpike consists of multi-source news articles collected within a fixed timeframe, and due to these properties the articles frequently discuss the same events but phrased in different ways, providing appropriate training evidence.



This evening, Mr. Boddy was killed.

$\Rightarrow \text{KILL}.2(\text{Mr.-Boddy})$

Figure 3.1: The MoNTEE system extracts relations. A sentence is CCG parsed, formed into a dependency graph (shown) using CCG dependencies, and traversed to extract a unary relation. MoNTEE traverses from a predicate to all connected arguments.

3.4.1 Extraction of Predicate Relations

A pipeline of steps is used to process raw article text into a list of propositions, predicates with associated typed arguments. The MoNTEE system (Bijl de Vroe et al., 2021) is used for this extraction of natural language relations from raw text². This system first parses sentences using the RotatingCCG parser (Stanojević and Steedman, 2019) (Combinatory Categorical Grammar; Steedman, 2000) and then forms dependency graphs from the parses. Finally, it traverses these graphs to extract the relations, each consisting of a predicate and its arguments. Figure 3.1 shows an example dependency graph and the relation extracted from it. Arguments may be either named entities³ or general entities (noun phrases). These entities are annotated with one of 49 FIGER base types (+ 1 default type :thing used in failure cases) (Ling and Weld, 2012). This is done by first linking entities using AIDA-Light (Nguyen et al., 2014) to their Freebase IDs (Bollacker et al., 2008), and mapping the IDs to the types.

Both binary and unary relations are extracted from the corpus if they contain at least one named entity, which helps anchor to a real-world event. This poses a challenge as noted by Szpektor and Dagan (2008). While binary predicates may be extracted from dependency paths between two entities, unary predicates only have one endpoint, so linguistic knowledge must be carefully applied to extract meaningful unary relations. The following neo-Davidsonian event cases are extracted:

- One-argument verbs including intransitives, e.g. “Knowles sang” \Rightarrow SING.1(Knowles) and passivized transitives, e.g.
“Bill H.R. 1 was passed” \Rightarrow PASS.2(Bill-HR1)
- Copular constructions, where copular “be” acts as the main verb, e.g.
“Chiang is an author” \Rightarrow BE.AUTHOR.1(Chiang)
and where it does not, e.g.
“Phelps seems to be the winner” \Rightarrow SEEM.TO.BE.WINNER.1(Phelps)

As with binaries in earlier work, unary predicates are lemmatized, and tense, aspect, modality, and other auxiliaries are stripped. The CCG argument position which corresponds to its case (e.g. 1 for nominative, 2 for accusative), is appended to the predicate. Passive predicates are mapped to active ones. Modifiers such as negation and predicates like “planned to” as in “Professor Plum planned to attend” are also extracted in the predicate.

²Modality tagging is disabled in this work.

³Identified by the CoreNLP Named Entity Recogniser (Manning et al., 2014; Finkel et al., 2005).

Special attention is paid to copular constructions, which always introduce stative predicates, rather than events (Vendler, 1967). These are interesting for modeling the properties of entities.

3.4.2 Learning Local Graphs

In previous research on binary predicate Entailment Graphs (Hosseini et al., 2018) a representation vector is computed for each typed predicate in the graph. These are compared via the DIH to establish entailment edges between predicates. The features of each vector are typically based on the argument pairs seen with that predicate. Specifically, a typed predicate p has typing $\tau(p) = (t_1, t_2)$, with $(t_1, t_2) \in \mathcal{T} \times \mathcal{T}$. The argument tuples observed with p are denoted by P , containing tuples $a \in \mathcal{E}_{t_1} \times \mathcal{E}_{t_2}$ with \mathcal{E}_t being the subset of all entities of some type t . For each predicate p and observed argument tuple $a \in P$, a corresponding score is calculated $v(p, a)$, the pointwise mutual information (PMI) of p and a .

For example, the predicate $p = \text{BUILD}(:\text{company}, :\text{thing})$ might have a feature (an argument tuple) of $(\text{Apple}, \text{iPhone}) \in P$, where the PMI of “build” with argument pair (Apple, iPhone) is $v(\text{BUILD}, (\text{Apple}, \text{iPhone}))$.

A Balanced Inclusion (BInc) score is calculated for the directed entailment from one predicate to another (Szpektor and Dagan, 2008). BInc is the geometric mean of two subscores: a directional score, Weeds Precision (Weeds and Weir, 2003), measuring how much one vector’s features “cover” the other’s; and a symmetric score, Lin Similarity (Lin, 1998), which downweights infrequent predicates that cause spurious false positives.

These scores are defined for some predicates p and q , with sets of observed argument tuples P and Q , respectively.

$$\begin{aligned}
 BInc(p, q) &= \sqrt{Lin(p, q) * Weeds(p, q)} \\
 Lin(p, q) &= \frac{\sum_{a \in P \cap Q} [v(p, a) + v(q, a)]}{\sum_{a \in P} v(p, a) + \sum_{a \in Q} v(q, a)} \\
 Weeds(p, q) &= \frac{\sum_{a \in P \cap Q} v(p, a)}{\sum_{a \in P} v(p, a)}
 \end{aligned}$$

In this work local binary graphs are computed following Hosseini et al. (2018), and the new MDIH is leveraged to compute additional entailments for unaries and entailments between binary and unary valencies. To do this, feature value subsets are computed for each argument slot respecting its position in the predicate. Slots are compared, rather than predicates, to learn these new entailments. For a predicate p , and slot $s \in \{1, 2\}$, p has features P_s , which are the subtuples formed by selecting arguments in slot s from all observed argument tuples in P . The function $v_s(p, a_s)$ maps to the PMI of observing the argument subtuple a_s in slot s of predicate p . $\tau_s(p) = t$ is also defined, the type of slot s in predicate p . A representation vector of feature PMI values is computed for the slot in unary relations and both slots in binaries. Each slot vector for p has size $|\mathcal{E}_t|$, the size of all possible entities with type t .

Continuing the example, two vectors are calculated for BUILD(:company, :thing): $\mathbf{v}_1 \in \mathbb{R}^{|\mathcal{E}_{\text{company}}|}$ which contains a feature for Apple, and $\mathbf{v}_2 \in \mathbb{R}^{|\mathcal{E}_{\text{thing}}|}$ which contains a feature for iPhone.

Slot vectors are comparable if they represent the same entity type. Edges are learned by comparing corresponding slot vectors between predicates, and calculating a BInc score as in earlier work (Hosseini et al., 2018). For instance, DEFEAT(:person_1, :person_2) \models BE.WINNER(:person_1) is learned by comparing the slot 1 vector of DEFEAT with the slot 1 vector of BE.WINNER. Here, the typed arguments are numbered for demonstration to show that unary entailments apply to a specific argument, even if both premise arguments have the same type. If the entities who have defeated someone are usually found amongst the entities who are winners then a high BInc score is obtained, indicating *defeat* entails that its subject *is a winner*.

Figure 3.2 illustrates a constructed Multivalent Entailment Graph. This includes two classes of subgraph: Bivalent Graphs which contain the entailments of binary predicate premises ($\mathbf{B} \rightarrow \mathbf{B}$ and $\mathbf{B} \rightarrow \mathbf{U}$ edges), and separate Univalent Graphs which contain the entailments of unary predicate premises (only $\mathbf{U} \rightarrow \mathbf{U}$ edges, since a unary is not allowed to entail a binary). As in previous research, separate disjoint subgraphs are learned for each typing, up to $|\mathcal{T}|^2$ bivalent and $|\mathcal{T}|$ univalent subgraphs (given enough data, such that predicates are observed with every combination of typings). For example, the bivalent (:person, :location) graph contains binary predicates such as FLY.INTO(:person, :location) which may entail unaries like BE.AIRPORT(:location).

Because a unary has only one type t_i it may be entailed by binaries in up to $2 * |\mathcal{T}| - 1$ subgraphs with types $\{(t_i, t_j) \mid j \in \mathcal{T}\}$, i.e. all bivalent graphs containing type t_i . For space efficiency, entailments are learned from unary predicate premises

($U \rightarrow U$ entailments) in separate 1-type univalent graphs. Thus, one set of entailments is learned for each unary, just as for each binary, but they may be freely entailed by higher-valency predicates, e.g. binaries in the higher bivalent graphs.

Following from this, bivalent graphs point transitively into univalent graphs. In Figure 3.2, $\text{DEFEAT}(\text{:person_1}, \text{:person_2}) \models \text{BE.WINNER}(\text{:person_1})$ in the person-person bivalent graph. Further entailments of $\text{BE.WINNER}(\text{:person})$ are learned in the person univalent graph.

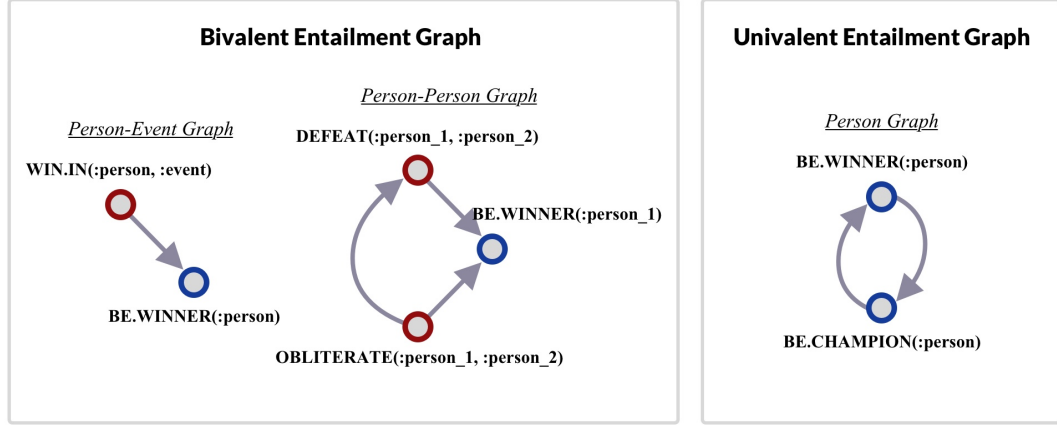


Figure 3.2: Bivalent graphs model entailments from binary predicate premises to equal- and lower-valency predicates (binary and unary). Univalent graphs model entailments from unary predicate premises to equal-valency unary predicates.

3.4.3 Learning Global Graphs

Entailment Graphs constructed from local learning suffer from edge sparsity, as discussed in Chapter 2. This can be improved by further applying “global” graph learning techniques. This work uses the soft constraint method of Hosseini et al. (2018) which has two optimizations. The paraphrase resolution constraint encourages predicates within the same-typed graphs that entail each other to have similar entailment patterns. For example, $\text{BUY}(\text{:person}, \text{:thing})$ mutually entails $\text{PURCHASE}(\text{:person}, \text{:thing})$, so entailments of $\text{BUY}(\text{:person}, \text{:thing})$ can be copied as entailments of $\text{PURCHASE}(\text{:person}, \text{:thing})$. The cross-graph constraint additionally encourages similar predicates across different typed graphs to share entailment patterns.

Global soft-constraint learning is applied to bivalent graphs and separately to univalent graphs. Globalization is valency-agnostic, using just the graphical structures between predicates, so bivalent graphs are optimized using both $\mathbf{B} \rightarrow \mathbf{B}$ edges (as in

Hosseini et al. (2018)) and the new $\mathbf{B} \rightarrow \mathbf{U}$ edges to optimize binary predicate entailments. The final graph size statistics are in Table 3.1.

Valency	Vertices		Edges
Bivalent	938K Binary	94M $\mathbf{B} \rightarrow \mathbf{B}$ / 30M $\mathbf{B} \rightarrow \mathbf{U}$	
Univalent	36K Unary		3.6M $\mathbf{U} \rightarrow \mathbf{U}$

Table 3.1: 546 typed bivalent subgraphs are learned, which contain entailments of binary predicate premises ($\mathbf{B} \rightarrow \mathbf{B}$ and $\mathbf{B} \rightarrow \mathbf{U}$); and 37 typed univalent subgraphs which contain entailments of unary predicates ($\mathbf{U} \rightarrow \mathbf{U}$).

3.5 Methods: Constructing a Natural Multivalent QA Task

An automatically generated QA task is posed to evaluate the multivalent model explicitly for directional inference between binary and unary predicates, as there are no known standard datasets for this problem. The task is to answer true-false questions about real events that are discussed in the news, for example, “Was Biden elected?” These types of questions are surprisingly difficult and frequently require inference to answer (Clark et al., 2019). For this, entailment is especially useful: a model must decide if the question (hypothesis) is true given a list of propositions from limited news text (premises), which are all likely to be phrased differently.

This task is designed independently of the Multivalent Entailment Graph as a challenge in information retrieval. Positive questions made from binary and unary predicates are selected directly from the news text using special criteria, and are then removed. From these positives are automatically generated false events to use as negatives, which are designed to mimic real, newsworthy events. The remaining news text is used to answer the questions. This design attempts to make every question answerable, but since questions are generated automatically there is no guarantee. However, the task is fair as all models are given the same information. The additive effects of multivalent entailment should be demonstrated: by using more kinds of entailment, the Multivalent Entailment Graph should find more textual support and answer more questions.

The task is presented on a text sample from NewsCrawl, a multi-source corpus of news articles (Barraut et al., 2019). A test set is extracted which contains 700K

sentences from articles over a period of several months, and also a development set from a further 500K sentences. Generated questions are balanced to a ratio of 50% binary questions / 50% unary; and within each 50% positive / 50% negative. Table 3.2 shows a sample from the dev set. 34,394 questions are generated for the test set: 17,256 unary questions and 17,138 binary.

3.5.1 Question Generation

For realism, questions should be both *interesting* and *answerable* using the corpus. A multi-step process extracts questions from the news text itself.

1. Partitioning First, the articles are grouped by publication date such that each partition covers a timespan of up to 3 consecutive days of news (49 partitions in the test set). True-false questions are asked about events drawn from the partition, and the news text within this 3-day window is used as evidence to affirm or deny them. The questions are asked as if happening presently in this time window to control for the variable of time, so that ambiguous questions may be asked like “Did the Patriots win the Superbowl?” which may be “true” or not depending on the date and timespan. The small 3-day window size was chosen so that multiple news stories about an event appear together, increasing the chances of finding question answers. Within each partition, the extraction of predicate-argument relations is done in a process mirroring §3.4.1.

2. Selecting Positives A selection process is adapted from Poon and Domingos (2009) to choose questions which are both interesting to a human and answerable from the partition text. First, the most repeated entities are identified in the partition; it is assumed that questions involving these entities will be interesting to a human, since they star in the events of the articles. Additionally, the frequency of mentions for these entities yields ample textual evidence for answering questions about them. In each partition the mentions are counted of each entity pair (from binary propositions) and single entities (from unary and binary ones). The most frequent entities and entity pairs mentioned more than 5 times in the partition are selected. After this, a pool of predicates is selected from those mentioned across the entire news corpus more than 10 times; it is assumed these are popular to report in news and thus are interesting to a human questioner. Finally, propositions are randomly selected from those featuring both a star entity and predicate to use as questions, and are removed from the partition.

3. Generating Negatives A simple strategy for producing negatives might seem to be substituting random predicates into the positive questions. However, this is unsatisfactory because modern techniques in NLP excel at detecting unrelated words. For example, a neural model will easily distinguish a random negative like DETONATE(Google, YouTube) from a news text discussing Google’s acquisition of YouTube, classifying it as a false event on grounds of dissimilarity alone.

To be a meaningful test of inference this task requires that negatives be difficult to discriminate from positives: they should be semantically related but should not logically follow from what is stated in the text. To this end negative questions are derived from the selected positives using linguistic relations in WordNet (Fellbaum, 1998). It is assumed that news text follows the Gricean cooperative principle of communication (Davis, 2019), such that it will report what facts are known and nothing more. To this end, noun hyponyms and their verbal equivalent, troponyms, are mined from the first sense of each positive in WordNet. For example, “burn” is extracted as a troponym of “hurt” and the phrase “inherit from” as a troponym of “receive from.” Therefore it is expected that these specific relations will be untrue of the argument tuple in question and may be used as negatives. Antonyms and other WordNet relations were also considered, but these have low coverage and are much sparser in English.

For fairness, generated negatives which actually occur in the current partition are screened out (0.1% of proposed negatives), as well as negatives which never occur in the entire corpus (76.8% of proposed negatives). Only challenging negatives are left, with predicates that actually do occur in real news text. See Table 3.2 for a sample of questions. In the error analysis it is found that these negatives are of good quality: they are uncommonly inferable from the text, accounting for a small percentage of false positives.

3.5.2 Question Answering Models

In each partition, models receive factual propositions extracted from 3 days of news text to use as evidence for answering true-false questions. A model scores how strongly it can infer the question proposition from each evidence proposition, and the maximum score is taken as the model confidence of a “true” answer.

Exact-Match The text is multi-source news articles, so world events are often discussed multiple times in the data, even with the same phrasing. An “exact-match”

Positive	Negative
Did the Ohio State Buckeyes play ?	Did the Ohio State Buckeyes fumble ?
Was Mitt Romney a candidate ?	Was Mitt Romney a write-in ?
Did voters reject Mike Huckabee?	Did voters discredit Mike Huckabee?
Did Roger Clemens receive from Brian McNamee?	Did Roger Clemens inherit from Brian McNamee?

Table 3.2: A sample of generated questions from the dev set. Positives are taken from the text and reworded as questions. Negatives are created from sampled positives by generating a more specific hyponym/troponym from the **bolded** predicate.

baseline is computed which shows how many questions can be answered from an exact string match in the text; the rest require inference.

Binary Entailment Graph The $\mathbf{B} \rightarrow \mathbf{B}$ model is roughly equivalent to the state of the art binary-to-binary entailment graph (Hosseini et al., 2018), so it serves as a baseline for the overall model.⁴

All graph models look for directed entailments from evidence propositions to the question proposition. For example, “Was YouTube sold to Google?” can be answered affirmatively by reading “Google bought YouTube” using the graph edge $\text{BUY}(x, y) \models \text{SELL.TO}(y, x)$. BInc scores range from 0 to 1; if no entailments are found it is assumed that it is false (score of 0).

Multivalent Entailment Graph The Multivalent Graph is made of 3 component models: (1) the $\mathbf{B} \rightarrow \mathbf{B}$ model which uses binary evidence to answer binary questions; (2) the $\mathbf{U} \rightarrow \mathbf{U}$ model which uses unary evidence to answer unary questions; and (3) the $\mathbf{B} \rightarrow \mathbf{U}$ model which uses binary evidence to answer unary questions. The Multivalent EG is able to answer questions using evidence across valencies, e.g. “Is J.K. Rowling an author?” is affirmed by reading “J.K. Rowling wrote *The Sorcerer’s Stone*” using the graph edge $\text{WRITE}(x, y) \models \text{BE.AUTHOR}(x)$. Individually, each model answers only binary or unary questions, not both. By combining them, all kinds of questions can be answered using all available evidence. At each precision level if any component model

⁴The Multivalent Graph is tested on the Levy/Holt dataset of 18,407 questions for $\mathbf{B} \rightarrow \mathbf{B}$ entailment (Levy and Dagan, 2016; Holt, 2018), and it achieves similar results to Hosseini et al. (2018).

predicts true, the overall model does too.

In some test instances the entity typer may make an error, resulting in a failure to find the question predicate in the typed subgraph. Similarly to [Hosseini et al. \(2018\)](#), in these cases the method defaults to backing off, querying all subgraphs for the untyped predicate and averaging the entailment scores found. 5% more unary questions are found and 18% more binaries by backing off.

Similarity Models BERT and RoBERTa predicate embeddings ([Devlin et al., 2019](#); [Liu et al., 2019](#)) are used in an unsupervised manner to answer questions based on similarity to the evidence. The question is encoded into a representation vector, and so is each evidence proposition with the same arguments. The cosine similarity is computed between the question and each evidence vector, adjusted to a scale of 0 to 1: $\text{sim}(\mathbf{p}, \mathbf{q}) = (\cos(\mathbf{p}, \mathbf{q}) + 1)/2$.

To compute each vector encoding, a simple natural language sentence is constructed from the proposition using its predicate and arguments and encoded with the language model. A representation includes *only* the encoding for the predicate in the context of its arguments, but not the arguments themselves to make this a true test of predicate similarity. To do this, an average is taken over all final hidden-state vectors from the model corresponding to the predicate, excluding those of the arguments. The base BERT model and RoBERTa model are tested, which has robustly pretrained on 160GB of text (76GB news).

PPDB Though supervised, PPDB 2.0 (The largest XXXL version is used) ([Pavlick et al., 2015](#)) is a useful comparison as it is a large, well-understood resource for phrasal entailment. PPDB relations are extracted from bilingual pivoting and are categorized using text-based features, which is very different from our argument-tracking method. PPDB may be viewed as a kind of Entailment Graph with 9M predicate phrases (vertices) and 33M combined “Equivalence” and “ForwardEntailment” edges. As with the other models, evidence and question propositions are converted into a natural text format and a PPDB relation score is extracted from each pairing of the question with an evidence phrase.

3.6 Experiment 1: All Questions from Text

For each partition, models are presented with all corresponding sampled questions and the relevant supporting propositions, which are pre-identified to contain the same query entities. Models must compare each supporting statement to the query and make a judgement as to whether an entailment holds. Models affirm the hypothesis question if any premise entails it (taking the maximum score if there are multiple entailments), or deny the hypothesis if no entailment holds from any supporting premise. The models produce a gradation of judgement scores between 0 (false) and 1 (true).

3.6.1 Results

As in earlier work, a classification threshold slides over the score range to produce a precision-recall curve for each model. Results are in Figure 3.3 (left).

Multivalent graph performance is shown incrementally. The **B**→**B** model can answer a portion of binary questions; the **U**→**U** model can answer more unary questions; adding the **B**→**U** model can answer still more unary questions using binary evidence. Successful inference of the kill/die example is observed and others. “Obama was elected to office” affirms the question “Was Obama a candidate?” and “Zach Randolph returned” affirms “Did Zach Randolph arrive?”

This test set is from multiple sources over the same time period. The exact-match baseline shows the limitations of answering questions simply by collecting more data; most questions require inference to answer. The complete Multivalent EG achieves ~3x this recall by drawing inferences.

The Multivalent Entailment Graph achieves higher precision than BERT and RoBERTa similarity models in the low recall range. The similarity models perform well, achieving full recall by generalizing for rarer predicates. Notably, RoBERTa bests BERT likely due to its extensive in-domain pretraining.

Notably, the **B**→**B** entailment type appears to struggle in terms of recall, relative to the other entailment types within the multivalent graph. In fact 90.5% of unary questions have a vertex in the graph, but only 64.1% of binaries do. The **B**→**B** model frequently cannot answer questions because the question predicate wasn’t seen in training. This difference is because binary predicates are more varied, so suffer more from sparsity: they are often multi-word expressions and have a second, typed argument. Indeed, most binary predicate research (in symbolic methods) focuses on only the top 50% of recall in several datasets (Berant et al., 2010, 2015; Levy and Dagan, 2016;

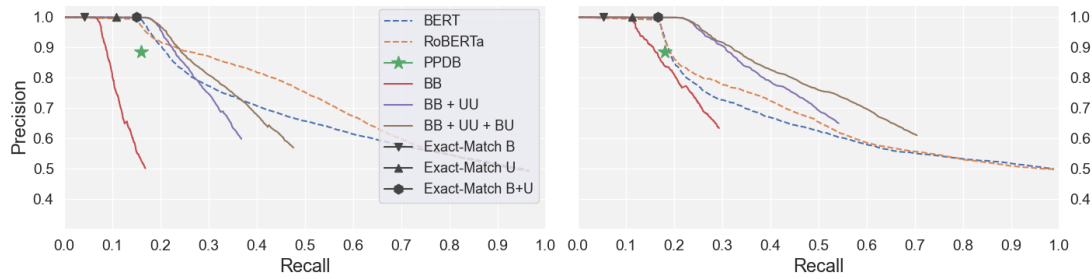


Figure 3.3: (Left) Overall performance on the QA task (Experiment 1). (Right) performance on the filtered task (Experiment 2). Note that $\mathbf{B} \rightarrow \mathbf{B}$, $\mathbf{U} \rightarrow \mathbf{U}$, and $\mathbf{B} \rightarrow \mathbf{U}$ models may individually reach a max recall of 50% because they answer only binary or unary questions.

Hosseini et al., 2018). This problem of vertex sparsity in Entailment Graphs is explored deeply in Chapter 4.

3.7 Experiment 2: Questions Within-Distribution

For an even comparison, a filtered question set is created to compare the models when both the Entailment Graphs and the similarity models have a chance to answer all the questions. From the question set in Experiment 1, all questions without a vertex in the Entailment Graph are removed, then the remaining questions are balanced as in §3.5, resulting in 20,519 questions (10,273 unary and 10,246 binary). The removed questions either come from outside the training distribution, or the long tail of it, with so few mentions that entailments cannot be learned.

Questions are constrained to exactly the EG training distribution, but this should also approximate RoBERTa’s training distribution, which consists of many similar news texts, so this test is fair between models. This comparison aims to show that when questions in-domain of training are identified, EG performance surpasses other baselines.

3.7.1 Results

The results are shown in Figure 3.3 (right), showing a very different outcome than Experiment 1. Head-to-head, the Multivalent Entailment Graph offers substantially better precision across all recall levels. At 50% recall, the EG has 76% precision while RoBERTa has 65%.

Model	Unary Questions		Binary Questions	
	@1451	@2000	@802	@2000
BERT	91.4%	76.9%	92.0%	82.9%
RoBERTa	92.5%	78.6%	91.5%	85.1%
PPDB	92.3%	—	81.8%	—
<i>Multivalent EG</i>				
U→U	96.5%	87.0%	—	—
B→U	97.6%	90.4%	—	—
B→B	—	—	100.0%	88.8%
1245 Exact-Match			597 Exact-Match	

Table 3.3: The filtered test. Models rank question/answer pairs by confidence. Accuracy is shown for the K most confident predictions, at two points. PPDB doesn’t answer enough questions to reach the @2000 cutoff, so the smaller PPDB maximum is also compared.

Notably, on both tests, more unary questions are answered using both unary *and* binary predicate evidence than just using unary evidence alone. On the filtered test, the **B→U** model increases max recall from 54% to 70%.

Finally, PPDB appears to have poor performance (highest recall shown), only 1% higher recall than the exact-match baseline despite having entries for 88% of questions. Though PPDB features many directional entailments, it suffers from edge sparsity worse than Entailment Graphs. This may be because the technique of bilingual pivoting used in PPDB’s construction excels at detecting near-paraphrases, not relations between distinct eventualities, e.g. it can’t learn “getting elected” entails “being a candidate.” Advantageously, the Entailment Graph learning method acquires this open-domain knowledge by tracking entities across all the events they participate in.

The results of the filtered test results are shown in detail in Table 3.3. Models don’t answer all the questions, so following Lewis and Steedman (2013) who design a similar QA task, models are evaluated on the accuracy of their K most confident predictions.

In agreement with the precision-recall curves, the most confident predictions by Multivalent Entailment Graphs are shown to be more accurate than those of similarity models or PPDB. In particular, the **B→U** predictions which answer unary questions

with binary predicate evidence achieve 90.4% accuracy at the @2000 cutoff, compared to 78.6% accuracy for RoBERTa using either valence of evidence. Using binary evidence for unary questions ($\mathbf{B} \rightarrow \mathbf{U}$) is even 3.4% more accurate at the same cutoff than using unary evidence ($\mathbf{U} \rightarrow \mathbf{U}$), likely due to the extra disambiguating information provided by the additional argument in the binary relation.

3.8 Error Analysis

300 false positives are sampled (100 for each model) and the results of a manual process of categorizing by error type is reported in Table 3.4. In all models, spurious entailments are the largest issue, and may occur due to normalization of predicates during learning, or incidental correlations in the data. The $\mathbf{U} \rightarrow \mathbf{U}$ and $\mathbf{B} \rightarrow \mathbf{U}$ models also suffer during relation extraction (parsing). In cases of failure to parse a second argument for a predicate it is assumed that it only has one and so a malformed unary is extracted, which can interfere with question answering (e.g. reporting verbs “explain,” “announce,” etc. which fail to parse with a long quote). Relatively few poorly generated negatives are found, which are actually true given the text. In these cases the model finds an entailment which the authors judge to be correct.

3.9 Conclusion

The MDIH is shown as an effective theory of unsupervised, open-domain predicate entailment, which learns entailments within and across valencies by respecting argument roles.

The Multivalent Entailment Graph’s performance has been demonstrated on a question answering task requiring fine-grained semantic understanding. The multivalent EG is able to answer a broader variety of questions than earlier entailment graphs, aided by drawing on evidence across valencies. Several baseline models are also outperformed, including a strong similarity measure using unsupervised BERT and RoBERTa models, while using far less training data. This shows that directional entailment is more helpful for inference on such a task than non-directional similarity, even with robust, in-domain pretraining.

This work indicates a potential complementarity between unsupervised methods. The symbolic Entailment Graph method achieves high precision for learned predicates, while sub-symbolic neural models achieve high recall by generalizing to unseen

Error Source	False Positive Example
Unary to Unary ($U \rightarrow U$) Judgements	
Spurious Entailment (57%)	The United States advances \models The United States falls
Parsing (26%)	Reuters reports \models Reuters notes
Poor Negative (actually true) (17%)	Productivity increases \models Productivity grows
Binary to Unary ($B \rightarrow U$) Judgements	
Spurious Entailment (65%)	New York Mets create through camerawork \models New York Mets benefit
Parsing (26%)	John McCain spent part of 5 years \models John McCain drew
Poor Negative (actually true) (9%)	The Yankees overwhelm the Mariners \models the Yankees prevail
Binary to Binary ($B \rightarrow B$) Judgements	
Spurious Entailment (53%)	A soldier was killed in Iraq \models A soldier was murdered in Iraq
Poor Negative (actually true) (32%)	Profits fall in the first quarter \models Profits decline in the first quarter
Parsing (17%)	medal than United States \models United States take the medal

Table 3.4: False positive analysis. Models predict entailments from the text (premise) to generated negatives (hypothesis).

predicates. Chapter 4 presents research leveraging the benefits of both models in an unsupervised way, while maintaining directional precision.

3.9.1 Limitations

This work introduces new development in the automatic learning of predicate inferences, with the addition of detecting entailments between predicates of different valencies. Entailment Graphs specialize on predicates, and do not model entailment for other sentential content, such as the entailment of nouns. Recent work has integrated other factors such as modality and temporal signals into EG construction (Bijl de Vroe, 2023), but EGs still lack coverage of other features of language such as control verbs

which scope over other verbs, like x *failed to* $p(y)$. Critically, while EGs can learn entailments for any simple predicate in the open domain of natural language, they face an important problem in their inability to handle out-of-vocabulary predicates. This makes EGs operating alone unable to complete a task, such as answering a question, if either the premise or hypothesis does not have a vertex in the graph.

This work also introduces a new evaluation method for Entailment Graphs, automatically-generated Boolean Question Answering from news text. While useful for evaluating models at a large scale of generated questions, the quality of questions themselves is not guaranteed to be as good as if manually created. Heuristics for question generation were carefully thought out such that in aggregate, the corpus of questions is a meaningful evaluation, but it is possible that individual questions may be imperfect. A generated question could be too obviously answerable (either positively or negatively), or be malformed through a fault in any of the relation-extraction steps, or even be uninteresting to actual humans.

3.9.2 Ethical Considerations

As unsupervised models, the learned entailments are dependent entirely on the quality of the training corpus, which risks learning social biases implicit in the data. For example, it is possible to learn an association $\text{BE.MALE}(x) \models \text{BE.MANAGER}(x)$. However, the learning of entailments via the (M)DIH and a score like BInc is a process designed to upweight correlations which are both: (1) strongly overlapping in their context-based representations, and (2) corroborated often, by multiple sources. Therefore, it is important to train on text which has gone under editorial review, such as news articles from reputable sources, which minimizes the risk of learning implicit biases. Increasing the size of the training dataset may also help distinguish majority vs. minority associations, and in general it is unlikely that a small number of incidental overlaps will lead to strong entailment scores. EGs are also explicit structures, and it is easy to edit or delete learned entailments after construction.

Chapter 4

Smoothing Entailment Graphs with Language Models

This chapter addresses the problem of general vertex sparsity in Entailment Graphs. Though the Multivalent Distributional Inclusion Hypothesis may be theoretically capable of learning predicate entailments in an open domain of natural language, a fundamental problem prevents this in practice. The diversity and Zipfian frequency distribution of natural language predicates in training corpora leads to inevitable vertex sparsity in Entailment Graphs (EGs) built by Open Relation Extraction (ORE). As symbolic models for natural language inference, Entailment Graphs fail if a novel premise or hypothesis predicate is missing at test-time.

First, to overcome general vertex sparsity, a theory is introduced to optimally “smooth” an Entailment Graph by finding suitable replacement predicates for missing entries. This is done by constructing transitive chains in order to preserve directional inference capability while extending beyond an EG’s predicate vocabulary. Next, an efficient, open-domain smoothing method is demonstrated using a simple off-the-shelf Language Model, which finds 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 similar boolean QA task as designed in Chapter 3, it is shown that EG smoothing of premises is most useful for answering questions with lesser supporting text, where missing predicates are more costly. Finally, in controlled experiments with WordNet it is shown that hypothesis smoothing is difficult, but also possible in principle.

4.1 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 typically 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. A *directional inference* is stricter than paraphrase or similarity, in that it is true only in one direction, but not both, e.g. $\text{DEFEAT} \models \text{PLAY}$ but $\text{PLAY} \not\models \text{DEFEAT}$ (where \models means “entails”). Directional inferences are difficult to learn, but crucial to language understanding.

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). EGs are relatively very data- and compute-efficient, requiring less than two 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 connectivity (Berant et al., 2015; Hosseini, 2021; Chen et al., 2022) but this work is the first to formally acknowledge **vertex sparsity**, arising when a predicate is not seen at all in training. EGs are structures of symbols, so they cannot handle missing 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’s impractical to scale up learning predicate symbols from corpora.

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., 2022a), offering relief to sparsity problems via techniques like in-context learn-

ing (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; or are costly to query via API; and centralizing models under private companies opens challenges of data privacy. Thus it remains important to research lower-compute and more data-efficient methods which run on the scale of a single GPU.

This work is the first to define vertex sparsity and approach the problem by applying a small, pretrained LM to improve an existing EG using the benefits of modern embeddings. Four contributions are offered in this chapter:

1. A theory is presented for optimal smoothing of EG vertices by constructing transitive chains, accounting for a distinction between premise and hypothesis.
2. A low-compute method is demonstrated for unsupervised smoothing of EG vertices using LM embeddings to find approximations of missing predicates (see Figure 4.1). Applied to premises, recall is improved by 16.3 and 25.1 percentage points on Levy/Holt and ANT entailment datasets while raising precision.
3. On a QA task, it is shown that LM premise smoothing is most helpful when there is less supporting context and missing a predicate is more costly.
4. Finally, in controlled experiments with WordNet relations, the behavior of the LM for premise smoothing is confirmed, and it is shown that hypothesis smoothing is possible, but more difficult.

4.2 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.

The most direct comparison for this work 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

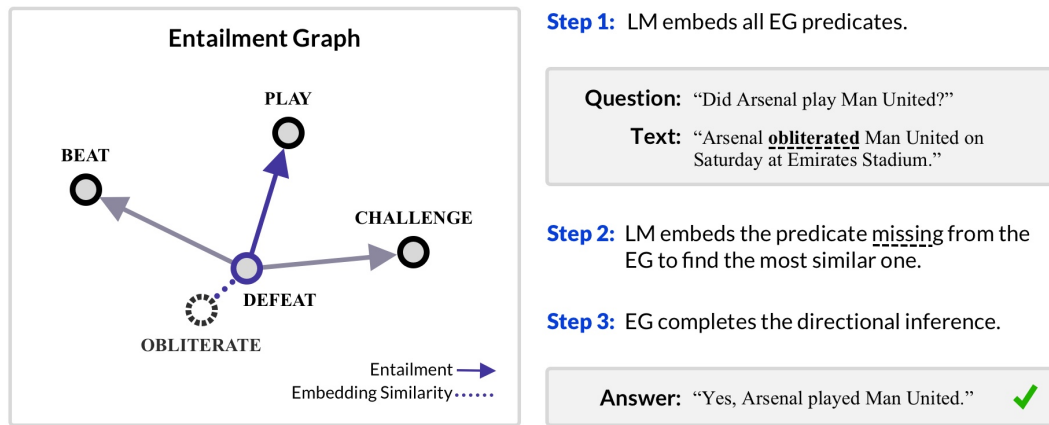


Figure 4.1: The question “Did Arsenal play Man United?” cannot be answered because the predicate “obliterate” in the text isn’t in the Entailment Graph. An LM embeds “obliterate” so a nearest neighbor in the EG can be found, completing the directional inference.

on premise-hypothesis pairs and labels from the development split of the Levy/Holt NLI dataset (Holt, 2018), used in the experiments of this work. 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. $\text{BUY} \models \text{OWN}$ and $\text{OWN} \not\models \text{BUY}$), this technique picks up dataset artifacts spuriously correlated with the labels in datasets such as Levy/Holt. In contrast, the approach in this chapter combines the strengths of each: open-domain encoding using a computationally tractable LM with the directional capability of an EG.

In this work sub-symbolic encoding by an LM is achieved leveraging WordPiece tokenization (Devlin et al., 2019) as a means of generalizing beyond a fixed vocabulary of predicates.

Briefly, WordPiece is an algorithm which translates the vocabulary learned from a fixed list of words into a fixed list of *subword* units, which may be combined to represent an infinite number of full words. The algorithm is trained on a corpus similarly to the BPE tokenization algorithm (Sennrich et al., 2016) to compute an optimal vocabulary, and thereafter the vocabulary may be applied to a text to tokenize it for model training or evaluation.

Initially, the tokenizer training corpus is broken into characters, then an iterative algorithm proceeds to build up a vocabulary. Characters which are not first in their

respective word have “##” prepended to identify them as part of a larger word. The iterative process of merging tokens occurs until the desired vocabulary size is met. Consecutive pairs of tokens are scored based on how often they each occur, and occur together, and are merged into a larger token and added to the vocabulary if they have the highest score. An example is shown:

“corpus words” \rightarrow “c” “##o” “##r” “##p” “##u” “##s” “w” “##o” “##r” “##d” “##s”
 V_8 : {c, ##o, ##r, ##p, ##u, ##s, w, ##d}
 V_9 : {c, ##o, ##r, ##p, ##u, ##s, w, ##d, ##or}
 V_N : ...

In this example, the pair “or” appears twice (the most often in this two-word corpus), so it is merged into a new token and the vocabulary size increases from 8 to 9. This process is iterated until the desired vocabulary size N is obtained. This yields a vocabulary of subword units which is optimal both for representing the training corpus using a minimal number of tokens, and for minimizing the number of tokens required to build out-of-domain words that are similar to the training domain. This vocabulary can be applied for language modeling, or other tasks. [Sennrich et al. \(2016\)](#) show that subword tokenization schemes are extremely effective for representing rare words which may occur at test time, but not in training, yet nonetheless may be composed of subword units which *do* occur often in training.

4.3 Theory of Smoothing

First, a theory for optimal smoothing of an EG is presented, which overcomes the problem of vertex sparsity. The name “smoothing” is chosen in reference to earlier work smoothing n-gram Language Models, where information for missing vocabulary words is approximated using existing vocabulary words ([Chen and Goodman, 1996](#)).

Afterward, the theoretical intuition behind applying an LM as an open-domain smoother is discussed. Importantly, this work distinguishes ways to **P-smooth** premises and **H-smooth** hypotheses.

4.3.1 Directionality by Transitive Chaining

It is most important when modifying EG predictions by smoothing to maintain the EG’s strong directional inference capability. A theory is now presented 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.

To begin, a query entailment relation is defined: $Q : p \Rightarrow h$. For queries, the symbol \Rightarrow is used instead of \models to denote that the query truth value is unknown, and must be verified by a model. In this situation, the model at hand is missing entries for at least p or h (denoted by a dashed underline). *Smoothing* is now defined as the process of generating a new query relation Q_s suitable for the model by identifying a replacement predicate p' and/or h' within the model's vocabulary (replacement denoted by a solid underline), such that the model can now verify if there is a relationship between premise and hypothesis or not. To maintain directional precision, this must be done by identifying a p' (or h') with a specific relation to p (or h). Doing so allows the creation of a transitive chain connecting Q and Q_s ; thus, confirmation of Q_s by the model may then be leveraged to confirm Q . There are three approaches to smooth Q depending on whether p is missing from the model, h is missing, or both.

- **Generalize missing p .** If missing p , identify a more general premise p' in the EG such that $p \models p'$. This yields a smoothed query $Q_s : p' \Rightarrow h$. Now, if the EG confirms $p' \models h$, then $p \models p' \models h$, and $p \models h$ (Q) is confirmed by transitivity.

$$\begin{array}{ccc}
 & p & h \\
 (Q) & \text{"a obliterated b"} & \Rightarrow \text{"a played b"} \\
 & \Pi & \\
 (Q_s) & \text{"a beat b"} & \Rightarrow \text{"a played b"} \\
 & p' & h
 \end{array}$$

- **Specialize missing h .** If missing h , identify a more specialized hypothesis h' in the EG such that $h' \models h$. This yields a smoothed query $Q_s : p \Rightarrow h'$. Now, if the EG confirms $p \models h'$, then $p \models h' \models h$, and $p \models h$ (Q) is confirmed by transitivity.

$$\begin{array}{ccc}
 & p & h \\
 (Q) & \text{"a bought b"} & \Rightarrow \text{"a shopped for b"} \\
 & & \sqcup \\
 (Q_s) & \text{"a bought b"} & \Rightarrow \text{"a paid for b"} \\
 & p & h'
 \end{array}$$

- **Generalize missing p and specialize missing h .** If missing both p and h , do a combination of the above approaches: identify new p' and h' as above, yielding a new $Q_s : p' \Rightarrow h'$. Now, if a model confirms $p' \models h'$, then $p \models p' \models h' \models h$, and $p \models h$ (Q) is confirmed by transitivity.

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

4.3.2 LM Embeddings and Specificity

It is assumed that p' and h' are respectively among the nearest neighbors of p and h in the embedding space of the LM, and this work proposes a method to leverage LM embeddings in an unsupervised way to find them. As defined later in §4.4, both the target query predicate and EG predicates are embedded, then the embedding space is searched for the K nearest neighbors to the target. It is predicted that doing so for a premise predicate will build a transitive chain satisfying the conditions of §4.3.1. Two factors are now identified which, combined, lead to predictions that are likely more semantically general than the target, which enables P-smoothing, but not H-smoothing.

Factor 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 XLNet (Shwartz and Choi, 2020).

Factor B: The natural anti-correlation of word frequency with specificity in text Probabilistically, the more frequent a word is in text, the lower its “semantic content” (in other words, the less specific it is) 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. This is illustrated further in §4.3.3.

This effect has even been studied elsewhere, such as in translation. Translated text by humans is often dubbed “Translationese,” and by algorithms, “Machine Translationese,” due to its typical use of a core subdomain of the target language, which results in translations with a non-specific tone (Gellerstam, 1986; Rabinovich and Wintner, 2015). In Machine Translation, Vanmassenhove et al. (2021) establish that bias in models toward generating tokens with high frequency in a training corpus accounts for this phenomenon, resulting in a quantified semantic generalizing effect from translation input to output.

4.3.3 The Specificity Taxonomy

To relate frequency and generality for this work, a hierarchical taxonomy of predicates ordered by specificity is illustrated, following from the theories of natural categories and prototype instances (Rosch and Mervis, 1975; Rosch et al., 1976). Very general predicate categories are placed at the top of this taxonomy such as “act” and “move,” with concrete subcategories beneath, and highly specific ones at the bottom, like “inoculate” 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). An analogous basic level is assumed in a predicate taxonomy, too, in Figure 4.2.

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,” “lunge”). 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. “elongate,” “defenestrate”).

This asymmetry encourages P-smoothing using an LM (and foreshadows the fail-

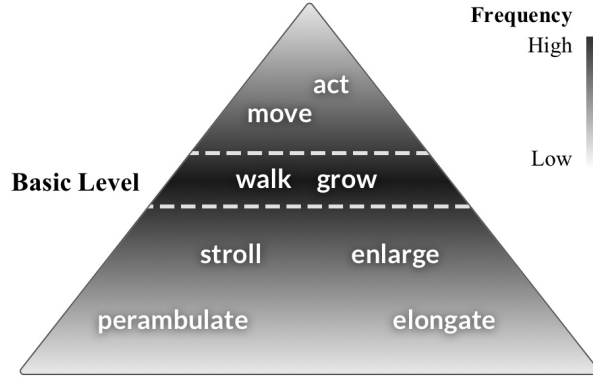


Figure 4.2: The specificity taxonomy. The basic level contains “everyday” predicates. Above this level predicates become more general, and below they become more concrete and specific. Usage frequency decreases away from the basic level.

ure of H-smoothing with 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.

4.4 Methods: Approximating Missing EG Predicates

As this work presents methods of smoothing existing Entailment Graphs, the focus is on previous, high-performing EGs 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 $\text{TRAVEL.TO}(:\text{person}, :\text{location}) \in V$, and the types $:\text{person}, :\text{location} \in \mathcal{T}$. An example entailment is $\text{TRAVEL.TO}(:\text{person}, :\text{location}) \models \text{ARRIVE.AT}(:\text{person}, :\text{location}) \in E$.

The smoothing method in this work may be applied to any existing EG. The benefits of vertex-smoothing are shown to be complementary with existing methods in improving edge sparsity by comparing two related baseline models, described in §4.5. These EGs are learned from the same set of vertices, but are constructed differently, so

they 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, $\text{KILL}(\text{:medicine}, \text{:disease})$ learned in the typed subgraph $g^{(\text{medicine-disease})}$ has a different meaning and entailments than $\text{KILL}(\text{:person}, \text{:person})$.

4.4.1 Nearest Neighbors Search

The K-nearest neighbors method of this work assumes that existing Entailment Graphs contain enough predicates already present in them to enable discovery of suitable replacements for an unseen target predicate, 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 4.1 for an illustration.

The smoothed retrieval function S 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 $g^{(t)}$, the EG predicate vertices $V^{(t)}$ are encoded as a matrix $\mathbf{V}^{(t)}$. For each predicate $v_i^{(t)} \in V^{(t)}$, a row vector $\mathbf{v}_i^{(t)} \in \mathbf{V}^{(t)}$ is encoded via $\mathbf{v}_i^{(t)} = L(v_i^{(t)})$.

At test-time a corresponding vector is encoded for the target predicate x , $\mathbf{x} = L(x)$. Then S retrieves the K-nearest neighbors of x in $g^{(t)}$:

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

$L(\cdot)$ is a function which encodes a typed natural language predicate using a pre-trained 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 4.1 for examples). In this work, the representations of WordPieces corresponding to the predicate are extracted, and averaged into the resulting predicate vector. **RoBERTa** (Liu et al., 2019) is used in experiments for encoding, which is 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.

For the KNN search metric, Euclidean Distance (L^2 norm) is calculated from the target vector \mathbf{x} to vectors in $\mathbf{V}^{(t)}$. A BallTree is precomputed using scikit-learn (Pe-

$x : (\text{join}.1, \text{join}.2) \# \text{person} \# \text{organization}$
\Rightarrow “person join organization”
$x : (\text{give}.2, \text{give.to}.2) \# \text{medicine} \# \text{person}$
\Rightarrow “ give medicine to person”
$x : (\text{export}.1, \text{export.to}.2) \# \text{location}_1 \# \text{location}_2$
\Rightarrow “location_1 export to location_2”

Table 4.1: 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.

dregosa et al., 2011) which spatially organizes the EG vectors in order to speed up the search for a closest neighbor to the target. The runtime bounds of search is improved from linear in the number of vertices $|V^{(t)}|$ to $\log |V^{(t)}|$.

4.4.2 Datasets

This smoothing method is demonstrated on two explicitly directional datasets, which test both directions of predicate inference, creating a 50% positive/50% negative class balance.

Levy/Holt The Levy/Holt dataset (Holt, 2018; Levy and Dagan, 2016) has been explored thoroughly in previous work on predicate entailment (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 fine-tuning in supervised models (Li et al., 2022a). This work tests on the 1,784 questions forming the purely directional subset, which is more challenging.

ANT The newer ANT dataset presents a quality improvement on the Levy/Holt format, and also tests predicate entailment in the general domain (Guillou and Bijl de Vroe, 2023). It was created by a multi-step process. First, experts annotated entailment relations between predicate pairs, then each predicate in the pairings was expanded into a cluster of predicate paraphrases automatically using WordNet and other dictionary resources. The human annotations apply between clusters, creating many-to-many comparisons between predicates. Like Levy/Holt, test questions take

“The audience applauded the comedian” \models “The audience observed the comedian”
“The audience observed the comedian” $\not\models$ “The audience applauded the comedian”

“The laptop satisfied the criteria” \models “The laptop was assessed against the criteria”
“The laptop was assessed against the criteria” $\not\models$ “The laptop satisfied the criteria”

Table 4.2: Example queries, ANT (dev) directional subset.

the format “given [premise], is [hypothesis] true?” This work again tests using the directional-only subset of 2,930 questions.

See Table 4.2 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. The MoNTEE system (Bijl de Vroe et al., 2021) is used to extract CCG-parsed and typed predicate relations (x) shown in Table 4.1, which are used as queries to Entailment Graphs.

4.4.3 Models

This work demonstrates smoothing on 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 it can be clearly observed 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** An LM is applied separately for both P- and H-smoothing on GBL and CTX. As described earlier, the RoBERTa-base LM (Liu et al., 2019) is used to produce embeddings for smoothing the EG.

S&S The finetuned RoBERTa model of Schmitt and Schütze (2021) (discussed in §4.2). As done in their work, this work follows a process of inserting each premise/hypothesis

pair into 5 prompt templates, taking 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 during finetuning on the dataset, so a fairer comparison with it is done instead on a different question answering task in §4.6.

These models are scored using the computed area under the precision-recall curve, or *AUC*. Li et al. (2022a) introduce AUC_{norm} (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-classifier baseline for the dataset, so a perfect classifier would score $AUC_n = 100\%$, while a random classifier would score 0%. Thus, AUC_n is highly discerning compared to *AUC*, which can inflate performance when there is a high random baseline. In this work, the high 50% random classifier baseline means that AUC_n scores are systematically much lower than the original *AUC* they are derived from.

4.5 Experiment 1: Entailment Detection

Two parallel investigations are run, reproduced on both Levy/Holt and ANT. (1) The unsupervised smoothing method of finding the K-Nearest Neighbors using LM embeddings is applied to augment the *Premise* of each test entailment, generating K new target premise predicates. Separately, (2) The *Hypothesis* of each test entailment is smoothed using the same method.

4.5.1 Results

A comparison of the performances of P- vs. H-smoothing of the CTX EG is shown in Figure 4.3, using the optimal $K_{prem} = 4$ and $K_{hyp} = 2$. Further, a comparison of P-smoothing performance with both the CTX and GBL EGs are shown in Figure 4.4. For both investigations, different values of the hyperparameter K were tried, for $K \in \{2, 3, 4\}$. $K_{prem} = 4$ and $K_{hyp} = 2$ are shown due to having the highest *AUC* values for P- and H-smoothing, respectively.

A few trends about hyperparameters are noticable. (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!).

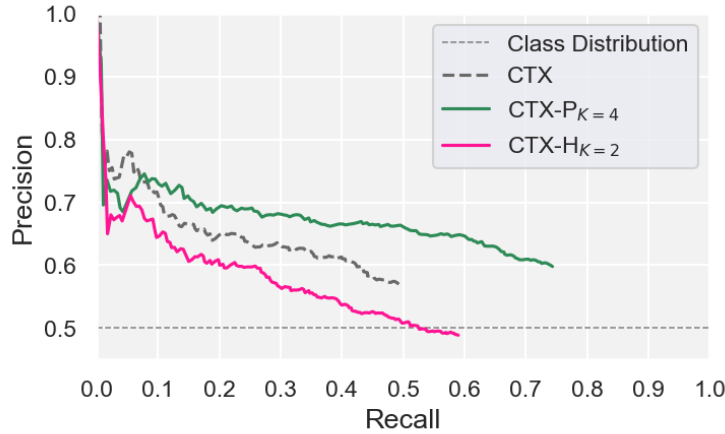


Figure 4.3: LM smoothing of EGs on the ANT dataset. Comparing P- and H-smoothing of the CTX model. Different values of K are tried in the choices $\{2, 3, 4\}$, and shown here are the best $K_{prem} = 4$ and $K_{hyp} = 2$. P-smoothing with an LM is shown to improve the CTX graph, while H-smoothing with an LM deteriorates performance.

In Figure 4.4 P-smoothing is shown in particular on the CTX graph vs. the GBL graph. For all models (best K selected) on both datasets, summary statistics are shown in Table 4.3, 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 4.4.

As predicted, the 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. Notably, P-smoothing with an LM 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 CTX on the ANT dataset, this smoothing method increases maximum recall by 25.1 absolute percentage points to 74.3% while increasing average precision from 65.66% to 67.47%. On Levy/Holt the maximum recall is increased by 16.3 absolute percentage points 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 random chance (50% precision).

Also notable is 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

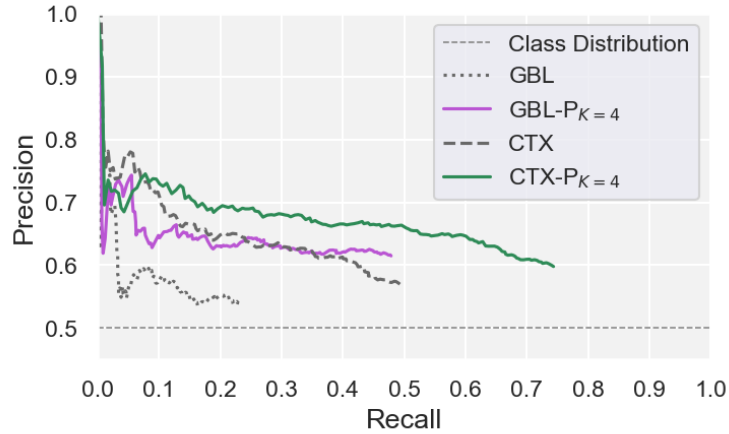


Figure 4.4: LM smoothing of EGs on the ANT dataset. Comparing optimal P-smoothing of the CTX model with that of the older GBL model. CTX and GBL contain the same vertices, but CTX improves on the edges in GBL, so this comparison shows that: (1) P-smoothing is an effective addition which can be applied across existing EGs, and (2) improvements to vertex sparsity are complementary to improvements in edge sparsity.

different aspects, which can be applied together.

4.6 Experiment 2: Boolean Question Answering

LM smoothing is now demonstrated in application on an applied task. A modified version of the Boolean Question Answering task from Chapter 3 is used to demonstrate, in which models answer true/false questions about entities mentioned in news articles from multiple sources. These questions are chosen to be adversarial to simple similarity baselines, and EGs have proven useful by using directional reasoning.

4.6.1 Boolean Open QA Dataset

The Boolean Open QA (BoOQA) dataset of Li et al. (2022a) used in this chapter is derived from the Multivalent QA task in McKenna et al. (2021), the work presented in chapter 3. This dataset is constructed using a similar process: 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

Model	ANT		Levy/Holt	
	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 4.3: P- and H-smoothing, compared to unsmoothed models. P-Smoothing with an LM is shown to improve AUC_n and AP in both CTX and GBL models.

Predicate Missing from EG	Nearest Neighbors by Embedding Distance
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 4.4: Sample of CTX outputs on ANT. A target PREDICATE(:type1, :type2) missing from CTX yields K=2 closest CTX predicates in LM embedding space.

entails the question by means of its relation, the question can be labeled True, otherwise False. As before, BoOQA also contains false questions derived from true ones using hyponyms from WordNet, so models must decide carefully what is supported by evidence and what isn't.

The task version introduced by Li et al. (2022a) makes improvements over that defined in Chapter 3 in several ways in order to strengthen the difficulty of the dataset.

4.6.1.1 Improvements over the Multivalent QA Dataset

First, the quality of generated negatives is improved. These negatives are now resolved to their most likely sense in WordNet (Fellbaum, 1998) using an introduced step of contextual word sense disambiguation, instead of simply using the most common word sense. This means that generated negatives will be more topically consistent with their respective contexts, making them harder to discriminate from positives on the basis of

similarity. Further, not only is the generated hyponym negative required to be absent from the current context window, but all of its WordNet synset siblings must be, as well, improving the confidence that generated negatives will not accidentally be true given the context.

Second, the robustness of the dataset is improved overall by making negatives less easily distinguishable from selected positives. To do this, the minimum corpus occurrence frequency of negatives is required to be similar to that of its corresponding positive.

Li et al. (2022a) measure these improvements using a hypothesis-only model, a modified version of Schmitt and Schütze (2021), which trains on a held-out portion of the same dataset. In theory, this model should not be able to perform this task, since it is not shown any premise statements and is asked to classify only the hypothesis as True or False. Thus, it would ideally achieve 0% AUC_{norm} on the test portion of the dataset. Training and testing this model on separate splits of the McKenna et al. (2021) dataset presented in Chapter 3 yields an AUC_{norm} of 78.3%, and doing so on the BoOQA dataset of Li et al. (2022a) yields an improved AUC_{norm} of 51.0%. This indicates that although it is more robust, some artifacts remain in the dataset after the improvement is made.

However, it must be stressed that this comparison is useful for estimating the quality of the dataset for training purposes, and does not indicate that the datasets are unfit for all use cases. The high AUC_{norm} scores are obtained by finetuning on only the hypotheses in a held-out portion of the data, so the model specifically attempts to learn any contained artifacts. Thus, using these datasets purely for evaluation and not for training does not risk learning these artifacts.

4.6.2 QA in the Natural Distribution of Contexts

This work aims to address vertex sparsity in a realistic setting, so the original entity restriction of Li et al. (2022a) and McKenna et al. (2021) which avoids the problem of vertex sparsity in models, is relaxed in order to achieve a more natural task. Previously, questions are sampled only for frequently-mentioned entities, which always have many context statements to decide from, and thus vertex sparsity is minimized as a problem since models can afford to miss a few context statements if there are many others available to try. In this work, the challenge is increased by sampling from corpus entities regardless of popularity. This creates a more natural distribution of questions

Context Size	Samples	CTX	CTX-P	CTX-H	S&S
[2, 5)	56,390	20.05	20.66	19.07	17.00
[5, 10)	56,425	29.13	29.17	29.01	23.05
[10, 15)	54,778	32.32	32.31	32.25	24.98
15+	54,926	36.58	36.57	36.51	26.13
All Questions	56,494	21.26	21.74	20.64	16.99

Table 4.5: Effect of P- and H-smoothing vs. baseline CTX and S&S across context sizes. AUC_n is reported.

in which some questions have many context statements to decide from, and others have much fewer. In these sparse situations, the cost of missing a predicate is much higher.

4.6.3 Results

Results on the natural distribution corroborate the earlier tests: P-smoothing improves AUC_n from 21.26% to 21.74% in “All Questions” sampled from the natural distribution, while H-smoothing worsens to 20.64% (as in §4.4, AUC_n is systematically lower than AUC). Smoothed EGs also outperform [Schmitt and Schütze \(2021\)](#), the most direct competition which uses a tractable-size LM. Despite facility to encode any predicate, it lacks directional precision useful for this task.

To demonstrate when smoothing an EG is helpful, the effect on different *context size bands* is further analyzed. For each question, the number of context sentences available to answer it are counted; in dataset generation, questions are bucketed into count bands of [2, 5), [5, 10), [10, 15), 15+. Approximately 55,000 questions are sampled for each context size. Within these bands an unsmoothed model is compared with both P-smoothing and H-smoothing, reported in Table 4.5.

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 $f \geq 15$, but 32.7% for $f < 5$.

4.7 Experiment 3: Controlled Smoothing of P and H with WordNet Relations

Language Model P-smoothing is shown to work well in the previous experiments, but not H-smoothing. Controlled experiments are now shown using WordNet relations (Fellbaum, 1998) to confirm that this is due to semantic generalization (in line with the theory in §4.3.1). It is shown by constructing a transitive chain using WordNet hyponyms that Hypothesis smoothing is possible in principle, without a claim that it provides a practical alternative to a Language Model.

4.7.1 Controlled Search with WordNet

Experiment 1 is re-run by smoothing the CTX and GBL models on the ANT dataset. However, the target premise or hypothesis is now approximated without the LM. Instead, replacements are generated using two WordNet relations. Of note, WordNet was partly used in ANT’s construction, so this result explains the Language Model effect, rather than offering a practical model or claim to any dataset high score.

In this test, specific WordNet lexical relations are chosen as instances of entailment, then used to generate smoothing predictions from the WordNet database. In this work, **hyponymy** is used for specialization and **hypernymy** for generalization, and both relations are compared for use with both P- and H-smoothing. To illustrate, if smoothing by generalizing, given a predicate *elect*, WordNet hypernyms are retrieved such as *choose*.

This is done by querying WordNet for relations of the predicate head word. Results are used from the first word sense, replacing the query word. E.g., from the predicate (receive.2, receive.from.2) which is missing from the EG, the WN query *hyponym*(“receive”) \Rightarrow “inherit” is used to generate a smoothed query predicate (inherit.2, inherit.from.2).

4.7.2 Results

Results are shown in Figure 4.5. Importantly, from these plots a switch in performance is observed between the application of hypernyms and hyponyms when used for P- and H-smoothing on CTX, and the same effect is shown with GBL. 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

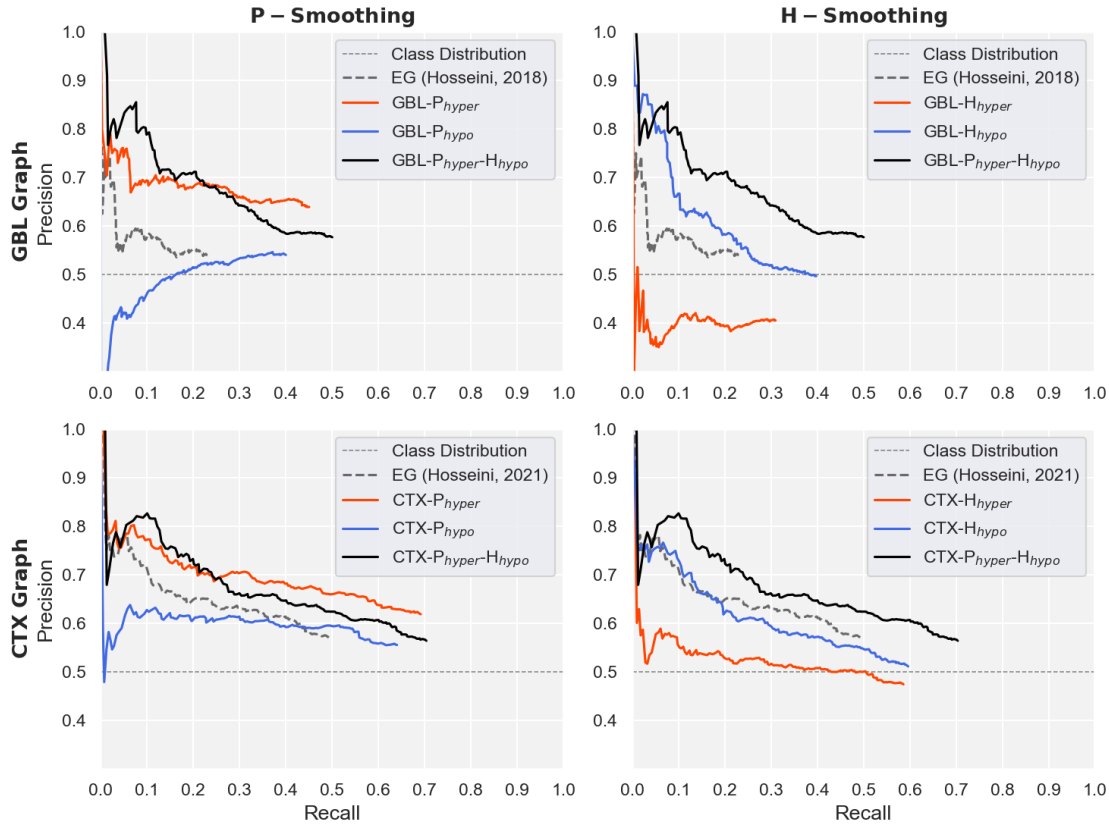


Figure 4.5: Comparison of WordNet relations used in smoothing the Premise, Hypothesis, and Premise+Hypothesis. Smoothing is applied to both GBL and CTX graphs on the ANT dataset. Hypernyms are shown to be consistently useful for P-smoothing on both GBL and CTX graphs, and hyponyms less so for H-smoothing, where they are beneficial to the weaker GBL graph, but not apparently useful for CTX. The combination of “optimal” smoothers does show improvement in both graphs in the low-recall end.

the hypothesis, the reverse is true: generalizing with hypernyms worsens performance, but specializing with hyponyms can lead to some performance gains: it improves the weaker GBL graph when applied by itself, and shows some small improvement on CTX when applied along with P-smoothing (discussed more later). Performance on Levy/Holt was also tested, on which a similar trend is observed.

These results nearly replicate the behavior of the LM-smoother in §4.4, verifying that nearest neighbor search in LM embedding space has a semantically generalizing effect suitable for P-smoothing. Table 4.4 shows examples of generalized predictions.

Finally, Figure 4.6 shows a comparison of P-smoothing between the LM (CTX- P_{LM} achieves $AUC_n = 25.86$) and WordNet (CTX- P_{hyper} achieves $AUC_n = 27.39$) on

the ANT dataset. Although WordNet performs within about 1.5% of the LM smoother in this “laboratory” experiment, the LM-smoother is preferable in real use, because it is fully automatic to learn and apply, and because it encodes an open domain of predicates, which may include out-of-vocabulary words, misspellings, etc. that WordNet cannot handle.

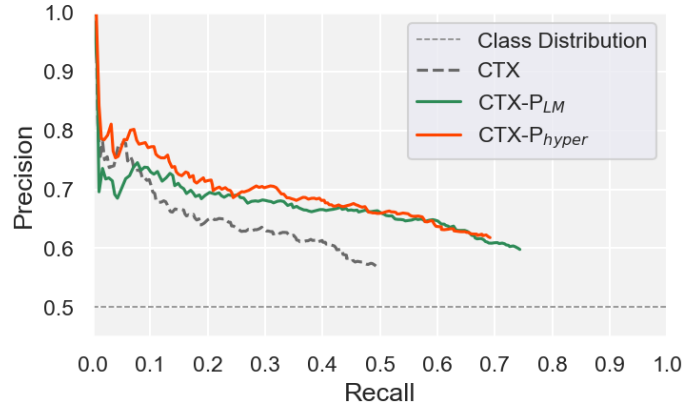


Figure 4.6: Comparison of P-smoothing methods on ANT: LM-based smoother and WordNet hypernym relations on the Entailment Graph CTX.

4.7.3 Discussion

Two phenomena of interest are observed. (1) For both CTX and GBL, precision is boosted in the low-recall 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 useful for the weaker graph GBL, but is not very useful for smoothing CTX.

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 (special-

izations) in corpora. If the hypothesis predicate h is missing from the EG (meaning it wasn't seen in training) then a derived candidate h' specialized from h will also be unlikely to occur in training. Thus, even if found in the EG h' may have few or poorly learned edges. Though it can be beneficial to precision, data sparsity makes H-smoothing fundamentally harder than P-smoothing.

4.8 Related Work

Following from this work, [Chen et al. \(2023\)](#) later approach the problem of vertex sparsity by a different means of constructing Entailment Graphs. These EGs are targeted to a specified domain, and rely on the mining of a Language Model for predicates within that domain to achieve better coverage. This method makes a step forward in increasing predicate coverage and entailment recall. On the complete Levy/Holt dataset, by generating an Entailment Graph from the Levy/Holt dev set, the method achieves 65% recall compared to 43% by the base CTX graph used in this work ([Hosseini et al., 2021](#)), a previously SOTA EG which was trained using corpora separate from the test domain. However, the method improves coverage through brute force expansion of the EG, and does not solve the fundamental problem of what to do when missing predicates at test time. Indeed, the method suffers reduced performance when switching between different test sets, showing the downsides of focusing on a specific target domain.

Another relevant area of research is in the “hubness” phenomenon in high dimensional vector spaces, especially for word vectors. It is empirically the case that under many distance metrics, some subset of the vectors in a word embedding space will tend to act as centralized hubs, featuring as a nearest neighbor for a disproportionate amount of other vectors. [Radovanović et al. \(2010\)](#) state that “the hubness phenomenon is an inherent property of data distributions in high-dimensional space under widely used assumptions, and not an artefact of a finite sample or specific properties of a particular data set.” The experiments conducted in this chapter do not screen for this effect specifically, and it is possible that hubs may play a role in the performance of the method described in §4.4.1. Though other work aims to reduce this hubness effect to improve performance in downstream applications ([Feldbauer et al., 2018](#)), this may not improve performance in these experiments. The embeddings-based approach presented in this chapter relies heavily on the skewed distribution of word vectors which naturally has the properties described in §4.3.2. The behavior of K-Nearest Neighbor search under these properties is what produces transitive chains for missing premise

predicates, and these chains are crucial to maintaining directional precision in application. In an extreme case where EG predicate vectors are completely “de-skewed” by transforming the embeddings into a uniform distribution, then the selection of nearest neighbor approximations for a missing target predicate might include predicates with no guarantee or even likelihood of constructing a transitive chain. This could be highly detrimental to method precision.

4.9 Conclusion

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

This work also demonstrates the theory of optimal smoothing by directing the search for smoothing predictions using controlled WordNet relations. These 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.

4.9.1 Limitations

This work presents 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. Nearest neighbors search within LM embedding space is biased toward returning predicates that are more semantically general, which is helpful for P-smoothing.

However, generalizing is detrimental to H-smoothing, which requires specialization. While empirical evidence of the benefit of specialization is shown 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. If a hypothesis is missing from the EG, it is already 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. It is assumed in this work that the vocabulary is sufficient, but if it is not suitable e.g. for a new target genre/domain, [Hosseini et al. \(2021\)](#) show that EG learning may be scaled up easily, which may provide a sufficiently scoped vocabulary for any application, but exploring this is left for future work.

Finally, this work is demonstrated only on the English language. However, there is no immediate indication that this method should fail for other widely spoken languages. [Li et al. \(2022b\)](#) demonstrate that learning Entailment Graphs for other languages (Chinese demonstrated) can be done using the same process as English, and the smoothing technique leverages a simple fundamental structure of Language Models which is characteristic of pretraining, as well as the natural Zipfian distribution of predicates in corpora, which is present across languages.

4.9.2 Ethical Considerations

This work is designed to extend the capabilities of Entailment Graphs, which are general-purpose structures of meaning postulates. These can be applied most readily to question answering applications, but they can also be used for other NLU or NLI tasks. As an unsupervised, corpus-based learning algorithm, EGs could be susceptible to learning biases in human beliefs present in corpora, but this algorithm is most sensitive to widely repeated statements, which may be easier to detect in data cleaning than uncommon statements. There is no immediate risk in their application in basic question answering as shown in this work, since the EGs used here were trained on published news articles ([Hosseini et al., 2021](#)) which are professionally edited to a standard. However, a malicious user could deploy any tool for language understanding such as an EG for other unethical purposes such as surveillance at scale, etc.

Chapter 5

Large Language Models and Open-Domain Predicate Inference

Though Entailment Graphs theoretically represent inferences spanning the open domain of natural language predicates, they are prevented from doing so in practice by the challenge of vertex sparsity, arising from sparsity in training data. However, Language Models are shown to be effective as a means for smoothing EGs while maintaining directional precision, demonstrating a directional signal of their own.

Large Language Models (LLMs) are recently claimed to be capable of Natural Language Inference (NLI), necessary for applied tasks like question answering and summarization. This chapter investigates the capability of LLMs to facilitate a fully open domain of predicate inference on their own, without using an Entailment Graph. A series of behavioral studies is presented on several LLM families (LLaMA, GPT-3.5, and PaLM) which probe their behavior using controlled experiments on NLI datasets. In contrast to the explicit reasoning in application of an EG, this chapter demonstrates that LLMs constitute an *approximation* of generalizing language inference. Despite promising superficial performance, two biases are established, originating from pre-training, which predict much of their behavior and represent major sources of hallucination in generative LLMs. Several LLMs are shown to perform significantly worse on NLI test samples which do not conform to these biases than those which do, and these are offered as valuable controls for future LLM evaluation.

5.1 Introduction

Large Language Models (LLMs) such as LLaMA, GPT-3/4, PaLM, and more (Touvron et al., 2023; Brown et al., 2020; Chowdhery et al., 2022), have been trusted by many to perform language understanding in downstream tasks such as summarization, question answering, and fact verification, among others (Zhang et al., 2023). However, due to the large-scale nature of LLM training on vast, often proprietary data, and the inherent opacity of LLM parameters, it is difficult to explain their behavior when answering user queries and the corresponding risks in terms of bias and robustness. In particular, one LLM behavior poses a significant challenge: “hallucination,” the phenomenon in which LLMs provide information which is incorrect or inappropriate, presented as fact.

This work investigates false positive hallucination through the mechanisms used by LLMs on the task of natural language inference. This is also called *textual entailment*, a basic component of language understanding which is critical in real language tasks which require more than mere approximation of memorized training data. For example, in retrieval-aided question-answering from documents that entail but do not state a direct answer to the question; or multi-document summarization, where models must detect redundancy of statements in one document that are entailed by others.

In this work, the broader context of NLI is examined, but focus remains on directional entailments, which hold in one direction, but not both. For example, *Arsenal defeats Man United* entails *Arsenal plays Man United* but the reverse is not true: *Arsenal plays Man United* does not entail *Arsenal defeats Man United*. Inferring directional entailment is more difficult than detecting symmetric paraphrase, so it more deeply probes understanding. In this setting, false positive judgements of Entail by a model may be understood as hallucinations.

The approach used is a behavioral study of prompted LLM decision-making across several LLM families (LLaMA, GPT-3.5, and PaLM). Existing NLI datasets are altered in targeted ways while the changes in model predictions are measured. Two sources of LLM performance on the NLI task are demonstrated, which are offered as explanations of general false positive hallucination: (1) LLM bias toward affirming entailment when the hypothesis may be attested in the training text, including reliance on named entity identifiers; and (2) a corpus-frequency bias, affirming entailments with premises less frequent than hypotheses.

Earlier chapters evaluate models on an extrinsic, question-answering task which demonstrates directional reasoning, however this chapter focuses on evaluation of

simpler NLI datasets. This is intentional, to constrain the LLMs to minimal scenarios which are easily controlled. Li et al. (2022a) demonstrate that extrinsic tasks like question-answering may easily (though accidentally) introduce artifacts correlated with sample entailment labels, which LLMs may take advantage of without learning the task of entailment. To minimize this risk, experiments are conducted with no dataset fine-tuning (so no additional artifacts can be learned), and on these simpler NLI datasets which, through controlled modification, are solvable only with linguistic reasoning. The simplicity and directness of this test is ideal for exploring the claims.

This study establishes that the two found biases originate from the LLM pretraining objective, in which statistical modeling of the natural distribution of human-generated text leads to (at the level of sentences) memorizing individual statements, and (at the level of corpora) learning typical patterns of usage. Though superficially performant, these experiments show that even powerful LLMs still use unsatisfactory tools instead of robust reasoning.

Three contributions are presented in this chapter: the demonstrations of both factors and an analysis of their impact.

1. In a prompting scenario, LLMs respond to entailment samples according to an *attestation bias*, affirming entailments more readily if the hypothesis is attested by the pretraining text. LLaMA-65B, GPT-3.5, and PaLM are respectively found to be 1.9, 2.2, and 2.0 times more likely to wrongly predict Entail when the model already asserts the hypothesis is attested, compared to when not. Further, LLMs recall from their propositional memory using named entities as identifying “indices,” even though they are irrelevant to the logic of the predicate inference task.
2. LLMs also rely on a simple corpus-statistic bias using relative term-frequencies, especially when propositional memory is not available. The three LLMs are 1.6, 1.8 and 2.0 times more likely to wrongly affirm entailments if the premise has lower term frequency than the hypothesis, than when not.
3. For the NLI test subsets consistent with these factors, LLM scores are misleadingly high; for NLI subsets adversarial to them, LLM performance degrades severely. It is shown that when labels go against the *attestation bias*, LLMs can be poor or even near-random classifiers; for the *relative frequency bias*, a similar substantial performance decrease is shown across all LLMs.

5.2 Background

Addressing task robustness, [Poliak et al. \(2018\)](#) found a range of NLI datasets to contain artifacts which are learnable by supervised models trained on only the hypothesis. In this work, a similar hypothesis-only test is used with LLMs, however it is used to probe model memory without any training involved. The attestation bias demonstrates inherent model bias, versus the dataset bias exposed by [Poliak et al.](#)

For supervised neural models on the NLI task, [Talman and Chatzikyriakidis \(2019\)](#) observed generalization failure when models are transferred between NLI datasets, even when they are formatted in the same way. On smaller Language Models such as RoBERTa ([Liu et al., 2019](#); 355M parameters), [Li et al. \(2022a\)](#) also observed a reliance on dataset artifacts when finetuned specifically for *directional* predicate inference. This work now studies the behavior of much larger LMs, which have demonstrated more robust performance across NLP tasks.

Recent work has also explored LLM memorization and generalization. [Carlini et al. \(2023\)](#) establish that LLMs are able to memorize more data than small LMs, whereas [Tirumala et al. \(2022\)](#) further demonstrate that LLMs pay special attention early in training to numbers and nouns, which act as unique identifiers for individual training sentences. Memories are also shown to be used in language inference, relating to specific named entities. While [Weller et al. \(2023\)](#) and [Kandpal et al. \(2023\)](#) find that entity frequency in training data is correlated with performance in factual recall about them, this work finds that entity frequency is *anti*-correlated with hypothetical generalization performance (§5.6).

5.3 Experimental Design

Behavioral experiments on LLMs are designed by modifying NLI datasets with rigorous controls, changing targeted informational aspects of these datasets. These changes illicit behavioral differences in major LLMs due to propositional-memory effects in §5.5 and §5.6, and corpus frequency in §5.7. Finally, the impact on real performance is shown in §5.8.

5.3.1 Two Biases in Inference Predictions

The major claim of this work is that the pretraining objective to fit the distribution of natural text leads to biases in LLM generations. Two such biases are explored, in the

narrow scope of individual sentences and at the wider scope of corpora.

5.3.1.1 The Attestation Bias

The over-reliance of an LLM on its propositional memory about a query statement is dubbed *attestation bias*. It is claimed that when a statement is likely to be attested in some way by an LLM’s training data, the model is more likely to affirm it as a conclusion in NLI tasks, regardless of any premise it is presented with. In this work, the attestation of a sample is measured by prompting the LLM, asking simply if the hypothesis in question is true, false, or unknown.¹ Model predictions of attestation are denoted with Λ .

A model with such attestation bias will appear to perform well on dataset samples with entailment labels that happen to align with the bias. For example, the pair of samples below from the Levy/Holt dev set are consistent with the bias, as reported by an LLM itself (Λ_{LLM}), because the sample labels may be predicted using hypothesis attestation alone, without considering the premise.

Sample 1

PREMISE: *Geysers are common to New Zealand*

HYPOTHESIS: *Geysers are found in New Zealand*

LABEL: Entail

Λ_{LLM} : hypothesis Attested

Sample 2

PREMISE: *Geysers are found in New Zealand*

HYPOTHESIS: *Geysers are common to New Zealand*

LABEL: No-Entail

Λ_{LLM} : hypothesis Not-Attested

As discussed in §5.2, inspiration is drawn from the hypothesis-only baseline of [Poliak et al. \(2018\)](#), but this work instead probes model memory without any training, exploring inherent model bias. Prompt generation is described in detail in §5.4.2, with an example in Table 5.3.

[Dasgupta et al. \(2022\)](#) show a similar effect in LLMs on abstract reasoning tests, related to sentential content, and remark that human tendencies are similar. In contrast,

¹Alternatively, LLM perplexity for a statement could be used to identify statements that are not attested by the training text; however, perplexity scores are not always available, e.g. with GPT-3.

this work examines the *risks* of propositional memory on more realistic inference tasks.

5.3.1.2 The Relative Frequency Bias

The *relative frequency bias* is the use of a simple rule for deciding entailment, calculable from corpus statistics. Entailment is affirmed if, ignoring named entities, the eventuality in premise P is less frequent in training than that in hypothesis H .

This bias is reflected in natural text: it is known that nouns follow a trend of becoming more specific as corpus-frequency decreases (Rosch et al., 1976; Caraballo and Charniak, 1999) and the same is observed for predicates (McKenna et al., 2023b). Very infrequent predicates tend to be very specific (e.g. *perambulate*, *hike*) compared to very frequent predicates which tend to be more semantically general (e.g. *walk*, *move*). A specific predicate may entail a general one (e.g. *hike* entails *walk*) but the reverse is not possible (*walk* does not entail *hike*). Thus, relative frequency can often indicate the direction of entailment; Language Models are known to be sensitive to frequency, so it follows that this bias may inherently be used in predictions. However, the relative frequency effect is an artifact of natural text, so the use of it as a signal of entailment is merely a bias with no direct relationship to meaning.

Test samples are labeled for agreement with this bias separately from models. Since LLM pre-train corpora are impractically large and/or proprietary, Google N-grams² is used instead as a proxy of the natural distribution of text, and thus the distributions of these corpora. Predicate frequencies are estimated by an average between the years 1950-2019, and compared between P and H . For robust comparison, generic eventualities are extracted from test sentences by masking any extracted entities and lemmatizing phrases; further, the problems of distributional noise and sparsity are addressed by requiring a wide margin of difference between P and H frequency estimates. Frequency decisions are denoted by Φ .

A model with such relative frequency bias will appear to perform well on dataset samples with entailment labels that happen to align with the bias. For example, the pair of samples below from the Levy/Holt dev set are consistent with the bias, as reported by the frequency estimate (Φ), because the sample labels may be predicted using relative frequency alone, without deeper consideration of entailment.

²<https://books.google.com/ngrams>

Sample 3PREMISE: *Whiskey consists chiefly of alcohol*HYPOTHESIS: *Whiskey contains alcohol*

LABEL: Entail

 Φ : Yes: $\text{freq}(\text{consists chiefly of}) < \text{freq}(\text{contains})$ **Sample 4**PREMISE: *Whiskey contains alcohol*HYPOTHESIS: *Whiskey consists chiefly of alcohol*

LABEL: No-Entail

 Φ : No: $\text{freq}(\text{contains}) > \text{freq}(\text{consists chiefly of})$ **5.3.2 Datasets**

Levy/Holt As described in Chapter 2, this dataset consists of premise-hypothesis pairs, with a task formatted: “Given [premise P], is it true that [hypothesis H]?” (Levy and Dagan, 2016; Holt, 2018). Each P - and H -statement has the property of containing one predicate with two entity arguments, (where the same entities appear in both P and H) as shown in Table 5.1. This targeted dataset is ideal for precisely measuring model understanding of predicates, because entailment between statements is decidable purely on the basis of the predicates and their attributes.

This work studies the challenging directional subset, where entailments hold in one direction but *not* both. The directional portion of the Levy/Holt dataset contains 630 entries in the dev set, and 1784 entries in the test set. Both have a 50%/50% class distribution between Entail and No-Entail labels, since all samples are directional and tested in both directions ($a \models b$ and the reverse, $b \not\models a$).

RTE-1 This dataset is one of the original and most difficult tests of NLI (Dagan et al., 2006). It is not purely directional on the basis of predicates or consistently structured like Levy/Holt, so it is left out of the behavioral experiments. However, RTE-1 is a widely understood dataset, and in this work it is used to demonstrate the impact of the two biases in general NLI situations, in §5.8.

The original RTE-1 dataset contains 567 entries in the dev set, and 800 entries in the test set. It has a 50%/50% class distribution between Entail and No-Entail labels (for RTE-1 dev set, the numbers of entries in the two label classes differ by 1).

Exclusions NLI datasets are excluded if they intentionally test knowledge of the world, since the aim is to test LLMs on their capability to reason purely about the semantics of natural language predicates without relying on memorized facts. Datasets such as MMLU (Hendrycks et al., 2021), Natural Questions (Kwiatkowski et al., 2019), OpenBookQA (Mihaylov et al., 2018) etc. are specifically avoided.

5.3.3 Dataset Transformations

Throughout this work, **the standard inference task** denoted I refers to the original NLI datasets, in which entailment is determinable by using general language inference on sentences. In Levy/Holt, it is determinable just by predicates.

Three dataset transformations are defined for use in this study; the change in model responses is observed as targeted information is altered in each transformation. These transformations include randomized premise predicates $I_{RandPrem}$, and two argument-transformations: generic arguments I^{GenArg} , and type-constrained randomized arguments $I^{RandArg}$.

Transformations involve first identifying the types of entities in statements, in order to constrain entity or predicate replacements. The process is similar to that in the previous two chapters. Initially, an entity linker is used (Nguyen et al., 2014) which identifies the Freebase ID (Bollacker et al., 2008) of each sentence entity. From the Freebase entry is obtained the entity type, classified as one of the 48 FIGER types (Ling and Weld, 2012), such as “person,” “location,” etc. An additional default type “thing” is assigned in failure cases.

5.3.3.1 The Random Premise Task $I_{RandPrem}$

$I_{RandPrem}$ is a test of model reliance on propositional memory: dataset alterations prevent all true entailments without modifying hypotheses. Thus, a good model of entailment should recognize that the premise does not entail the hypothesis, and respond by predicting No-Entail, even if the hypothesis is attested by training data.

This task replaces the original premise predicate with a random predicate, while maintaining the same entity arguments. This manipulation produces a dataset in which all samples are labeled No-Entail, since two randomly paired predicates are very unlikely to be related by entailment. Hence, any positive decision by the model is a false positive hallucination.

To maintain naturalness and grammaticality, a new predicate is constrained to have

Task	Label	Dev Sample Query: [premise] \Rightarrow [hypothesis]
I	Entail	George Bush <u>was Governor of</u> Texas \Rightarrow George Bush <u>is a politician from</u> Texas
$I_{RandPrem}$	No-Entail	George Bush <u>resided in</u> Texas \Rightarrow George Bush <u>is a politician from</u> Texas

Table 5.1: From the original dataset task (I) is derived the Random Premise task ($I_{RandPrem}$), respecting type-constraints. A random premise predicate is highly unlikely to entail the hypothesis, so all labels are No-Entail.

argument slots of the same types as the original premise. For example, “[medicine] is indicated for patients with [disease]” is swapped for “[medicine] does not cure [disease]”. Candidates are sourced from dev set premises satisfying the target type-constraints, and sampled uniform randomly. The original entities are mapped to their respective slots in the new premise. Examples are shown in Table 5.1.

5.3.3.2 The Generic Argument Task I^{GenArg}

This task replaces original entities with unique FIGER-typed identifiers, e.g. “location X” and “food Y”. In using these generic identifiers to mask the identities of entities, this test is designed to remove extraneous information while maintaining the same entailment label, as a baseline control setting. Unique identifiers are appended (e.g. “X”, “Y”) to allow tracking of entity slots across the premise and the hypothesis. A good model of entailment should not significantly change its predictions between I and I^{GenArg} , since predicates and their semantics remain unchanged, and can still be clearly disambiguated using entity types. Examples are shown in Table 5.2.

5.3.3.3 The Random Argument Task $I^{RandArg}$

This task replaces original entities with other real, random entities of the same FIGER-type. Like I^{GenArg} , this test is designed to create novel strings by modifying statements without changing entailment labels. But this is a test of model sensitivity to added extraneous information. Examples are shown in Table 5.2.

As with I^{GenArg} , entity type constraints are used to ensure polysemous predicates maintain the same sense. For example, a different sense of *run* is used in “[person] runs [organization]” vs. “[person] runs [software]”, but between different entities of the same type, the same sense is maintained, so the exact entity IDs do not affect entailment labels. New entities are sourced from NewsCrawl (Barrault et al., 2019), a decade-long span of multi-source news text, in which entities are typed as above.

Task	Label	Dev Sample Query: [premise] \Rightarrow [hypothesis]
I	(Entail)	<u>India</u> exports tons of <u>rice</u> \Rightarrow <u>India</u> exports <u>rice</u>
I^{GenArg}	(Entail)	<u>location X</u> exports tons of <u>food Y</u> \Rightarrow <u>location X</u> exports <u>food Y</u>
$I^{RandArg\downarrow}$	(Entail)	<u>Sloterdijk</u> exports tons of <u>oatmeal cookies</u> \Rightarrow <u>Sloterdijk</u> exports <u>oatmeal cookies</u>
$I^{RandArg\uparrow}$	(Entail)	<u>Helsinki</u> exports tons of <u>Granny Smith</u> \Rightarrow <u>Helsinki</u> exports <u>Granny Smith</u>

Table 5.2: An original dev sample (I) is transformed by insertion of entity types (I^{GenArg}); by real entities sampled from the 5% least frequent in NewsCrawl ($I^{RandArg\downarrow}$); and also from the 5% most frequent ($I^{RandArg\uparrow}$).

The new entities are drawn uniform randomly from the 5% least common entities in NewsCrawl ($I^{RandArg\downarrow}$), and the 5% most common ($I^{RandArg\uparrow}$).

5.4 Methods: Querying Models with Prompts

The methodology for model selection, prompt development, and model scoring are now described.

5.4.1 Model Selection

LLaMA A recent LLM model family which rivals or surpasses GPT-3 performance while being open to scientific study. LLaMA provides a range of model sizes, and this work tests the largest **LLaMA-65B** model. LLaMA is not fine-tuned; while there have been fine-tuned variants (Taori et al., 2023; Chiang et al., 2023), they were not found to be more competent than LLaMA-65B on the task, so they are left out.

GPT-3 Series Though closed to deep scientific review, these are a widely-used comparison due to their performance, and have been reasonably well studied (Brown et al., 2020). This work evaluates on **text-davinci-003 (GPT-3.5)**, as it is the largest, and has undergone instruction- and RLHF-finetuning, enabling interesting comparisons.

PaLM One of the largest available LLM families, this work tests with the largest 540 billion parameter model, which often claims state-of-the-art on evaluation datasets (Chowdhery et al., 2022). As it is only pretrained, this model serves as a further comparison point to LLaMA.

Later GPT models (like text-davinci-003 in this work) have been pre-trained and fine-tuned, while base LLaMA and PaLM have only undergone pre-training, so their contrast indicates what stage of training is responsible for the phenomena studied. The aim of this work is not to judge which LLM is superior, but to evaluate this class of models’ capability for NLI, and show the common sources of hallucination they share.

Current open models are also omitted if they are superseded in performance by LLaMA (e.g. OPT, GPT-J, etc.), as well as products that are closed to scientific review (e.g. GPT-4, Bard, etc.).

5.4.2 Prompt Design

5.4.2.1 Formatting

Each test sample is formatted for the model by insertion of the premise and hypothesis into a prompt template, which is used to query the model in natural language. Following this, a three-way answer choice is appended to the sample: “(A) Entailment, (B) Neutral, (C) Contradiction”, following the typical format in NLI (Bowman et al., 2015). Models generate a textual response conditioned on this input, and general patterns in model responses are analyzed between test conditions.

5.4.2.2 Template Features

Besides the test sample premise and hypothesis, in prompt-based interactions with the LLMs, several kinds of context information may be added to aid models in producing accurate and robust predictions. Three design choices in particular are examined in the prompt engineering phase of this work: varying the language of prompt templates, adding in-context examples, and chain-of-thought reasoning.

Template Candidates A prompt template is a natural language string containing slots, into which information from a dataset sample is inserted, creating a uniformly-formatted string query for an LLM. These are known to have a direct and sometimes decisive impact on LLM behavior due to their sensitivity to wording. As such, a range of clear and concise templates are carefully selected as promising candidates. Each template is ranked according to AUC score by evaluating with the dev portion of each dataset. The final template is selected which achieves the best score, which is then used on the test portion. The set of candidate templates includes 3 novel templates:

1. If [PREMISE], then [HYPOTHESIS].
2. [PREMISE], so [HYPOTHESIS].
3. [PREMISE] entails [HYPOTHESIS].

Also considered are the 5 prompt templates used in entailment work with small LMs (Schmitt and Schütze, 2021) and later in larger LMs (Webson and Pavlick, 2022):

4. [PREMISE], which means that [HYPOTHESIS].
5. [HYPOTHESIS], because [PREMISE].
6. It is not the case that [HYPOTHESIS], let alone that [PREMISE].
7. [HYPOTHESIS]_{NEG}, which means that [PREMISE]_{NEG}.
8. [PREMISE]_{NEG}, because [HYPOTHESIS]_{NEG}.

In preliminary experiments with GPT-3.5, it is observed that the model is not responsive to the 3 contrapositive prompts from Schmitt and Schütze (2021) (colored gray), performing at random. Prompt number 5 also consistently underperforms the other 4 templates, so this work uses the remaining 4 templates (numbers 1, 2, 3, 4) as the final candidate set.

In-Context Examples Brown et al. (2020) demonstrated that it is useful to insert task examples annotated by the modeler into the prompt. This can significantly improve the accuracy of LLM generations for a task. Use of such examples is called “in-context learning” because the model conditions its response on the actual task as well as the additional demonstrative examples. However, the term “learning” is used differently here since no parameter weights are updated using this technique; thus, any in-context examples must be prepended to every input given to the LLM.

On the other hand, Ouyang et al. (2022) has suggested that instruction-tuned LLMs are also capable of performing tasks in zero-shot, without exposure to any in-context examples. Zero-shot settings (no in-context learning by example) are compared with few-shot settings (several in-context examples provided with the query sample) for this study, in preliminary experiments with LLaMA and GPT-3.5 on the Levy/Holt directional dev set. Following Touvron et al. (2023), for zero-shot, a textual description of the task is prepended to each test sample; for few-shot, a minimal 4 examples with

explanations is prepended. Instantiated prompts in the two settings are demonstrated in Table 5.3 and Table 5.4.

For the two pretrained-only LLMs, LLaMA and PaLM, it is found that zero-shot performance on the Levy/Holt directional dev set is near-random, at 56.6% and 61.5% AUC respectively (random is 50%); with 4 in-context examples, the models begin to exhibit non-trivial behavior, with 65.0% and 80.2% AUC, respectively. This is not surprising, since pre-trained LLMs without instruction fine-tuning should not be expected to perform complex tasks zero-shot. For GPT-3.5, the performance is still much lower in zero-shot, at 64.5%, compared to 74.6% in few-shot.

Ideally LLMs will have zero-shot natural language inference ability readily available for downstream tasks. However, in order to evoke a positive response from the models using a minimal stimulus, the primary experiments are conducted in the few-shot setting throughout, in order to better explore the abilities of these LLMs.

Chain-of-Thought Reasoning Further, [Wei et al. \(2022b\)](#) has demonstrated that including chain-of-thought explanation, namely step-by-step explanations, in the in-context examples, helps LLMs perform reasoning tasks. The NLI task is not a multi-step reasoning task (it is only a single step of inferring one predicate given another), but a brief explanation is written into the prompt for each hand-annotated example of why the entailment does or does not hold.

Entailment Prompt: Zero-shot (No Examples)

Please check the entailments between the following statements.

If kanamycin kills infections, then kanamycin is useful in infections.

- A) Entailment
- B) Neutral
- C) Contradiction

Answer:

Entailment Prompt: Few-shot (With Annotated examples)

If Google bought Youtube, then Google owns Youtube.

- A) Entailment
- B) Neutral
- C) Contradiction

Answer: A) Entailment. Owning is a consequence of buying.

If Google owns Youtube, then Google bought Youtube.

- A) Entailment
- B) Neutral
- C) Contradiction

Answer: B) Neutral. Owning does not imply buying, the ownership may come from other means.

If John went to the mall, then John drove to the mall.

- A) Entailment
- B) Neutral
- C) Contradiction

Answer: B) Neutral. John may have gone to the mall by other means.

If John drove to the mall, then John went to the mall.

- A) Entailment
- B) Neutral
- C) Contradiction

Answer: A) Entailment. Driving is a means of going to the mall.

If ephedrine is widely used in medicine, then ephedrine is used in medicine.

- A) Entailment
- B) Neutral
- C) Contradiction

Answer:

Table 5.3: Example instantiated prompts in Zero-shot / Few-shot settings, for the test entry “PREMISE: [ephedrine is widely used in medicine], HYPOTHESIS: [ephedrine is used in medicine]”. The few-shot prompts in part B are used throughout the main experiments in this work.

Attestation Prompt: Few-shot (With Annotated examples)
<p>Google bought Youtube.</p> <p>A) True</p> <p>B) Unknown</p> <p>C) False</p> <p>Answer: A) True.</p>
<p>Yoshua Bengio likes oak trees.</p> <p>A) True</p> <p>B) Unknown</p> <p>C) False</p> <p>Answer: B) Unknown.</p>
<p>The sun rises from the west.</p> <p>A) True</p> <p>B) Unknown</p> <p>C) False</p> <p>Answer: C) False.</p>
<p>ephedrine is used in medicine.</p> <p>A) True</p> <p>B) Unknown</p> <p>C) False</p> <p>Answer:</p>

Table 5.4: Example instantiated prompt querying for the attestedness of the test entry “HYPOTHESIS: [ephedrine is used in medicine]”. The hypothesis-only test Λ measures the model’s prior exposure to a sample hypothesis as described in §5.3.1.

5.4.2.3 Tuning and Selecting Templates

For evaluation on the test datasets, a prompt template is selected from the 4 candidate templates which scores the highest AUC on each respective dev set: for testing on Levy/Holt, performance on the Levy/Holt dev set is used for ranking templates, and for testing on RTE-1, performance on the RTE-1 dev set is used.

As discussed, each LLM was initially tested in zero-shot, but they exhibited severely degraded, even near-random performance. The main experiments are thus formatted with few-shot examples, with four hand-annotate examples in the style of the template, also with added explanations about why the given answer is correct for each

example. These examples are prepended before the query. The goal of this work is to study model behavior as conditions change, not to maximize the score on any particular dataset. Therefore, the minimal four-example setup is used, which is found to evoke positive responses from all three LLMs on each dev set, across most templates.

This work examines the behavior and performance of three major LLM families on two NLI datasets: Levy/Holt and RTE-1. In Table 5.5, dev set performances are reported on the best prompt template used for each model on each dataset. Note that no training is involved in this work, and prompt template selection is the only hyperparameter tuned on the dev sets.³

Model	Task	Levy/Holt		RTE-1	
		Best Tplt. ID	Dev Set AUC_{norm}	Best Tplt. ID	Dev Set AUC_{norm}
LLaMA	I	#4	30.0	#3	62.5
	I^{GenArg}	#1	34.6	#3	52.3
	$I^{RandArg\downarrow}$	#1	31.8	#1	51.3
	$I^{RandArg\uparrow}$	#1	26.3	#3	43.8
GPT-3.5	I	#1	49.2	#3	74.8
	I^{GenArg}	#1	39.8	#3	64.8
	$I^{RandArg\downarrow}$	#1	43.4	#3	63.6
	$I^{RandArg\uparrow}$	#1	34.2	#3	66.0
PaLM	I	#1	60.9	#4	84.5
	I^{GenArg}	#1	48.1	#4	79.4
	$I^{RandArg\downarrow}$	#1	43.6	#3	79.8
	$I^{RandArg\uparrow}$	#1	35.3	#3	78.3

Table 5.5: LLM **dev set** performance on the two datasets, measured with AUC_{norm} (0% = random chance performance). AUC is calculated using estimated model scores as in §5.4.2 and then normalized into AUC_{norm} . The highest scoring template is selected on each dev task (shown in this table) and this template is used in the corresponding test set evaluation in later sections.

³For Random-Premise experiments, AUC values cannot be meaningfully calculated because gold labels are always No-Entail. For these experiments, the most frequently-selected prompt template in the other settings is used on each dataset, namely template #1 for Levy/Holt dataset, and template #3 for RTE-1 dataset.

5.4.3 Scoring Model Output

Model textual generations are used for scoring in two ways.

For behavioral experiments in §5.5, §5.6, and §5.7, the model is scored solely based on its textual response. In these experiments, choice A is converted into Entail and both B and C choices are collapsed into No-Entail, in order to align with Levy/Holt and RTE-1 annotation. In initial tests, models choose one of A/B/C on all dev questions, showing compatibility with the QA format.

For the analysis in §5.8, which measures model performance allowing for a tunable confidence threshold, the letter choice is converted to a score with the mapping:

$$S_{\text{ent}} = 0.5 + 0.5 * \mathbb{I}[\text{tok} = \mathbf{A}] * S_{\text{tok}} \\ - 0.5 * \mathbb{I}[\text{tok} \in \{\mathbf{B}, \mathbf{C}\}] * S_{\text{tok}}$$

Where \mathbb{I} is the indicator function, and S_{ent} estimates the likelihood of Entail from a textual output ($0 \leq S_{\text{ent}} \leq 1$) with token probability S_{tok} using a linear transformation, preserving the ordering of model confidences, which is sufficient for calculating a precision-recall curve. The token probability S_{tok} is simply retrieved from the model’s distribution over possible outputs.

5.5 Experiment 1: Attestation Bias

First, an assessment of LLMs’ reliance on propositional memory of training text by conditioning each model’s entailment task predictions I on its own predictions of attestation Λ . This is done by comparing the estimated probability of predicting Entail conditioned on whether the hypothesis is predicted Attested or not.

Further, a control setting is tested which accounts for the possibility that original Levy/Holt entailments may coincidentally refer to attested facts, which could lead to spurious correlation between inference and attestation scores without clearly demonstrating use of memory versus true entailment. This controlled setting is the random premise task I_{RandPrem} , which converts entailments into non-entailments without altering the hypothesis. An ideal model capable of drawing inferences from information in context should detect that in the I_{RandPrem} task it is no longer possible to infer the hypothesis based on the premise (even if the hypothesis is itself attested in training), and never predict Entail. Thus, in I_{RandPrem} , all Entail predictions are assumed to be false positive hallucinations.

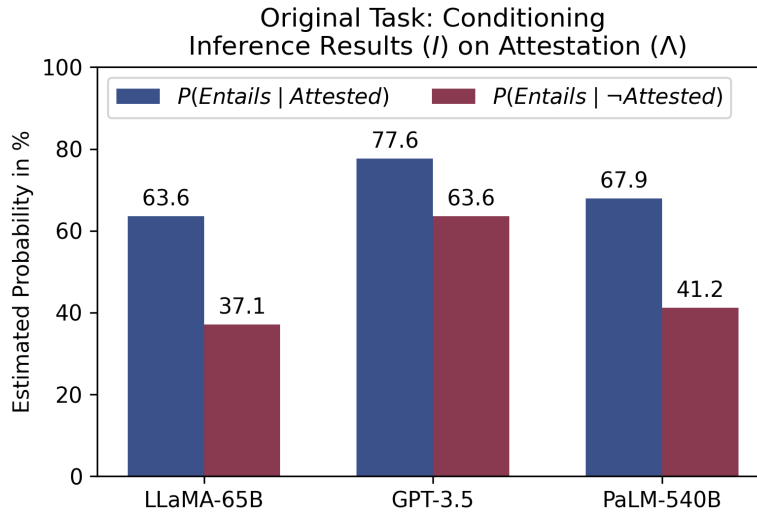


Figure 5.1: Estimated probability of predicting Entail for **original** entries in Levy/Holt, conditioned on LLMs’ attestation of hypotheses (Λ). This setting is intuitive but may be subject to spurious correlations, thus is included but colored darker.

5.5.1 Results

With I , $I_{RandPrem}$ and Λ predictions acquired as described in §5.3.1, the conditional probabilities are presented in Figure 5.1 and Figure 5.2. It is clear that a model’s memory about the hypothesis plays a part in its predictions of the hypothesis given a premise, either related or random.

For I , a significantly higher probability of predicting Entail is observed when the hypothesis is attested. In the random premise task $I_{RandPrem}$, this trend continues. LLaMA, GPT-3.5, and PaLM, respectively, show a 1.9x, 2.2x, and 2.0x higher chance of falsely predicting that a random premise Entails the hypothesis if it already predicts the hypothesis is attested. The impact of such hallucination on NLI performance is further investigated in §5.8.

This behavior is observed across model families (LLaMA, GPT, and PaLM), establishing that it is due to pretraining rather than instruction-finetuning or RLHF, since LLaMA and PaLM have only undergone pretraining. This is undesirable, because model predictions on NLI tasks should be based solely on general language understanding, not prior propositional knowledge. It may be concluded that memory of training data is a significant contributor in LLM inference, and may be an important source of hallucination.

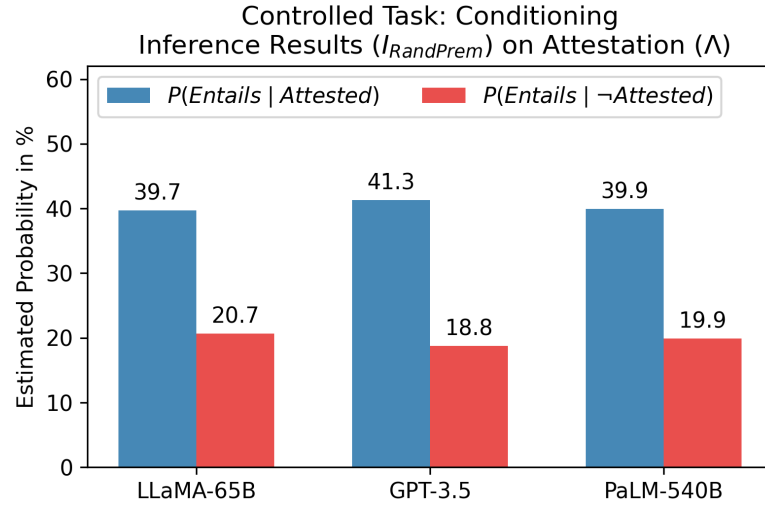


Figure 5.2: Estimated probability of predicting Entail for **Random-Premise** entries in Levy/Holt, conditioned on LLMs’ attestation of hypotheses (Λ). In this setting, predicting Entail is false positive hallucination (lower is better). Models are sensitive to hypothesis attestation, and hallucinate more when the hypotheses are attested.

5.5.2 Implications for Real Applications

Using prior knowledge as part of language inference has bad implications for the use of LLMs in real applications. An example scenario is now illustrated of a question-answering task where user questions are answered from a Knowledge Base (KB). In typical formulations of this task, if a statement in the KB (premise) entails a user query (hypothesis), the premise may be formulated into an answer. Consider a KB such as a legal document or HR rulebook. Assume that the text is prepended to the user query and presented to the LLM, as in other works (Srinivasan et al., 2022). Given the findings of this work, the LLM may hallucinate answers to questions using information which is not present within the KB text, but may have been read by the LLM in text from other sources during pretraining. These answers could be illogical, contradictory, and could misrepresent the views of the KB, or other harms. Such poor use of in-context learning has already been observed in specific domains like medicine (Jimenez Gutierrez et al., 2022).

In general, this is a risk for LLMs which (a) are deployed for tasks like QA by feeding novel text (e.g. a legal document) in-context as part of the user query, and (b) are trained on datasets which are private or otherwise infeasibly large to read manually, containing many facts and human opinions unknowable to both the user and modeler.

5.6 Experiment 2: Entities are Indices to Memory

In §5.5, it is established that propositional memory explains a significant portion of false positives in LLM inference predictions. This section further explores the importance of named entities in the process of LLMs’ memory recall.

As described in §5.3.3, the entities are manipulated in the I^{GenArg} generic argument replacement setting, and in the two random entity replacements: one with infrequent-entities $I^{RandArg\downarrow}$ and one with frequent-entities $I^{RandArg\uparrow}$ (examples in Table 5.2).

By replacing arguments constrained by type, entailment labels are maintained; however, these new samples should contain novel strings which are not attested in pre-train corpora. It is expected that an ideal, generalizing model would maintain its predictions across all conditions; a flawed model utilizing the *attestation bias* would predict fewer Entail than on an original dataset, since entities can no longer identify sample statements from training.

5.6.1 Results on Levy/Holt

Results are reported across conditions for the Levy/Holt directional subset in Table 5.6. Two phenomena are observed across all three models, aligning with the above conjecture of “flaws.”

First, all models’ behavior significantly changes in the same way when original entities are replaced by either entity types or random real entities. Despite similar (or marginally increasing) precision across conditions, recall degrades sharply from original entities (I) (GPT-3.5 @92.3) to random frequent entities ($I^{RandArg\uparrow}$) (GPT-3.5 @55.3). Generic-argument I^{GenArg} performance also degrades in this way, showing that this is not a matter of poorly selected real entities, but rather a loss of information from the original dataset which models were using to answer questions.

Second, across the 3 models, recall is significantly different between the two real entity conditions $I^{RandArg\downarrow}$ and $I^{RandArg\uparrow}$, which are both composed of unattested statements, but involve entities that differ in typical corpus frequency. Infrequent entities ($I^{RandArg\downarrow}$) yield better generalization and a higher recall (GPT-3.5 @66.5) than frequent entities ($I^{RandArg\uparrow}$) (GPT-3.5 @55.3).

These findings corroborate those from §5.5, that LLMs draw from memorized statements when queried to perform language inference. Additionally, it is shown that these memories are recalled using named entities acting as indices. These experiments demonstrate that too much prior exposure to an entity may impede model generaliza-

Model	Task	Levy/Holt (Directional)		
		Precision	Recall	Δ -Recall
LLaMA	<i>I</i>	67.0	68.4	0
	<i>I</i> ^{GenArg}	69.0	66.9	-1.5
	<i>I</i> ^{RandArg} _↓	64.0	63.8	-4.6
	<i>I</i> ^{RandArg} _↑	67.2	<u>53.7</u>	-14.7
GPT-3.5	<i>I</i>	62.4	92.3	0
	<i>I</i> ^{GenArg}	65.1	75.7	-16.6
	<i>I</i> ^{RandArg} _↓	65.5	66.5	-25.8
	<i>I</i> ^{RandArg} _↑	68.8	<u>55.3</u>	-37.0
PaLM	<i>I</i>	72.8	76.2	0
	<i>I</i> ^{GenArg}	79.8	<u>50.8</u>	-25.4
	<i>I</i> ^{RandArg} _↓	69.5	58.7	-17.5
	<i>I</i> ^{RandArg} _↑	70.8	52.4	-23.8

Table 5.6: Results for models across different argument-replacement tasks on **Levy/Holt**. The **highest** and lowest recall scores are indicated across replacement settings for a given model. Notably, recall decreases sharply across settings in all models.

tion when that entity is discussed in novel inferences: the more a model has read about an entity during pretraining, the less capable it is of drawing novel natural language inferences involving it, even though those inferences do not require detailed knowledge of the entity. Like §5.5, the effect is consistent across models, indicating LLM pretraining is responsible.

5.6.2 Results on RTE-1

The RTE-1 dataset contains complex natural language statements with varied linguistic features, so predictions about entailment are not decidable only on the basis of contained predicates. However, RTE-1 is a difficult challenge set for models, and interesting to compare to in the broader domain of NLI. Though the sentences are much more complex, an analogous experiment is conducted by first identifying spans of named entities and their respective entity types, then replacing the entities with new ones. Results are shown in Table 5.7.

Model	Task	RTE-1		
		Precision	Recall	Δ -Recall
LLaMA	<i>I</i>	74.5	52.5	0
	<i>I</i> ^{GenArg}	70.9	57.3	+4.8
	<i>I</i> ^{RandArg} ↓	66.9	60.5	+8.0
	<i>I</i> ^{RandArg} ↑	70.6	<u>51.5</u>	-1.0
GPT-3.5	<i>I</i>	80.6	96.5	0
	<i>I</i> ^{GenArg}	79.7	91.3	-5.2
	<i>I</i> ^{RandArg} ↓	80.1	82.5	-14.0
	<i>I</i> ^{RandArg} ↑	81.9	<u>80.5</u>	-16.0
PaLM	<i>I</i>	90.3	84.0	0
	<i>I</i> ^{GenArg}	92.3	<u>71.5</u>	-12.5
	<i>I</i> ^{RandArg} ↓	87.8	82.5	-1.5
	<i>I</i> ^{RandArg} ↑	88.2	82.0	-2.0

Table 5.7: Results for models across different argument-replacement tasks on **RTE-1**. The **highest** and lowest recall score are indicated across replacement settings, per-model.

Similar trends are observed to those reported on Levy/Holt. GPT-3.5 performs very consistently between Levy/Holt and RTE-1 in terms of degrading recall when information is changed in each sample. Model performance is significantly worse than the original dataset when using generic arguments, and worse still using type-constrained random arguments. Further, across all three LLMs across both datasets, models consistently achieve worse recall using high-frequency entities than low-frequency entities, supporting the claim that increasing the frequency of entity occurrence in training data impedes generalization.

Different from in Levy/Holt, some noisiness in LLaMA’s predictions are observed on RTE-1; the recall on the original task is actually lower than the generic argument condition and the low-frequency entity condition. It is important to note that overall, LLaMA is the weakest LLM tested in this experiment on both Levy/Holt and RTE-1, and that its performance on RTE-1 is particularly low. It may be that the increased difficulty of RTE-1 over Levy/Holt (due to having much more linguistic variation) is simply too complex for LLaMA, which is neither the largest LLM tested, nor instruction-

Task	GPT-3.5	Instructed to Ignore Attestedness	Not Instructed
I	$P(\text{Entail} \mid \text{Attested})$	74.3	77.6
I	$P(\text{Entail} \mid \neg\text{Attested})$	57.8	63.6
I_{RandPrem}	$P(\text{Entail} \mid \text{Attested})$	39.0	41.3
I_{RandPrem}	$P(\text{Entail} \mid \neg\text{Attested})$	17.6	18.8

Table 5.8: The probability of positive predictions in I and I_{RandPrem} tasks are estimated given that the hypothesis is predicted attested, $\Lambda = \text{Attested}$. **Not instructed** results are copied from Figure 5.2 and listed here for ease of comparison; also note that all $I_{\text{RandPrem}} = \text{Entail}$ predictions are false positives. Very little change is observed between instruction settings.

finetuned.

A smaller gap is also observed between PaLM’s recall rates across dataset conditions, though the gaps are consistent with the claims of this work. And while the model appears able to generalize to conditions in which random real arguments are inserted, recall on the generic argument condition is significantly degraded. Failure on this control condition indicates that the model may not be generalizing as well as the other conditions would imply.

5.6.3 Instructing LLMs to Ignore Attestation

In §5.5 and §5.6, it is shown that entailment predictions from LLMs are strongly biased by their predictions on the attestation of hypotheses. It is possible that there are intuitive prompt engineering techniques to steer LLM behavior away from utilizing attestation.

Towards this goal, a further investigation was conducted by prepending a brief task description to the few-shot prompts (illustrated in Table 5.3, bottom) explicitly instructing the models to ignore the attestedness of individual statements: “Please check the entailments between the following hypothetical statements. Ignore the veracity of these statements.”

The experiments in §5.5 and §5.6 are re-done using the new template and GPT-3.5, since GPT-3.5 is an instruction-finetuned model trained to be responsive to prompts, where the other two LLM families are only pre-trained. Despite having been instruction-finetuned, the results with GPT-3.5 show marginal improvement in model behavior.

GPT-3.5 Condition	Task	Levy/Holt (Directional)		
		Precision	Recall	Δ -Recall
Few-shot, instructed to ignore attestedness.	I	64.9	90.8	0
	I^{GenArg}	73.5	69.3	-21.5
	$I^{RandArg\downarrow}$	64.6	68.4	-22.4
	$I^{RandArg\uparrow}$	67.5	<u>58.1</u>	-32.7
Few-shot, no instructions.	I	62.4	92.3	0
	I^{GenArg}	65.1	75.7	-16.6
	$I^{RandArg\downarrow}$	65.5	66.5	-25.8
	$I^{RandArg\uparrow}$	68.8	<u>55.3</u>	-37.0

Table 5.9: GPT-3.5 predictions when models are explicitly instructed to avoid taking the attestedness of individual statements into account. In the upper half are the instructed behavior, and in the lower half are the previous few-shot results reproduced from Table 5.6. Differences in recalls remain at a similar scale, with precision again stable, and the benefit from the explicit instruction is marginal.

In Table 5.8, it is shown that instructing GPT-3.5 to ignore attestation does not help narrow the gap between $\Lambda = \text{Attested}$ and $\Lambda = \neg\text{Attested}$; instead, probabilities of predicting Entail went down by similar amounts, indicating that the model becomes slightly more conservative in predicting positives when instructed to ignore attestation, but not in a useful manner.

Further, as shown in Table 5.9, despite the explicit instruction, recall still drops at similar scales when arguments are randomly replaced with the same sets of frequent/infrequent replacement entities as before. Since GPT-3.5 has been instruction-finetuned to respond to prompts, its failure means eradicating such biases from model outputs is a difficult task, one that needs further research attention.

5.7 Experiment 3: Relative Frequency Bias

The conditioning experiments from §5.5 are continued, now exploring the relative frequency bias. Sample labels for this bias are denoted by the model-agnostic Φ as described in §5.3.1. Φ labels the conformance of sample predicates to the bias: $\Phi_{<}$ means P is less corpus-frequent than H by a margin (positive class), $\Phi_{>}$ means P more

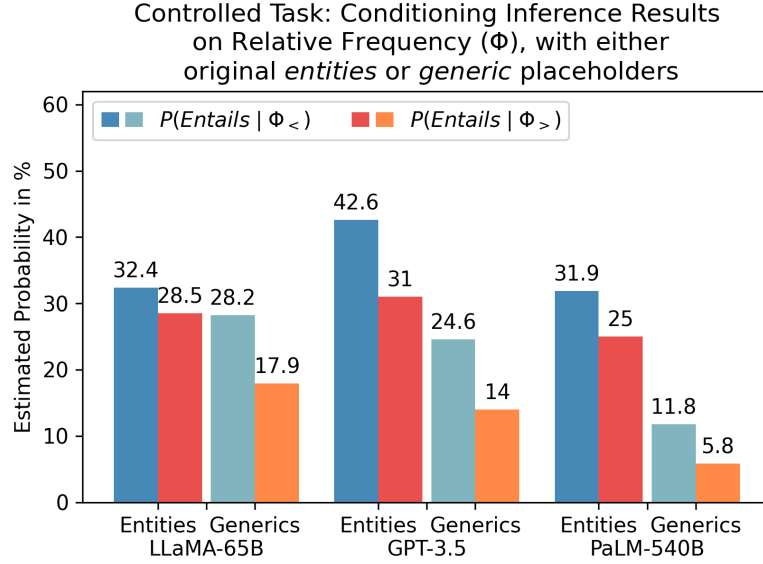


Figure 5.3: Estimated probability of predicting Entail for **random-premise** Levy/Holt conditioned on relative frequencies (Φ), with original ($I_{RandPrem}$) or generic ($I_{RandPrem}^{GenArg}$) entities. Predicting Entail is false positive hallucination (lower is better). Models hallucinate more often when test samples conform to the relative frequency bias ($\Phi_{<}$) than when not ($\Phi_{>}$).

frequent than H by the margin (negative class). To control for differences between datasets, the margin is set so that 1/3 of samples are classed as “roughly equal” (Φ_{\approx}), which are then discarded.

Following the observations in §5.6, a generic-argument transformation is further applied to control for attestation, yielding $I_{RandPrem}^{GenArg}$. With the entities masked, models cannot recall propositional memory for this task: by re-calculating the Λ measure with generic arguments, only 2 hypotheses are still predicted as Attested by GPT-3.5, whereas for LLaMA and PaLM, the numbers are also only 6.2% and 3.9%. Additionally, as with $I_{RandPrem}$, here the entailment label of each sample remains No-Entail, so any Entail prediction is false positive hallucination.

5.7.1 Results

The probabilities of models predicting Entail are estimated conditioned on the Frequency label Φ , between $I_{RandPrem}$ and $I_{RandPrem}^{GenArg}$ settings. The results are presented in Figure 5.3. A clear and consistent rise of hallucination is observed when samples conform to the bias. Namely, in case of $\Phi_{<}$, models are more likely to predict Entail, even though no semantic relation exists between P and H .

Number of Entries	Levy/Holt			RTE-1		
	LLaMA	GPT-3.5	PaLM	LLaMA	GPT-3.5	PaLM
$\Lambda_{\text{CONSISTENT}}$	955	947	999	479	447	480
$\Lambda_{\text{ADVERSARIAL}}$	829	837	785	321	353	320
$\Phi_{\text{CONSISTENT}}$	972			286		
$\Phi_{\text{ADVERSARIAL}}$	220			247		

Table 5.10: Subsets defined by the consistency between entailment label L and either Λ (hypothesis attestation prediction from each LLM) or Φ (model-agnostic relative frequency bias). CONSISTENT subsets are where L agrees with Λ/Φ . ADVERSARIAL subsets are where L disagrees with Λ/Φ .

In I_{RandPrem} , when entities are available, this effect is moderate. On the other hand, with $I_{\text{RandPrem}}^{\text{GenArg}}$ when entity-based memory is blocked, a decrease is observed in the overall level of hallucination, but the separation between $\Phi_{<}$ and $\Phi_{>}$ becomes more drastic, to 1.6x, 1.8x and 2.0x for LLaMA, GPT-3.5 and PaLM respectively. This indicates a tension between Λ and Φ : propositional memory may be used when available, and if not, the predicate pairing may be attended to more closely. Again, the Φ effect is observed across the three model families, revealing its root in the large-scale pretraining process, rather than model peculiarities or fine-tuning.

5.8 Impact of Bias on Performance

Two sources of hallucination by LLMs have been demonstrated on inference tasks. Now, their impact on model performance is assessed.

LLMs’ performance is compared between NLI subsets that are *consistent* or *adversarial* to each factor. A sample $P \models H?$ is *consistent* with a factor when the prediction by the factor **agrees with** the gold entailment label; conversely, it is *adversarial* to a factor when the prediction with the factor **disagrees with** the label.

For example, “Google **bought** YouTube \models Google **owns** YouTube” is *consistent* with the attestation bias of every model, because the conclusion *Google owns YouTube* is attested in every LLM’s training data, and the sample label is Entail; “Apple **owns** Samsung $\not\models$ Apple **bought** Samsung” is also *consistent*, because its conclusion is not attested and the sample label is No-Entail. The reverses of these two samples are

Model	Task	Levy/Holt					
		Attestation (Λ)			Rel. Frequency (Φ)		
		<i>cons.</i>	<i>adv.</i>	<i>diff.</i>	<i>cons.</i>	<i>adv.</i>	<i>diff.</i>
LLaMA	<i>I</i>	65.5	8.1	-57.4	42.1	32.3	-9.8
GPT-3.5	<i>I</i>	85.0	10.8	-74.2	53.5	43.2	-10.3
PaLM	<i>I</i>	79.1	31.5	-47.6	63.3	53.0	-10.3
LLaMA	<i>I^{GenArg}</i>	52.1	34.4	-17.7	55.3	34.9	-20.4
GPT-3.5	<i>I^{GenArg}</i>	67.1	18.8	-48.3	50.4	35.0	-15.4
PaLM	<i>I^{GenArg}</i>	58.1	46.6	-11.5	59.9	47.3	-12.6

Table 5.11: LLM performance on subsets where Λ/Φ is *consistent/adversarial* to entailment labels, measured with AUC_{norm} (0% = random chance performance). Decrease from *cons* to *adv* subsets are shown in the *diff.* columns.

adversarial, since their respective attestedness (unchanged) does not agree with the entailment labels (now flipped). For each subset, there is substantial representation in both Levy/Holt and RTE-1 (see Table 5.10).

While earlier experiments inspected model textual responses to characterize behavior change, area under the precision-recall curve (AUC) is used here to summarize model performance over a tunable confidence threshold (scoring described in §5.4.2), which is better for measuring practical discriminative power when a modeler is able to tune a threshold for the desired application. Following Li et al. (2022a), AUC values are re-scaled to normalize over the label distribution, yielding AUC_{norm} values that assign random classifiers 0% and perfect classifiers 100%.

5.8.1 Results

Results are reported in Table 5.11 and Table 5.12. On the standard inference task *I*, the performance drop from $\Lambda_{\text{CONSISTENT}}$ to $\Lambda_{\text{ADVERSARIAL}}$ is severe for all 3 LLMs: they deteriorate from very good classifiers to poor or even near-random ones. This fragility from the *attestation bias* can be alleviated by masking entities with type-identifiers (condition *I^{GenArg}*), which reduces the performance drop.

On the other hand, with the generic arguments in *I^{GenArg}*, LLMs are forced to focus on the predicates in each proposition. As a result, the impact of the *relative frequency bias* is intensified. From the standard inference task *I* to *I^{GenArg}*, the average

Model	Task	RTE-1					
		Attestation (Λ)			Relative Frequency (Φ)		
		<i>cons.</i>	<i>adv.</i>	<i>diff.</i>	<i>cons.</i>	<i>adv.</i>	<i>diff.</i>
LLaMA	<i>I</i>	62.1	37.4	-24.7	55.5	51.7	-3.8
GPT-3.5	<i>I</i>	84.6	47.5	-37.1	77.6	43.4	-34.2
PaLM	<i>I</i>	87.1	83.4	-3.7	87.5	81.0	-6.5
LLaMA	<i>I^{GenArg}</i>	59.2	30.4	-28.8	51.7	39.4	-12.3
GPT-3.5	<i>I^{GenArg}</i>	80.1	56.4	-23.7	79.6	49.1	-30.5
PaLM	<i>I^{GenArg}</i>	78.1	84.4	+6.3	85.4	78.7	-6.7

Table 5.12: LLM performance on subsets where Λ/Φ is *consistent/adversarial* to entailment labels, measured with AUC_{norm} (0% = random chance performance). Decrease from *cons* to *adv* subsets are shown in the *diff.* columns.

performance drop from the *cons.* to *adv.* subsets with respect to Φ is widened from 10.1% to 16.1% for Levy/Holt and from 14.8% to 16.5% for RTE-1. The differences for Φ -consistency subsets are generally narrower than Λ -consistency subsets, possibly because the relative frequencies require generalizing from instances, and may be more difficult to capture, and potentially because frequency measures with Google N-gram are a crude estimate of the actual frequencies in LLM pretraining corpora.

5.9 Conclusion

Despite promising performance on original datasets, across several major LLM families and experimental settings, two important biases in the performance of LLMs are demonstrated on natural language inference tasks, which may also manifest in applied tasks as hallucination. Contrary to claims of LLM general reasoning capabilities, this work shows that much of this performance is achieved by (1) recall of relevant memorizations and (2) corpus-based biases like term frequency. Since these factors are reproduced in all models, it is established that they originate in LLM pre-training, the common training phase, and that they are not corrected during GPT-3.5 fine-tuning.

In conclusion, LLMs, though powerful, use unsatisfactory tools for the basic tasks of language understanding and inference. This work proposes several approaches to control for these biases in evaluation, and ultimately recommends that further research

is done to alleviate these biases before LLMs may be trusted to reason robustly about language, especially in domains in which precision is crucial.

5.9.1 Limitations

This work discusses two prominent sources of hallucination for LLMs in natural language inference tasks. It is important to acknowledge that this is not an exhaustive search of all the sources, and further exploration should be done in future work.

Of note, after controlling for the factors discussed, there remains residual, unexplained performance on NLI tasks. This residual might be due to other undiscovered biases or even generalising inference capability. Further exploration of this residual is left to future work.

As discussed in §5.4.2.2, a range of popular LLM prompting techniques are compared, and the most promising approaches are selected. There could also be other novel prompting techniques which help the LLMs resist the biases discussed in this work. This is an open question and indicated for future research.

5.9.2 Ethical Considerations

This work discusses two major sources of hallucination in LLM output when asked to perform natural language inference, which is a capability required of many downstream tasks such as summarization, question answering, etc. This work shows that users of LLMs may be subjected to faulty judgements if the content of their request overlaps with data in pretraining. However, it is difficult to ascertain for both a user or modeler exactly what is contained in pretraining data, or how this will interact with a user's query. The proposed attestation query of this work shows promise in detecting potential overlaps, but model responses in applications of these cases are not explored. Further, the relative frequency bias demonstrates a much more subtle problem of corpus distribution that is naturally inherent to model pretraining on human generated text.

In light of these, the potential harms of LLM use for drawing natural language inferences may include: offering inaccurate or irrelevant information to a user's query or contradiction of information provided in-context with a user's query.

Chapter 6

Conclusion

This thesis addresses the problem of directional inference over predicates in the broad and open domain of natural language. Sometimes called “commonsense” inference or “entailment,” it is a capability that is foundational to humans and models alike for many downstream tasks such as question answering, summarization, and more. And while simple non-directional methods such as similarity between terms are widely used, only *directional* entailment enables high precision language inference, a central aspect of this thesis.

6.1 Summary of Findings

Several key contributions are made to the problem of directional predicate inference in the open domain. This thesis begins by considering Entailment Graphs, which are constructed by machine reading of text and the application of a learning algorithm, the Distributional Inclusion Hypothesis. They contain vertices representing natural language predicates encountered in training corpora and edges representing their explicit directional entailments. However, they are restricted in terms of the predicates learned in these graphs. First, EGs learned using the DIH are restricted in their inability to represent entailments between predicates of different valencies. Second, EGs learned by machine reading of corpora face a significant practical restriction in learning predicate symbols for the totality of natural language predicates, which are unbounded.

This thesis expands on the underlying learning theory of Entailment Graphs, the DIH, demonstrating novel learning of entailments across valencies in a fully open domain of natural language. However, this thesis recognizes that an EG, especially one built using machine reading of corpora, faces a practical limitation. Due to the distri-

bution of predicates in natural corpora, it is impractical to learn entailments for every predicate in a language by machine reading, yet simultaneously very likely that many of these unlearned predicates may be involved in real queries. The number of natural language predicates is unbounded, so this problem likely affects *any* extant EG. Thus, this thesis turns to sub-symbolic encoding of predicates using Language Models to explore the possibility of smoothing EGs at test-time to handle unseen predicates. Language Model embeddings are shown to benefit in detecting directional entailment coupled with an Entailment Graph, but only for smoothing premise predicates. The smoothing of hypothesis predicates is shown to be possible in principle, but remains a challenge. Finally, following from this, this thesis explores the use of Large Language Models on their own for directional entailment detection. While they show superficial performance, LLMs are demonstrated to be only an approximation of linguistic reasoning, exploiting memorized world knowledge and other corpus-related artifacts in their predictions.

6.1.1 Multivalent Entailment Graphs

Chapter 3 presents an extension to the Distributional Inclusion Hypothesis in the context of predicate entailment learning. This thesis presents the Multivalent Distributional Inclusion Hypothesis, which offers a more flexible interpretation which accounts for the roles of arguments in eventualities in text. This allows for the learning of Entailment Graphs which contain entailments *within* and *between* predicate valencies. Now, entailments may be learned such as “ x kills y ” entails “ y is dead.” The MDIH describes that, provided enough training text, entailments may be learned from a predicate relating any number of arguments to predicates relating any subset of its arguments. This is demonstrated by learning entailments from a binary predicate relating arguments x and y to a unary predicate applying to just x or y . These binary-to-unary entailments are learned in addition to binary-to-binary and unary-to-unary entailments.

Additionally, a new automatic boolean question answering task is developed, which can be tailored to generate test questions for models, which verify if hypotheses are true or false, conditioned on a span of news text. This new evaluation is the first to test for multivalent entailment of predicates, and the novel Multivalent Entailment Graphs show a clear advantage over baselines. They draw from both binary and unary antecedents to answer more questions than using a single valency alone. Further, on these questions of fine-grained semantics, the utility of directional inference is demonstrated

to surpass non-directional similarity measures implemented using computationally-comparable, unsupervised Language Models like RoBERTa. Yet, the limitation of extant EGs is also shown; like binary-only EGs, Multivalent EGs also suffer from vertex sparsity, showing limited recall compared to Language Models due to missing predicates in the graph at test-time.

6.1.2 Combining Entailment Graphs and Language Models

Following from these findings, Chapter 4 explores a means of combining the benefits of Entailment Graphs with Language Models, while maintaining an unsupervised approach with high directional precision. First, a theory is developed for the smoothing of symbolic inference models such as Entailment Graphs to overcome sparsity of predicate symbols by introducing a related, replacement predicate which completes a transitive chain from the original premise to the original hypothesis. Such chains are guaranteed to maintain directional precision, if they can be constructed.

A method of EG smoothing is presented in which a Language Model encodes a query predicate into a vector embedding so that a nearest neighbor approximation can be found amongst the learned predicates in the EG, and the EG is then able to complete the directional inference. Due to the naturally arising frequency/generalization gradient in Language Model embedding space, it is predicted that this method should produce more semantically general approximations for a target predicate, and will thus complete a transitive chain when used for premise-smoothing. As predicted, this method is demonstrated for premise-smoothing to positive effect, improving recall and precision on several test datasets. Premise-smoothing even shows a benefit in boolean question answering, on a dataset similar to that of Chapter 3. In particular, it is useful for questions which have sparse supporting evidence, when missing a possible premise is most harmful. Further, this smoothing method is shown to be detrimental to hypothesis smoothing, which requires semantic specialization, not generalization.

A controlled experiment using WordNet relations to produce generalized or specialized approximations corroborates these findings. It is concluded that hypothesis-smoothing is intrinsically more difficult due to the increasing sparsity of highly-specialized predicates in text and Entailment Graphs.

6.1.3 Entailment by Large Language Models Alone

The effective leveraging of Language Model embedding space for directional entailment, as well as contemporaneous advances in Language Model research, naturally lead to the research question investigated in Chapter 5. This chapter investigates the best available Language Models for their ability to perform predicate inference on their own, unsupervised and in the open-domain of natural language, by processing free text without an Entailment Graph. A series of behavioral experiments are conducted on several Large Language Models (LLaMA-65B, GPT-3.5, and PaLM-540B) to test their capability for directional predicate inference. High performance on basic directional entailment datasets would indicate this capability is available for other useful tasks like question answering or summarization, important use-cases for these interactive models. Indeed, on several datasets including one used in earlier tests with EGs, LLMs appear to perform excellently. But by altering targeted aspects of test statements and observing model behavior change, two unsatisfactory sources of model performance are identified.

Across all three model families and various prompting techniques, LLMs are shown to use two biases in inference decisions, and their origin is established in the pretraining phase, the step common to all models. The objective of pretraining, which is the statistical modeling of the natural distribution of human-generated text, leads to (at the level of sentences) memorizing individual statements, and (at the level of corpora) learning typical patterns of usage. LLMs are shown to use factual recall when answering predicate inference questions, affirming entailments more readily when the hypothesis is attested by training data, regardless of the premise or actual entailment label. Further, LLMs are similarly shown to use a corpus-based heuristic, the relative frequency of predicates in training data (the same phenomenon leveraged in Chapter 4) to affirm hypotheses regardless of entailment label. In data subsets designed to be adversarial to these biases, LLM performance degrades significantly, sometimes even to near-random levels. Though there may be other unknown biases, there is still unexplained performance, and LLMs may be capable of generalizing language inference to a degree. However, it is clear that LLMs present only an *approximation* of linguistic reasoning for predicate inference, which may be useful in tasks where false positives can be tolerated, but not in tasks where precision is critical.

6.2 Directions for Future Work

The work presented in this thesis points toward several directions for future research.

6.2.1 Discovering Metarelations in Entailment Graphs

Entailment Graphs present many benefits, such as being explainable and editable, as well being theoretically-founded in construction using the Distributional Inclusion Hypothesis. Further refinement to their construction may yield interesting and useful results, similar to the work presented here in multivalent learning, or the work in [Bijl de Vroe \(2023\)](#) that adds temporal disambiguation of individual predicate occurrences which are used for learning general typed predicate entailments.

One such avenue is *metarelations* which can be derived to describe entailment relations in a more fine-grained manner. Currently, all EGs coarsely define entailment between two predicates, such as *x buys y* entails *x owns y* but do not elaborate on the nature of this entailment. In this case, *buys* is a direct cause of *owns*, a sufficient, but not necessary, causal factor, since there may be other sufficient causes which lead to ownership, such as inheritance. But entailment may not always signal causation. For example, *x gets elected to y* entails *x is a candidate for y*, but *gets elected* does not cause *is a candidate*. This is instead an example of precondition: *is a candidate* is a necessary but not sufficient causal factor for *gets elected*, so given a predicate such as *gets elected to* it must necessarily also be true that there was *is a candidate* leading up to it.

Knowledge of causal factors goes beyond the capability of directional inference and would be very useful for robust models of reasoning, to answer “why” and “how” questions such as *how did x come to own y*? It may be that all entailment relations also possess an underlying causal metarelation, or there may be other kinds of entailments as well. Investigating such a classification scheme and deriving metarelations is open for future work.

6.2.2 Improvement in Entailment Graph Coverage

The problem of vertex sparsity remains a challenge for Entailment Graphs. This thesis contributes its discovery, an explanation of the problem, and a first attempt at solving it using unsupervised means. But there may be better ways of overcoming this issue.

First, though more difficult than premise-smoothing, it may still be possible to

provide hypothesis-smoothing by leverage Language Model embeddings to discover specializations of missing predicates which are present in the EG. Controlled experiments with WordNet indicate that specializations are indeed present in the Entailment Graph of [Hosseini et al. \(2018\)](#), which are useful in smoothing. Recovering these using a more robust method such as embedding search is an open direction.

Second, it may be possible to use the Entailment Graph learning signal (machine reading using the DIH) to train a different kind of model which solely represents predicates in a subsymbolic way, without construction of an extant EG. For example, a neural network which processes subsymbolic word embeddings. The RoBERTa-finetuned model in [Schmitt and Schütze \(2021\)](#) is an example of this, and is shown to be very performant on datasets, though it overfits to artifacts during training, leaving an open question if such a model is capable of learning directional entailment in the open domain, using generalizing linguistic reasoning by attending to different features in the input. Training a model using confident Entailment Graph edges may be a promising direction.

6.2.3 Improvement in LLM Training Objectives

Large Language Models are being continuously improved, and it may be advantageous to improve model pretraining in a way which induces better generalizing language inference. For example, it may be that including synthetic data which demonstrates the usage of a predicate with a more comprehensive variety of arguments may help LLMs learn predicate representations which are not tied as strongly to particular entities. Of course, the objective of next-word prediction characteristic of LLM pretraining is not actually tied to triples at all, and this fact could (speculatively) be leading to leakage and blending of facts, another source of hallucination. It may ultimately be required to rethink the training objective in order to prevent LLMs from hallucinating new facts, though this may not solve the problem of LLMs using attestation in lieu of genuine language inference.

Bibliography

- Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet project. In *COLING-ACL '98: Proceedings of the Conference*, pages 86–90, Montreal, Canada.
- Barraut, L., Bojar, O., Costa-jussà, M. R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., Müller, M., Pal, S., Post, M., and Zampieri, M. (2019). Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.
- Berant, J., Alon, N., Dagan, I., and Goldberger, J. (2015). Efficient global learning of entailment graphs. *Computational Linguistics*, 41(2):221–263.
- Berant, J., Dagan, I., and Goldberger, J. (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.
- Bijl de Vroe, S., Guillou, L., Stanojevic, M., McKenna, N., and Steedman, M. (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).
- Bijl de Vroe, S. G. C. (2023). *Temporality and Modality in Entailment Graph Induction*. PhD thesis, University of Edinburgh.
- Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of*

- Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.
- Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Caraballo, S. A. and Charniak, E. (1999). Determining the specificity of nouns from text. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.
- Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. (2023). Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.
- Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318, Santa Cruz, California, USA. Association for Computational Linguistics.
- Chen, Z., Feng, Y., and Zhao, D. (2022). Entailment graph learning with textual entailment and soft transitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5899–5910, Dublin, Ireland. Association for Computational Linguistics.
- Chen, Z., Feng, Y., and Zhao, D. (2023). From the one, judge of the whole: Typed entailment graph construction with predicate generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

- Papers*), pages 3534–3551, Toronto, Canada. Association for Computational Linguistics.
- Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. (2022). Palm: Scaling language modeling with pathways. ArXiv.
- Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling instruction-finetuned language models. ArXiv.
- Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. (2019). BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.
- Dagan, I., Glickman, O., and Magnini, B. (2006). The pascal recognising textual entailment challenge. In Quiñero-Candela, J., Dagan, I., Magnini, B., and d’Alché Buc, F., editors, *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

- Dagan, I., Lee, L., and Pereira, F. C. (1999). Similarity-based models of word cooccurrence probabilities. *Machine learning*, 34(1-3):43–69.
- Dasgupta, I., Lampinen, A. K., Chan, S. C. Y., Creswell, A., Kumaran, D., McClelland, J. L., and Hill, F. (2022). Language models show human-like content effects on reasoning. ArXiv.
- Davidson, D. (1967). The logical form of action sentences. In Rescher, N., editor, *The Logic of Decision and Action*. University of Pittsburgh Press.
- Davis, W. (2019). Implicature. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2019 edition.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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.
- Fedus, W., Zoph, B., and Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.
- Feldbauer, R., Leodolter, M., Plant, C., and Flexer, A. (2018). Fast approximate hubness reduction for large high-dimensional data. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 358–367.
- Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.
- Finkel, J. R., Grenager, T., and Manning, C. (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.
- Geffet, M. and Dagan, I. (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.

- Geiger, A., Richardson, K., and Potts, C. (2020). Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.
- Gellerstam, M. (1986). Translationese in swedish novels translated from english. *Translation studies in Scandinavia*, 1:88–95.
- Grice, H. P. (1975). Logic and conversation. In Cole, P. and Morgan, J. L., editors, *Syntax and Semantics: Vol. 3: Speech Acts*, pages 41–58. Academic Press, New York.
- Guillou, L. and Bijl de Vroe, S. (2023). Ant dataset.
- Guillou, L., Bijl de Vroe, S., Hosseini, M. J., Johnson, M., and Steedman, M. (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.
- Guillou, L., Bijl de Vroe, S., Johnson, M., and Steedman, M. (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.
- Harris, Z. S. (1954). Distributional structure. *WORD*, 10(2-3):146–162.
- Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. (2021). Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J., and Sifre, L. (2022). An empirical analysis of compute-optimal large language model training. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030. Curran Associates, Inc.
- Holt, X. (2018). Probabilistic models of relational implication. Master’s thesis, Macquarie University.

- Hosseini, M. J. (2021). *Unsupervised Learning of Relational Entailment Graphs from Text*. PhD thesis, University of Edinburgh.
- Hosseini, M. J., Chambers, N., Reddy, S., Holt, X. R., Cohen, S. B., Johnson, M., and Steedman, M. (2018). Learning typed entailment graphs with global soft constraints. *Transactions of the Association for Computational Linguistics*, 6:703–717.
- Hosseini, M. J., Cohen, S. B., Johnson, M., and Steedman, M. (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.
- Hosseini, M. J., Cohen, S. B., Johnson, M., and Steedman, M. (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.
- Jimenez Gutierrez, B., McNeal, N., Washington, C., Chen, Y., Li, L., Sun, H., and Su, Y. (2022). Thinking about GPT-3 in-context learning for biomedical IE? think again. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Kandpal, N., Deng, H., Roberts, A., Wallace, E., and Raffel, C. (2023). Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.
- Kartsaklis, D. and Sadrzadeh, M. (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.
- Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

- Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Kelcey, M., Devlin, J., Lee, K., Toutanova, K. N., Jones, L., Chang, M.-W., Dai, A., Uszkoreit, J., Le, Q., and Petrov, S. (2019). Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.
- Levy, O. and Dagan, I. (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.
- Lewis, M. and Steedman, M. (2013). Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.
- Lewis, M. and Steedman, M. (2014). A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000, Doha, Qatar. Association for Computational Linguistics.
- Li, B., Zhou, H., He, J., Wang, M., Yang, Y., and Li, L. (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.
- Li, T., Hosseini, M. J., Weber, S., and Steedman, M. (2022a). Language models are poor learners of directional inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 903–921, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Li, T., Weber, S., Hosseini, M. J., Guillou, L., and Steedman, M. (2022b). Cross-lingual inference with a Chinese entailment graph. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1214–1233, Dublin, Ireland. Association for Computational Linguistics.
- Lin, D. (1998). Automatic retrieval and clustering of similar words. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*.
- Ling, X. and Weld, D. S. (2012). Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI’12*, page 94–100. AAAI Press.

- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.
- Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit.
- Luo, C., Liu, W., Lin, J., Zou, J., Xiang, M., and Ding, N. (2022). Simple but challenging: Natural language inference models fail on simple sentences. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3449–3462, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Maienborn, C. (2011). *Event semantics*, pages 802–829.
- Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- McKenna, N., Guillou, L., Hosseini, M. J., Bijl de Vroe, S., Johnson, M., and Steedman, M. (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.
- McKenna, N., Li, T., Cheng, L., Hosseini, M. J., Johnson, M., and Steedman, M. (2023a). Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore. Association for Computational Linguistics.
- McKenna, N., Li, T., Johnson, M., and Steedman, M. (2023b). Smoothing entailment graphs with language models. In *Proceedings of the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 13th International Joint Conference on Natural Language Processing*, Bali, Indonesia. Association for Computational Linguistics.
- Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2017). Pointer sentinel mixture models. In *International Conference on Learning Representations*.
- Mervis, C. B., Catlin, J., and Rosch, E. (1976). Relationships among goodness-of-example, category norms, and word frequency. *Bulletin of the psychonomic society*, 7(3):283–284.

- Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. (2018). Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.
- Nguyen, D. B., Hoffart, J., Theobald, M., and Weikum, G. (2014). Aida-light: High-throughput named-entity disambiguation. In Bizer, C., Heath, T., Auer, S., and Berners-Lee, T., editors, *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014*, volume 1184 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2020). Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Gray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.
- Pavlick, E. and Kwiatkowski, T. (2019). Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.
- Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2015). PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China. Association for Computational Linguistics.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Courn-

- peau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (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.
- Petroni, F., Rocktäschel, T., Miller, A. H., Lewis, P., Bakhtin, A., Wu, Y., and Riedel, S. (2019). Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.
- Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. (2018). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.
- Poon, H. and Domingos, P. (2009). Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Singapore. Association for Computational Linguistics.
- Rabinovich, E. and Wintner, S. (2015). Unsupervised identification of translationese. *Transactions of the Association for Computational Linguistics*, 3:419–432.
- Radford, A. and Wu, J. (2019). Rewon child, david luan, dario amodei, and ilya sutskever. 2019. *Language models are unsupervised multitask learners*. *OpenAI blog*, 1(8):9.
- Radovanović, M., Nanopoulos, A., and Ivanović, M. (2010). Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(86):2487–2531.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Rosch, E. and Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7(4):573–605.

- Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., and Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439.
- Schank, R. (1975). The structure of episodes in memory. In Bobrow, D. and Collins, A., editors, *Representation and Understanding*, pages 237–272. Academic Press, New York.
- Schmitt, M. and Schütze, H. (2019). SherLLiC: A typed event-focused lexical inference benchmark for evaluating natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 902–914, Florence, Italy. Association for Computational Linguistics.
- Schmitt, M. and Schütze, H. (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.
- Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Shwartz, V. and Choi, Y. (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.
- Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.
- Srinivasan, K., Raman, K., Samanta, A., Liao, L., Bertelli, L., and Bendersky, M. (2022). QUILL: Query intent with large language models using retrieval augmentation and multi-stage distillation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 492–501, Abu Dhabi, UAE. Association for Computational Linguistics.
- Stanojević, M. and Steedman, M. (2019). CCG parsing algorithm with incremental tree rotation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technolo-*

- gies, *Volume 1 (Long and Short Papers)*, pages 228–239, Minneapolis, Minnesota. Association for Computational Linguistics.
- Steedman, M. (2000). *The Syntactic Process*. MIT Press, Cambridge, MA, USA.
- Szpektor, I. and Dagan, I. (2008). Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 849–856, Manchester, UK. Coling 2008 Organizing Committee.
- Takahashi, S. and Tanaka-Ishii, K. (2017). Do neural nets learn statistical laws behind natural language? *PLOS ONE*, 12(12):1–17.
- Talman, A. and Chatzikyriakidis, S. (2019). Testing the generalization power of neural network models across NLI benchmarks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 85–94, Florence, Italy. Association for Computational Linguistics.
- Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023). Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Tarski, A. (1935). Der wahrheitsbegriff in den formalisierten sprachen. *Studia Philosophica*, 1:261–405.
- The Fracas Consortium, Cooper, R., Crouch, D., Eijck, J. V., Fox, C., Genabith, J. V., Jaspars, J., Kamp, H., Milward, D., Pinkal, M., Poesio, M., Pulman, S., Briscoe, T., Maier, H., and Konrad, K. (1996). Using the framework.
- The Library of Congress (2023). General information: 2021 at a glance.
- Tirumala, K., Markosyan, A. H., Zettlemoyer, L., and Aghajanyan, A. (2022). Memorization without overfitting: Analyzing the training dynamics of large language models. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models. ArXiv.

- Tulving, E. (1972). Episodic and semantic memory. In Tulving, E. and Donaldson, W., editors, *Organization of Memory*, pages 381–403. Academic Press.
- Vanmassenhove, E., Shterionov, D., and Gwilliam, M. (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.
- Vendler, Z. (1967). *Facts and Events*, pages 12–146. Cornell University Press, Ithaca.
- Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2019a). Superglue: A stickier benchmark for general-purpose language understanding systems. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019b). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.
- Webson, A. and Pavlick, E. (2022). Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.
- Weeds, J. and Weir, D. (2003). A general framework for distributional similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, page 81–88, USA. Association for Computational Linguistics.
- Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. (2022a). Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., brian ichter, Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. (2022b). Chain of thought prompting elicits reasoning in large

- language models. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.
- Weller, O., Marone, M., Weir, N., Lawrie, D., Khashabi, D., and Durme, B. V. (2023). "according to ..." prompting language models improves quoting from pre-training data. ArXiv.
- Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.
- Yarowsky, D. (1993). One sense per collocation. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.
- Zhang, C. and Weld, D. S. (2013). Harvesting parallel news streams to generate paraphrases of event relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1776–1786, Seattle, Washington, USA. Association for Computational Linguistics.
- Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., and Hashimoto, T. B. (2023). Benchmarking large language models for news summarization. ArXiv.