Multivalent Entailment Graphs for Question Answering

Nick McKenna[†] **Liane Guillou**[†] **Mohammad Javad Hosseini**[†‡*]
Sander Bijl de Vroe[†] **Mark Johnson**[§] **Mark Steedman**[†]
[†]University of Edinburgh [‡]Alan Turing Institute, UK [§]Oracle Digital Assistant
{nick.mckenna, javad.hosseini, sbdv}@ed.ac.uk
{lguillou, steedman}@inf.ed.ac.uk
mark.mj.johnson@oracle.com

Abstract

Drawing inferences between open-domain natural language predicates is a necessity for true language understanding. There has been much progress in unsupervised learning of entailment graphs for this purpose. We make three contributions: (1) we reinterpret the Distributional Inclusion Hypothesis to model entailment between predicates of different valencies, like DEFEAT(Biden, Trump) ⊨ WIN(Biden); (2) we actualize this theory by learning unsupervised *Multivalent Entailment Graphs* of open-domain predicates; and (3) we demonstrate the capabilities of these graphs on a novel question answering task. We show that directional entailment is more helpful for inference than non-directional similarity on questions of fine-grained semantics. We also show that drawing on evidence across valencies answers more questions than by using only the same valency evidence.

1 Introduction

We are reading a mystery about a dark and foreboding manor and have one question: "is Mr. Boddy dead?"[1] Our text 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 A kills B, then B is dead."

Entailment Graphs contain vertices of open-domain natural language predicates and entailments between them are represented as directed edges. Previous models learn predicates of a single *valency*, the number and types of arguments controlled by the predicate. Commonly these are binary graphs, which cannot model single-argument predicates like the entity states "is dead" or "is an author." This means they miss a variety of entailments in text that could be used to answer questions such as our example. 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 (Geffet and Dagan, 2005; Berant et al., 2010; Hosseini, 2021).

In this work the DIH is reinterpreted in a way which supports learning entailments between predicates of different valencies such as KILL(Mustard, Boddy) ⊨ DIE(Boddy). We extend the work of Hosseini et al. (2018) and develop a new *Multivalent Entailment Graph* (MGraph) where vertices may be predicates of different valencies. This results in new kinds of entailments that answer a broader range of questions including entity state.

We further pose a true-false question answering task generated automatically from news text. Our model draws inferences across propositions of different valencies to answer more questions than using same-valence entailment graphs. We also compare with several baselines, including unsupervised pretrained language models, and show that our directional entailment graphs succeed over non-directional similarity measures in answering questions of fine-grained semantics.

Advantageously, EGs are structures designed to be queried, so they are inherently explainable. This research is conducted in English, but as an unsupervised algorithm it may be applied to other languages given a parser and named entity linker.

2 Background

The 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." From here, research has moved in several directions. We study predicates, including verbs and phrases that apply to arguments.

*Now at Google Research.

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

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10758–10768
November 7–11, 2021.

Research in predicate entailment graphs has evolved from "local" learning of entailment rules (Geffet and Dagan, 2005; Szpektor and Dagan, 2008) to later work on joint learning of "globalized" rules, overcoming sparsity in local graphs (Berant et al., 2010; Hosseini et al., 2018).

These graphs 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 successfully used as these contextual features, but only predicates of the same valency are considered (e.g. binary predicates entail binary; unary entail unary), and further research computes additional edges in these same-valency graphs such as with link prediction (Hosseini et al., 2019). However, this leaves out crucial inferences that cross valencies such as the kill/die example, which are easy for humans. We generalize the DIH to learn entailments within and across valencies.

Typing is very helpful 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. enables generalized learning across instances and disambiguates word sense, e.g. "running a company" has different entailments than "running code."

We compare our model 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 we compare with on a task of predicate inference. We further test RoBERTa (Liu et al., 2019) 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 Multivalent Distributional Inclusion Hypothesis

We pose a new, multivalent interpretation of the DIH (the MDIH) which models the entailment of predicates across valencies. 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 event variable. Entailments about one or more of the arguments arise from their roles in this eventuality. We may infer that "Mr. Boddy died" due to his role as a direct object in the killing/murdering event. No other information is needed, including who murdered Mr. Boddy, where, or with what instrument. Boddy is dead simply because he was murdered. We build on this insight to develop the MDIH.

Here, a predicate is represented (as in §2) by features which are the argument tuples it appears with. We recognize a tuple as a proxy for a world event, e.g. VISIT(Obama, Hawaii) identifies one instance of a real VISIT event. Our method learns by tracking entity tuples across events in the world. The MDIH signals an entailment from a premise p to hypothesis h if, distributionally, subtuples of p are always found amongst tuples of h. Crucially, we allow h to drop in valency so that we learn entailments about subsets of p's arguments. We now formalize the MDIH and then illustrate with an example.

We define the argument tuple structures for a premise and hypothesis predicate:

$$P = \{(a_{k,1}, \ldots, a_{k,I}) \mid k \in \{1, \ldots, M\}\}$$
$$H = \{(b_{k,1}, \ldots, b_{k,J}) \mid k \in \{1, \ldots, N\}\}$$

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. We limit $J \leq I$, e.g. we learn about relations on *realized* entity subsets. We do not learn entailments to higher valencies (such as a unary entailing a binary) because additional arguments must be existential, not real. We leave this to future work.

To select argument subtuples from tuples in P, we define a vector of indices $\mathbf{j}$ 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. We define the Multivalent Distributional Inclusion Hypothesis:

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

Here $m : \mathbb{N}^J \to \mathbb{N}^J$ is a simple bijective mapping from argument indices of p to h. An example where m is needed for argument swapping is "x bought y" entails "y sold to x."

We illustrate by working the kill/die example on a hypothetical corpus. We might find that

KILL$(x, y) \models$ DIE(y) by trying $\mathbf{j} = [2]$ and $m([2]) = [1]$. We start with P, all 2-tuples of *killings*, and H, all 1-tuples of *dyings* and apply **j** and *m*. We may find 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, we observe that arg 2 of a *killing* often occurs elsewhere in the corpus with a *dying*, and thus we infer the entailment between predicates. Intuitively this is true for arbitrarily large valencies: MURDER(Mustard, Boddy, kitchen, candlestick) entails KILL(Mustard, Boddy) and both entail DIE(Boddy).

Though arguments may be dropped from the premise, they still influence entailments. This is because the MDIH tracks *eventualities*. "Person writing a book" is a different kind of event than "person writing software" with a different distribution of argument tuples, so we learn that the former entails being an author, while the latter entails being a programmer.

4 Learning Multivalent Graphs

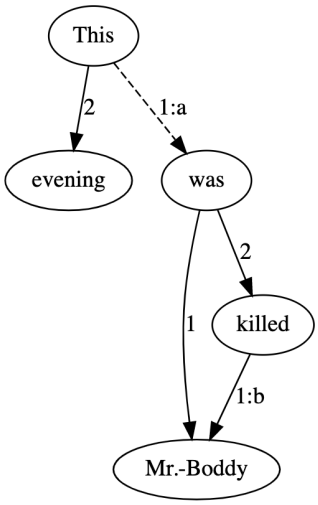
We define an Entailment Graph as a directed graph of predicates and their entailments, $G = (V, E)$. The vertices V are the set of predicates, where each argument has a type from the set of 49 FIGER base types $\mathcal{T}$, e.g. TRAVEL.TO(:person, :location) $\in V$, and :person, :location $\in \mathcal{T}$. 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 we expand V 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 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 (**BB** 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 (**BU**)
3. $u_i(x) \models u_j(x)$
   Unary entails unary (**UU**)

Predicates with valence > 2 are sparse in the text, but are also included in the MGraph by decomposing them into binary relations between pairs of entities. This is another application of our Multivalent DIH. We maintain argument roles, 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 we develop a method of local entailment rule learning using the MDIH. As in §2, the local step learns the initial directed edges of the entailment graph, which are further improved with global learning. Our local step learns entailments by machine-reading the NewsSpike corpus (2.3GB), 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."
$\Rightarrow$ KILL.2(Mr.-Boddy)

Figure 1: 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.

4.1 Extraction of Predicate Relations

Our pipeline processes raw article text into a list of propositions: predicates with associated typed arguments. We use the MoNTEE system (Bijl de Vroe et al., 2021) to extract natural language relations between entities from raw text [2]. This system first parses sentences using the RotatingCCG parser (Stanojević and Steedman, 2019) (Combinatory Categorial Grammar; Steedman, 2000) and then forms dependency graphs from the parses. Fi-

[2]We disable modality tagging in our experiments.

nally, it traverses these graphs to extract the relations, each consisting of a predicate and its arguments. Figure 1 shows an example dependency graph and the relation extracted from it. Arguments may be either named entities[3] or general entities (noun phrases). These entities are mapped to types by linking to their Freebase IDs (Bollacker et al., 2008) using AIDA-Light (Nguyen et al., 2014), and mapping the IDs to the 49 base FIGER types (Ling and Weld, 2012).

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 we must carefully apply linguistic knowledge to extract meaningful unary relations. We extract these neo-Davidsonian event cases:

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

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

4.2 Learning Local Graphs

In previous entailment graph research (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, for a typed predicate p with corresponding vector $\mathbf{v}$, $\mathbf{v}$ consists of features f_i which are the pointwise mutual information (PMI) of p and the argument pair $a_i \in \{(e_m, e_n) \mid e_m \in \mathcal{E}_{t_1}, e_n \in \mathcal{E}_{t_2}\}$. Here $t_1, t_2 \in \mathcal{T}$, and $\mathcal{E}_t$ is the subset of entities of type t. For example, the predicate BUILD(:company, :thing) might have some feature f_{37}, the PMI of "build" with argument pair (Apple, 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 symmetrical score, Lin Similarity (Lin, 1998), which downweights infrequent predicates that cause spurious false positives.

In this work we compute local binary graphs following Hosseini et al. (2018) and leverage the new MDIH to compute additional entailments for unaries and between valencies. To do this we compute a vector for each argument slot respecting its position in the predicate. For a predicate p, a slot vector $\mathbf{v}^{(s)}$ for $s \in \{1, 2\}$ consists of features $f_i^{(s)}$. We define $\tau(p, s) = t$, the type of slot s in predicate p. Each $f_i^{(s)}$ is the PMI of p and the argument in slot s, $a_i^{(s)} \in \mathcal{E}_t$. Slot vectors are computed for the slot in unary relations and both slots in binaries. Each slot vector for p has size $|\mathbf{v}^{(s)}| = |\mathcal{E}_t|$, the number of entities in the data with the same type t.

Continuing the example, we calculate two vectors 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. For instance, DEFEAT(:person1, :person2) ⊨ BE.WINNER(:person1) [4] is learned by comparing the slot 1 vector of DEFEAT with the slot 1 vector of BE.WINNER. If the entities who have defeated someone are usually found amongst the entities who are winners then we get a high BInc score, indicating *defeat* entails that its subject *is a winner*.

Figure 2 illustrates a Multivalent Graph. This

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

[4] Here we number the typed arguments for demonstration to show which :person argument is in the entailment.

includes Bivalent Graphs which contain the entailments of binary predicates (BB and BU edges), and separate Univalent Graphs which contain the entailments of unary predicates (only UU edges, since we do not allow a unary to entail a binary). We follow previous research and learn separate disjoint subgraphs for each typing, up to $|\mathcal{T}|^2$ bivalent and $|\mathcal{T}|$ univalent subgraphs given enough data. For example, we learn a bivalent (:person, :location) graph containing 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. We learn entailments from unaries (UU) in separate 1-type univalent graphs. This efficiently learns one set of entailments for each unary, but allows them to be freely entailed by higher-valency predicates, e.g. binaries.

Bivalent graphs point transitively into univalent graphs. In Figure 2, DEFEAT(:person1, :person2) ⊨ BE.WINNER(:person1) in the person-person graph. E.g. further entailments of BE.WINNER(:person) are in the person univalent graph.

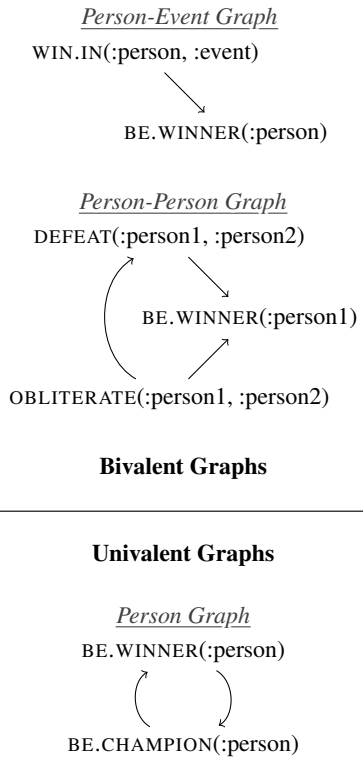


Figure 2: Bivalent graphs model entailments from binary predicates to equal- and lower-valency predicates (binary and unary). Univalent graphs model entailments from unaries to equal-valency unary predicates.

4.3 Learning Global Graphs

Local learning of entailments suffers from sparsity issues which can be improved by further learning of "global" graphs. We use 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. The cross-graph constraint additionally encourages compatible predicates across different typed graphs to share entailment patterns.

We apply global learning to bivalent graphs and separately to univalent graphs. Globalization is valency-agnostic, using just the common structures between predicates, so bivalent graphs can use BB and BU edges to optimize binary predicate entailments. Final graph size statistics are in Table 1.

Valency	Vertices	Edges
Bivalent	938K Binary	94M BB / 30M BU
Univalent	36K Unary	3.6M UU

Table 1: We learn 546 typed bivalent subgraphs which contain entailments of binary predicate antecedents (BB and BU); and 37 typed univalent subgraphs which contain entailments of unary predicates (UU).

5 Evaluation: Question Answering

We pose an automatically generated QA task to evaluate our model explicitly for directional inference between binary and unary predicates, as we are not aware of any standard datasets for this problem. Our 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: we 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 MGraph 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 we automatically generate false events to use as negatives, which are designed to mimic real, newsworthy events. The remaining news text is used to answer the questions. We at-

tempt to make every question answerable, but since they 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: with more kinds of entailment, the MGraph 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, to be published separately. 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. We generate questions balanced to a ratio of 50% binary questions / 50% unary; and within each 50% positive / 50% negative. Table 2 shows a sample from the dev set. We generate 34,394 questions on the test set: 17,256 unary questions and 17,138 binary.

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). We ask yes-no questions about events drawn from the partition, and the news text within this 3-day window is used as evidence to answer them. We ask questions as if happening presently in this time window to control for the variable of time, so we can ask ambiguous questions 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 multiple news stories about an event appear together, increasing the chances of finding question answers. Within each partition we do relation extraction in a process mirroring §4.1.

2. Selecting Positives. We adapt a selection process from Poon and Domingos (2009) to choose good questions which are interesting to a human and answerable from the partition text. First, we identify repeated entities that star in the events of the articles; these will yield interesting questions as well as ample textual evidence for answering them. In each partition we count the mentions of each entity pair (from binary propositions) and single entities (from unary and binary ones). The most frequent entities and pairs mentioned more than 5 times in the partition are selected. Predicates which are mentioned across the entire news corpus 10 times or fewer are filtered out; we assume those remaining are popular to report in news and thus are interesting to a human questioner. We randomly select propositions featuring both a star entity and predicate to use as questions, and remove them 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 we derive negative questions from the selected positives using linguistic relations in WordNet (Fellbaum, 1998). We assume 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, we extract "burn" as a troponym of "hurt" and the phrase "inherit from" as a troponym of "receive from." We therefore expect that these specific relations will be untrue of the argument tuple in question and may be used as negatives. We also considered antonyms and other WordNet relations, but these are much sparser in English and have low coverage.

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, which actually do occur in real news text. See Table 2 for a sample of questions. In the error analysis we find these negatives to be of good quality: they are uncommonly inferable from the text, accounting for a small percentage of false positives.

5.2 Question Answering Models

In each partition, models receive factual propositions extracted from 3 days of news text to use

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 2: A sample of dev set questions.

as evidence for answering true-false questions. A model scores how strongly it can infer the question proposition from each evidence proposition, and we take the maximum score as the model confidence of a "true" answer.

Exact-Match. Our text is multi-source news articles, so world events are often discussed multiple times in the data, even with the same phrasing. We compute an "exact-match" baseline which shows how many questions can be answered from an exact string match in the text; the rest require inference.

Binary Entailment Graph. Our BB 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. [5]

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 we assume it is false (score of 0).

Multivalent Entailment Graph. The MGraph is made of 3 component models: (1) the BB model which uses binary evidence to answer binary questions; (2) the UU model which uses unary evidence to answer unary questions; and (3) the BU model which uses binary evidence to answer unary questions. The MGraph 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 predicts true, the overall model does too.

In some test instances the entity typer may make an error, and so we fail to find the question predicate in the typed subgraph. Similarly to Hosseini et al. (2018), in these cases we back off, querying all subgraphs for the untyped predicate and averaging the entailment scores found. We find 5% more unary questions and 18% more binaries.

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. We encode the question into a representation vector, and each evidence proposition with the same arguments. We compute the cosine similarity 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 we construct a simple natural language sentence from the proposition using its predicate and arguments and encode it with the language model. Our 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. We average all final hidden-state vectors from the model corresponding to the predicate, excluding those of the arguments. We test the basic BERT model and RoBERTa model, which has robustly pretrained on 160GB of text (76GB news).

PPDB. Though supervised, PPDB 2.0 (we use XXXL) (Pavlick et al., 2015) is a useful comparison as it is a large, well-understood resource for phrasal entailment. PPDB relations come from bilingual pivoting and are categorized using text-based features, which is very different from our argument-tracking method. We view PPDB as a kind of Entailment Graph with 9M predicate phrases (vertices) and 33M "Equivalence" and "ForwardEntailment" edges. We convert evidence and question propositions into a natural text format and extract a PPDB relation score from each evidence phrase to the question.

6 Question Answering Results

The models produce a gradation of judgement scores between 0 (false) and 1 (true). As in earlier work, we slide a classification threshold over the score range to produce a precision-recall curve for each model. Results are in Figure 3 (left).

[5] We test the MGraph on the Levy/Holt dataset of 18,407 questions for BB entailment (Levy and Dagan, 2016; Holt, 2018), and achieve similar results to Hosseini et al. (2018).

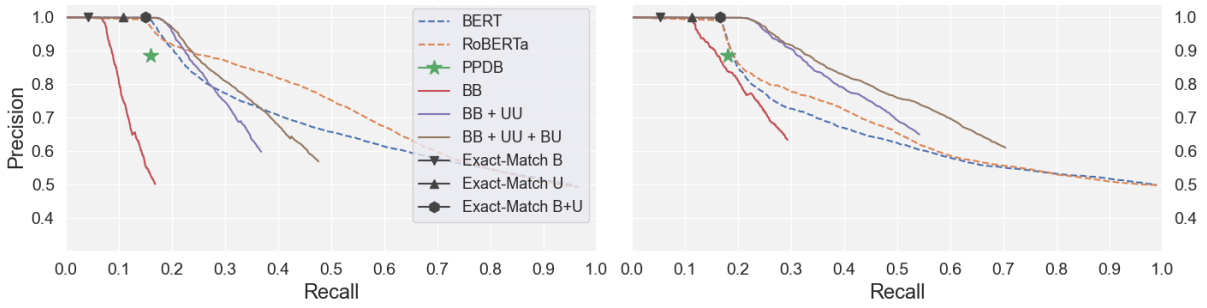


Figure 3: (Left) Overall performance on the QA task. (Right) performance on the filtered task. Note that BB, UU, and BU models may individually reach a max recall of 50% because they answer only binary or unary questions.

Multivalent graph performance is shown incrementally. The BB model can answer a portion of binary questions; the UU model can answer more unary questions; adding the BU model can answer still more unary questions using binary evidence. We observe successful inference of our kill/die example and others. "Obama was elected to office" affirms the question "Was Obama a candidate?" and "Zach Randolph returned" affirms "Did Zach Randolph arrive?"

Our 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 MGraph achieves ~3x this recall by drawing inferences.

Our model 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. We note that RoBERTa bests BERT due to extensive in-domain pretraining.

The BB model appears to struggle. In fact 90.5% of unary questions have a vertex in the graph, but only 64.1% of binaries do. The BB model frequently cannot answer questions because the question predicate wasn't seen in training. This difference is because binary predicates are more diverse so suffer more from sparsity: they are often multiword 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; Hosseini et al., 2018).

For an even comparison we create a filtered question set. From all questions we remove those without a vertex in the MGraph, then balance them as in §5, resulting in 20,519 questions (10,273 unary and 10,246 binary). This filtered test directly compares the models, since both the entailment graphs and the similarity models have a chance to answer all the questions. Results are shown in Figure 3 (right), with a very different outcome. Head-to-head, the MGraph offers substantially better precision across all recall levels. At 50% recall, the MGraph has 76% precision with RoBERTa at 65%.

	Unary Questions		**Binary Questions**	
Model	@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%	—
MGraph				
UU	96.5%	87.0%	—	—
BU	97.6%	90.4%	—	—
BB	—	—	100.0%	88.8%
	1245 Exact-Match		597 Exact-Match	

Table 3: The filtered test. Models rank question/answer pairs by confidence. We show accuracy on the K most confident predictions, at two points. PPDB doesn't answer enough questions to reach the @2000 cutoff, so we also compare at the smaller PPDB maximum.

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 BU model increases max recall from 54% to 70%.

Finally, we note PPDB's 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, this sparsity of edges useful for the task is likely because bilingual pivoting excels at detecting near-paraphrases, not relations between distinct eventualities, e.g. it can't learn "getting elected" entails "being a candidate." Advantageously, our method learns this open-domain knowledge by tracking entities across all the events

they participate in.

We show a breakdown of the filtered test results in Table 3. Models don't answer all the questions, so following Lewis and Steedman (2013) who design a similar QA task, we evaluate models on the accuracy of their K most confident predictions.

7 Error Analysis

We sample 300 false positives (100 for each model) and report analyses in Table 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 UU and BU models also suffer during relation extraction (parsing). When we fail to parse a second argument for a predicate we assume it only has one and extract a malformed unary, which can interfere with question answering (e.g. reporting verbs "explain," "announce," etc. which fail to parse with their quote). We also find relatively few poorly generated negatives, which are actually true given the text. In these cases the model finds an entailment which the authors judge to be correct.

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

Table 4: False positive analysis. Models predict entailments from the text (left) to generated negatives (right).

8 Conclusions

The MDIH is shown as an effective theory of unsupervised, open-domain predicate entailment, which crosses valencies by respecting argument roles.

Our multivalent entailment graph's performance has been demonstrated on a question answering task requiring fine-grained semantic understanding. Our method is able to answer a broader variety of questions than earlier entailment graphs, aided by drawing on evidence across valencies. We outperform baseline models including a strong similarity measure using unsupervised BERT and RoBERTa, 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.

We also noted a complementarity between unsupervised methods. Our symbolic graph method achieves high precision for learned predicates, while sub-symbolic neural models achieve high recall by generalizing to unseen predicates. Future work may leverage our MDIH signal to train a directional neural classifier and combine benefits.

Acknowledgments

This work was supported in part by ERC H2020 Advanced Fellowship GA 742137 SEMANTAX, and an Edinburgh and Huawei Technologies Research Centre award.

References

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

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

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

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 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.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ido Dagan, Lillian Lee, and Fernando CN Pereira. 1999. Similarity-based models of word cooccurrence probabilities. *Machine learning*, 34(1-3):43–69.

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Press.

Wayne Davis. 2019. Implicature. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, fall 2019 edition. Metaphysics Research Lab, Stanford University.

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

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

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

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

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

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

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

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

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

Omer Levy and Ido Dagan. 2016. Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 249–255, Berlin, Germany. Association for Computational Linguistics.

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

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*.

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

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

Claudia Maienborn. 2011. *Event semantics*, pages 802–829.

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

D.B. Nguyen, Johannes Hoffart, M. Theobald, and G. Weikum. 2014. Aida-light: High-throughput named-entity disambiguation. volume 1184.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 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.

F. Petroni, T. Rocktäschel, A. H. Miller, P. Lewis, A. Bakhtin, Y. Wu, and S. Riedel. 2019. Language models as knowledge bases? In *In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019*.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 1–10.

Miloš Stanojević and Mark Steedman. 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 Technologies, Volume 1 (Long and Short Papers)*, pages 228–239, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Idan Szpektor and Ido Dagan. 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.

Zeno Vendler. 1967. *Facts and Events*, pages 12–146. Cornell University Press, Ithaca.

Julie Weeds and David Weir. 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.

Congle Zhang and Daniel S. Weld. 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.