The Fortieth AAAI Conference on Artificial Intelligence (AAAI-26)

Why Isn't Relational Learning Taking Over the World?

David Poole
Dept of Computer Science, University of British Columbia
Vancouver, BC, Canada
https://www.cs.ubc.ca/~poole/

Abstract

Artificial intelligence seems to be taking over the world with systems that model pixels, words, and phonemes. The world is arguably made up, not of pixels, words, and phonemes but of entities (objects, things, including events) with properties and relations among them. Surely we should model these, not the perception or description of them. You might suspect that concentrating on modeling words and pixels is because all of the (valuable) data in the world is in terms of text and images. If you look into almost any company you will find their most valuable data is in spreadsheets, databases and other relational formats. These are not the form that are studied in introductory machine learning, but are full of product numbers, student numbers, transaction numbers and other identifiers that can't be interpreted naively as numbers. The field that studies this sort of data has various names including relational learning, statistical relational AI, and many others. This paper explains why relational learning is not taking over the world – except in a few cases with restricted relations – and what needs to be done to bring it to it's rightful prominence.

Introduction

AI has hit the news recently with models that learn to predict images, text, sound and video. Many have even speculated that such technologies will lead to artificial general intelligence (AGI), and be more intelligent than people.

In my AI courses, I often include the clicker question of Figure 1. Reading machine learning books and papers would lead one to assume the answer is A. Reading the press on generative AI would lead one to think the answer is B. Very few of the students choose A or B.

What is the real world made of?

A Features or random variables
B Words, pixels, phonemes . . .
C Entities and events (e.g., plants, people, diseases, lectures, university courses)
D Huh? There is a real world?

Figure 1: A clicker question

Learning and reasoning about entities, events and relations among them is central to AI:

> *The mind is a neural computer, fitted by natural selection with combinatorial algorithms for causal and probabilistic reasoning about plants, animals, objects, and people.*
>
> *– Steven Pinker (1997)*

By relational learning, I mean learning models that make probabilistic predictions about entities (things, objects, including events), their properties and relations among them. The aim is to model entities rather than modelling their manifestations in language or images.

Part of the theme of this paper is that we should not lose connection to the downstream task that a learned model is used for. Learning is usually not an end in itself, but is used for decision making. This could be for autonomous decision making, where probabilities and utilities are needed, or human-supported decision making where multiple scenarios can be evaluated by a human decision maker.

Relational Data

If you were to examine company data, environmental data, or government data – with few exceptions – you will find that the most valuable data not in natural language text or images but in spreadsheets, relational databases and other relational formats. This relational data is typically not the sort of tabular data used in introductory machine learning courses – for which standard algorithms such as gradient tree boosting (Chen and Guestrin 2016) work directly on – but is full of product numbers, student numbers, transaction numbers and other identifiers. These identifiers are often represented as numbers, but the numbers themselves don't convey meaning. For gradient tree boosting to be effective for such data, it needs to exploit relational structure (Natarajan et al. 2012).

The simplest form of relational data is exemplified by collaborative filtering datasets such as used in the Netflix challenge (Bell and Koren 2007) and Movielens datasets (Harper and Konstan 2015). Movielens contains *rating* datasets containing user-movie-rating-timestamp triples, where the user and movie are identifiers represented by arbitrary numbers. There are also tables for user properties and movie prop-

---

erties, including links into the Internet Movie Database (IMDB). An even simpler version is a *rated* dataset that ignores the actual ratings, and is just a user-movie dataset (perhaps with timestamps). To predict rating (or rated) from just the rating dataset, you can use latent properties (embeddings) of users and of movies to predict ratings. Representing latent properties as a vector, matrix factorization (Koren, Bell, and Volinsky 2009) learns about movies and users, without learning general knowledge. Another possible technique is to do a self-join, which can result in a user-user table or a movie-movie table.

Most real-world databases are much more complicated, with multiple relations.

When considering relational data, the following questions are often implicit and affect what is represented:

- Does the system know what exists? Most databases assume that they know what exists; things that exist are given identifiers, and what doesn't exist is ignored. In many real-world cases, we don't know what exists. Was there a person in the house last night? If so, who? Who threw the rock that broke the window? (It might have not been a rock or maybe no one threw it.)
- Do different identifiers denote different entities? This is known as the *unique names assumption* and features in logic programming and most databases. Identifying equality (two descriptions describing the same entity) arises in record linkage (Fellegi and Sunter 1969) and citation matching (Pasula et al. 2003). It is a big problem in medicine where drug dispensers need to decide whether someone asking for drugs is the same person who asked previously.

Knowledge Graphs

(Subject, verb, object) triples are the basis for knowledge graphs. The subject is an entity. The verb is a relation or property. The object is an entity or a datatype value, such as a number or a string. A collection of triples can be treated as a labelled directed graph where the entities and datatype values are nodes. There is an arc from the subject to the object, labelled with the verb. Triples are universal representations of relations: any relation can be represented as subject-verb-object triples (See e.g., Poole and Mackworth 2023, Section 16.1).

One large public knowledge graph is Wikidata (Vrandečić and Krötzsch 2014). Table 1 shows tables about the football (soccer) player Christine Sinclair. She is the player (man or woman) with the most goals in international play. Figure 2 shows part of the corresponding Wikidata knowledge graph. In Wikidata, Christine Sinclair is represented by the identifier Q262802. The country Canada is represented by the identifier Q16.

This shows two ways of converting relations to triples. The first is to group columns such as "Canada"-en, giving (Q16, name, "Canada"-en). Grouping columns does not work in general as there are exponentially many grouping of columns. It is only done in restricted cases.

The second way to convert relations to triples is to use (row, column, value) triples. The row can be a primary key if there is one, or is a new identifier – a *reified* entity. Reify means "make into an entity" (or "thingify"). In Figure 2, Q262802-3B1 and Q262802-9c5 are reified entities. The number of facts about a reified entity is the number of columns in the table, which is typically just a handful. To convert to the original relation, one can do a database join of the triples on the reified entities and project onto the other properties (which is how I generated the tables from the Wikidata triples). Reification does not lose information. It makes the representation more modular: other properties, including meta-properties (such as the author) can be easily added to a knowledge graph.

entity	name	language
Q262802	"Christine Sinclair"	en
Q16	"Canada"	en
Q1446672	"Portland Ferns"	en

member	team	start	#matches	#goals
Q262802	Q499946	2000	190	319
Q262802	Q1446672	2013	218	75

Table 1: Facts about football player Christine Sinclair (Q262802) in tabular form

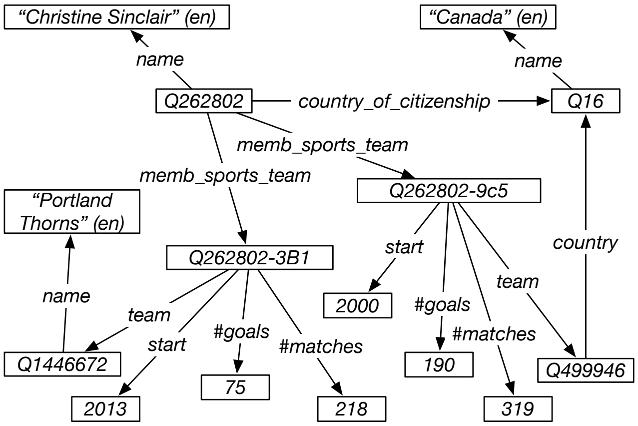

Figure 2: Part of the Wikidata knowledge graph about Christine Sinclair; from Figure 16.1 of Poole and Mackworth (2023)

Another way to convert relations into triples is to pair attributes, called the star-to-clique method (Wen et al. 2016). This, however, loses information. Figure 3 shows some examples of the sort of triples that arise from this method.

Note that graph learning (Hamilton 2020) and relational learning are isomorphic problems. A graph can be represented as a tail-head-label *arc* relation. Relations can be represented as knowledge graphs. These are complementary fields. It is not clear that a typical relational database is anything like a typical graph. Knowledge graphs tend to be much more heterogenous than graph benchmarks (Bechler-Speicher et al. 2025); the only general structure is that induced by reification. Many of the questions of interest are different. For example, there tends to be only a single knowledge graph so graph matching (Roy et al. 2025) is not really a problem of interest. Most of the issues in this paper do not apply to graph learning in general.

Datasets

One of the most used knowledge graphs in the literature is FB15k (Bordes et al. 2013), a large subset of Freebase. Freebase (Bollacker et al. 2008) was a free and open knowledge base developed from 2007-2016. It was acquired by Google, who donated the contents to Wikidata, and use another branch as the Google knowledge base. In building FB15k, Bordes et al. (2013) selected entities and relations that appeared in more than 100 triples. They also created a larger knowledge graph, FB1M, with the most frequently occurring 1 million entities, which has been used much less in subsequent research.

Figure 3 shows some of the triples in FB15k. The last two are obviously projections of relations that are more interesting, telling us who Noel Gallagher was married to (and probably when) and when Hannah Montana: The Movie was released in Egypt.

We[1] created English version of the test set that, together with the plausible examples that existing methods ranked highly, could be used to crowdsource ground truth. The ground truth could then be used to evaluate prediction methods. However, we decided determining whether a triple was true wasn't a skill people could do quickly and accurately, even when they had access to Wikipedia, Google and other online sources. Positive examples were straightforward, but determining something is false requires guessing.

Toutanova and Chen (2015) noticed that a predictor could do reasonably well on FB15k by just determining that two predicates were (approximately) equivalent or inverses. To determine whether a predictor was doing more than this, they created FB15k-237 with the inverse functions removed (it would remove one of the first two triples of Figure 3). FB15k-237 isn't better than FB15k; they test different things. Together they provide interesting evidence: the difference tells us whether a predictor can recognize inverses; the performance on FB15k-237 tells us whether the method can exploit other regularities. We would want a predictor that can determine inverses, but would like it to do more than that.

Including only most frequent entities means that there are *no* reified entities in the resulting triples. In Wikidata, over 98% of the entities appear as the subject of fewer than 10 triples (see Poole 2025, Appendix A). Training and evaluating on datasets where all such entities are removed will not lead to better predictors on more realistic datasets.

Big data often implies more entities, but less data, on average, about each entity. As entities are added, they start off as stubs with few triples, resulting in a long tail of entities about which little is known. As more detail is added about an entity, there are more associated reified entities with inherently few triples (see Poole 2025, Appendix A).

Another standard dataset is WN18 (Bordes et al. 2013), based on WordNet (Miller 1995), a large lexical database of English. The entities are words, and the relations include hyponym (subclass), hypernym (superclass), part_of and type, with loosely connected hierarchies for nouns, verbs, adjectives and adverbs. Most of the relations are transitive. I would not expect that words are prototypical entities.

Graph learning also has issues with standard benchmarks (Bechler-Speicher et al. 2025; Roy et al. 2025).

JF17K (Wen et al. 2016), FB-AUTO, M-FB15K (Fatemi et al. 2020) and RelBench (Robinson et al. 2024) are benchmarks that do not convert to triples. Such datasets are a start in building more realistic test sets.

Relational Predictions

As far as relational learning is concerned, it is important to distinguish two types of relational datasets.

In the *complete knowledge* or *closed world* assumption, statements not in a database are false. E.g., Airlines know the actual departure and arrival times of flights in the past, who is booked on a flight, and who showed up for boarding. They do not know who will show up tomorrow. When data is considered complete up to some time, the predictive problem is to predict the future from the past. Consider, for example, the problem of predicting the passenger load or whether I will miss a connection, of a trip from Vancouver BC to Boston MA on July 2 2026. This is a tricky prediction as July 2 is a special day – is one day after a holiday in Canada, and 2 days before one in USA – and 2026 is unusual in that many Canadians are boycotting USA and the FIFA World Cup is being held then; Vancouver and Boston are host cities. Rather than guessing, we should be able to make an informed prediction based on data.

Surprisingly, there is very little work on making predictions about the future from arbitrary complete databases. We can't add an airline reservation system to a learner and get it to predict future demand. It might be because most company databases are proprietary and include much metadata, such as the database schema. Individual companies such as Amazon and Netflix have huge datasets they are mining, but the techniques are proprietary and are undoubtedly tuned to the particular task, such as maximizing profit and engagement, rather than allowing general queries of what is predicted. Steck et al. (2021) describe techniques used by Netflix. Feature engineering (such as carefully chosen joins and projections) can created tables that methods such as XGBoost or deep learning can work on.

The alternative to the complete knowledge assumption is the *open-world assumption*: the data is not complete and there is missing data. Wikipedia contains a very small proportion of the true statements even about those entities chosen to be notable enough to be included. Building predictive models from such models has two main challenges:

- There are no negative statements.
- Missing data is not missing at random.

Databases do not store arbitrary relations. Some relations are stored and others are derived. Which relations to store is carefully designed (Codd 1970). Database designers choose

---

[1] Bahare Fatemi, Mehran Kazemi, Ali Mehr and Martin Wang. See https://cs.ubc.ca/~poole/res/FB15k-triples.txt for example triples for all 1344 predicates.

```
(A.S. Livorno Calcio, /soccer/football_team/current_roster,
        Forward (association football))
(Forward (association football), /soccer/football_roster_position/team,
        Cambridge United F.C.)
(California, /location/religion_percentage/religion, Methodism)
(Ambient music, /music/genre/artists, Portishead (band))
(Marriage, /people/marriage/spouse, Noel Gallagher)
(Hannah Montana: The Movie,
      /film/film_regional_release_date/film_release_region, Egypt)
```

Figure 3: Some of the triples in FB15k. The subject and object entities have their identifier replaced with the name. The verb (property) names have been abbreviated.

to store only true statements for some of the possible relations, selected to make the database small. They don't represent "did-not-buy" or "has-watched-movie-starring" as explicit relations, instead choose smaller relations to explicitly represent, and use queries and rules to infer other ones.

The lack of negative statements is a feature of databases, but with the open world assumption, it makes predicting probabilities problematic. There is no way to estimate probabilities without external information about the entities and relations not represented (e.g., knowing that people are typically married to zero or one person, or that celebrities are richer than average). This meta information cannot be assumed to be explicit in a relational database.

Predicting property values, when the set of values is a constant set, such as the positions of football players, may be amenable to techniques for tabular learning, such as gradient tree boosting, following feature engineering. However, when making predictions where the values are entities (called *link prediction*) this is infeasible.

Training

A standard way to make probabilistic predictions is to use a softmax or sigmoid output and to optimize log loss (an observation with predicted probability p incurs a cost of $\log p$, the expectation of which is cross entropy) on the training set with some regularization.

In relational datasets with no negative statements, the prediction that optimizes log loss is to predict everything is *true*, with a loss of zero on any data set that only includes positive statements. However for a test set that includes statements that should be false, this prediction has an infinite test loss. In training, the lack of negative statements is generally handled by a form of contrastive learning: adding random triples and using these as negative examples (the few that might be correct are noise). The number of negative versus positive examples provides input that is just fiction. Unlike a prior probability, this input is not overwhelmed by observed data.

Relational models differ from standard tabular models in two main ways:

- parameter sharing or weight tying; namely *logical variables* and *quantification* in logic approaches (Poole 1993; De Raedt, Kimmig, and Toivonen 2007; Domingos and Lowd 2009), *exchangeability*[2] (Niepert and Van den Broeck 2014), or the *convolutional* in convolutional graph neural networks (Schlichtkrull et al. 2018).
- aggregation, where a property of an entity or relation among entities depends on other entities, such as predicting the gender or age of a person from the movies they have watched.

When considering relational data as a graph, the neighbours of an entity are typically either all entities or those known to be related by existing relations. When considering entities n hops away from an entity, the number of entities is exponential in n. This is, of course, bounded by the number of entities. Having all entities as neighbours of an entity allows for richer hypotheses about which pairs are related, but impacts on what methods are appropriate. With large knowledge bases, algorithms that are quadratic in the number of entities are impractical. Learning needs to be done in linear time, at most, and inference in sublinear time. Fortunately many questions can be approximated by sampling.

There are a myriad of, often ingenious, learning techniques used for relational data which I won't go into.

Evaluation

The evaluation of a knowledge graphs depends on what is asked. To predict whether a random triple is in Wikidata, answering "no" is over 99.99999999995% accurate[3]. Accuracy of random triples is not a reasonable measure. Most random triples are nonsense, which no one would ask and could be answered just by knowing the types of the entities.

One could evaluate predictions using log loss, as in the training outlined above. Those papers that report log loss typically do the following: Given a test triple, mutate it by replacing the subject or object to create a negative example. The predictor is then evaluated on these positive and negative examples. (The examples generated to be negative that

---

[2] Identifiers can be exchanged and the probability distribution should not change. Exchangeability of identifiers does not imply exchangeability of random variables (de Finetti 1974).

[3] Wikidata has about 1.65 billion triples on 117 million items and 290,000 properties (https://grafana.wikimedia.org/d/000000175/wikidata-datamodel-statements). With no domain or range constraints in Wikidata, any entity can be the subject or object of any property. (Let's ignore strings and other datatypes as values, as there are too many possible values). There are thus $(1.17e8)^2{*}2.9e5 \approx 4e21$ possible triples, of which $1.6e9/4e21 = 4e{-}13$ are in Wikidata.

are actually true are just noise.) Unfortunately the probabilities are just fiction, reflecting the statistics of the knowledge base, not the statistics of the world.

The most common way to evaluate a knowledge graph predictor is to use ranking: given test triple (s, v, o), ask the predictor to rank the entities related to a given subject s and verb v, i.e., to predict which triples of the form $(s, v, ?)$ are true (and similarly to rank the subjects for a given verb and object). Given a total ordering of predictions $o_1, o_2, o_3, \ldots,$ the rank is the i such that $o = o_i$. In filtered prediction (Bordes et al. 2013), the o_i that are known to be true (because the corresponding triple is in the training or test set) are removed from the ordering.

The mean rank is one possible measure but is dominated by a few very large ranks, and is not used much any more. A common measure is *hit-at-k* for some integer k (such as 1, 3 or 10), the proportion of test cases where the rank of the correct answer is k or less. The k in hit-at-k seems like an arbitrary parameter. A method that is common and seems to not have arbitrary parameters is the mean reciprocal rank (MRR) (Radev et al. 2002). This weights the kth rank as $1/k$. It can be interpreted as an agent who always considers the first item, considers the second item half the time, the 3rd item 1/3 of the time on so on. However this interpretation is problematic (see Poole 2025, Appendix B).

Ranking has many problems, including:

- Some questions cannot be asked, such as “who is the Pope married to?” (where the answer is “nobody”).
- Asking $(s, v, ?)$ leaks information about the test set. Asking who e is married to tells us that e is a person, an adult and not the Pope. Knowing that someone rated a movie, as can be obtained from asking for a prediction for the rating of a movie, is valuable information that can be exploited to produce better predictions. One could imagine a task that provides information about an entity, as a prompt to an LLM might, and asks questions about that entity. For many tasks, questions should not provide information. Whether leaking information about the test set is appropriate depends on the downstream application.
- Some queries are trivial; for example, with fewer than 10 football (soccer) positions in FB15k and most teams having all of them, asking for a position in a team (see Figure 3), any list of all positions will give a hit-at-10.
- Some queries are almost impossible, for example asking which soccer team has someone playing *forward* – one of the queries of the FB15k test set – means listing all soccer teams. Guessing which one the query is about is almost impossible. You would expect that an omniscient agent – one who knows everything – would go well in any reasonable evaluation. However, an omniscient agent who knows all soccer teams and all players and their positions, would not be able to guess what team the query was about.
- The actual probabilities are lost. One predictor might be very sure of one answer, and have to fill in the rest. Another predictor might have a handful of plausible answers, they are not sure about. These predictors could produce the same ordering. There is no penalty for being overconfident or reward for being sure (and correct) or for actually specifying uncertainty, even though this is valuable information for making decisions.
- It loses sight of the downstream task. Considering top-k is reasonable for cases where the results can be checked, such as in movie recommendation, web search or suggesting to a doctor the k most likely diagnoses so they can be reminded of something they may have overlooked. It seems less useful when just presenting k possible answers for which there is no easy way to check which, if any, are correct. State-of-the-art methods that combines existing models (Li et al. 2024; Jeon, Lim, and Suk Choi 2024) have hit-at-10 rates on FB15k-237 of 55.8%. and 52.58% respectively, which does not seem very useful for any real-world task where correctness cannot be easily verified.

Looking Forward

There are many success stories from instances of relational learning from protein prediction (Jumper et al. 2021), where the atoms are entities and relations includes *bonds*, to predicting congestion on roads and at intersections for route-planning (Delling et al. 2015) to the combination of ML and the internet of things in agriculture (Sharma and Shivandu 2024; Mureithi 2024). These build models particular to the application. Generalizing particular models so we can make predictions from arbitrary relational datasets and queries is still in its infancy, despite years of research. This provides a great opportunity for researchers to make fundamental contributions. For relational learning to reach its full potential, the following (at least) need to be done:

We need to consider real public datasets on problems that someone cares about. Unfortunately someone caring means that the datasets are often valuable and so propriety. Relational learning is ideal for making predictions on electronic health records (Weiss et al. 2012; Natarajan et al. 2013), but such records are confidential and cannot be made public for other researchers. Many of the current public datasets are public because no one (currently) cares about them. One class of datasets that should be exploited more is environmental datasets produced and published by government agencies, such as European Environmental Agency[4]. These often contain all measurements taken before a certain time, even if they are not complete with respect to related causes of these. These datasets, however, are often overwhelming for researchers who want to concentrate on the general problem of relational learning, and are much more difficult to use than the datasets in the ML repositories (but see Tschalzev et al. 2025). What is useful to actually predict from these is not obvious for non-domain experts. We need to interact with diverse sets of domain experts to find useful tasks.

The main use of predictions is make decisions. One of the greatest inventions of the 20th century was utility theory (Neumann and Morgenstern 1953): utilities are measures of preference that can be combined with probabilities. To make decisions, we thus need probabilistic predictions and utility

---

[4] See e.g., https://www.eea.europa.eu/en/datahub

functions. For relational models, we need better evaluation to handle the predicament that we want to predict probabilities but only have positive statements. One solution is to include and exploit meta-information, such "there are no more triples involving this subject and verb". There is a continuum between closed-world and open-world assumptions; datasets are often complete for some narrow part of the world.

For incomplete knowledge bases, we need to explicitly consider why data is missing. Having test examples that are a random subset of all of the triples, does not provide a surrogate to real-world problems where data is invariably not missing at random (Rubin 1976). I doubt there are any Taylor Swift officially released albums missing from Wikidata, but for most of the recording artists on Wikidata most of their recordings are missing. Probabilistic models of missing data (Marlin et al. 2011; Mohan, Pearl, and Tian 2013) tend to be heavyweight, requiring detailed modelling of why data is missing. This is difficult when virtually all of the triples are missing. We made an attempt to build a lightweight model of missing data (Poole, Mehr, and Wang 2020) with the aim of extended it to relational models.

When predicting entities, rather than assuming that the correct answer is one of the known entities, there are three possible types of answers: one or more of the known entities, an entity or entities not represented, or that there is no entity. Consider asking for birth mother. For most people in a knowledge base the answer is "someone not represented". Asking for a child of a person, a person may have (perhaps multiple) children represented, children not represented (or both) or have no children. Similarly, asking for an event that caused something, it might be a known event, an unknown event or there might no event. An unknown event may need to be given an identifier to give it properties. A prediction should not be judged on the actual identifier assigned.

We need to be clear about whether we want to predict general knowledge or learn about the entities in the training set. Unless the test sets are carefully designed, there will tend to be a bias towards learn about just one of these. Many models such as probabilistic logic programs (Poole 1993; De Raedt, Kimmig, and Toivonen 2007) and convolutional graph neural networks (Schlichtkrull et al. 2018) learn what looks like general knowledge that is then be applied to the training set to predict properties and relations of the entities in training set. Tests sets from the same dataset do not test whether these models generalize to new populations or related domains.

To have statistics, we need to combine the data in at least one dimension. E.g., we could ask for the average congestion at a particular intersection at a particular time, or the average congestion at intersections on a particular date and time, but not at at a particular intersection at a particular date and time, as there is at most a single data point. There could be latent features (embeddings) for intersections and dates, but some intersections and dates are peculiar and it might be better to learn about particular entities and not just general knowledge.

Many methods rely on embeddings/latent features, often fixed-sized embeddings. It seems bizarre to have the same size embeddings for the USA and the relationship between Sinclair and the Portland Ferns (Figure 2). Somehow we need the embedding/latent complexity to depend on the complexity of (the information about) the entity.

Aggregation (e.g., predicting the gender of a user from the movies they watched) is the Achilles heel of relational learning. Most models either have a built-in aggregation such as noisy-or for probabilistic logic programs (Poole 1993) or logistic regression for weighted logical formula models (Domingos and Lowd 2009; Kazemi et al. 2014), use a distribution of related entities in relational attention (Velickovic et al. 2018; Wang et al. 2019) or use explicit operations such a sum or mean (Schlichtkrull et al. 2018). They are problematic theoretically (e.g., asymptoticly) (Poole et al. 2014; Buchman and Poole 2015; Adam-Day et al. 2024) and in practice (Kazemi et al. 2017). They need to work with no related entities and as the number of related entities goes to infinity. The models either implicitly assume the related entities provide independent evidence (as do sum, noisy-or and logistic regression), or act the same as if there were one or few related entities (as do max, average and attention). Determining whether the evidence is independent (so can be accumulated) or is dependent (in which case extra evidence can be discounted) is difficult, and invariably ignored.

To realize the full potential of relational learning, we need to combine the information from *all* datasets, and build predictive models from these. Combining multiple heterogenous datasets is the challenge of the semantic web (Berners-Lee, Hendler, and Lassila 2001). Building predictive models from heterogenous data might seem impossible, but we know how to; it's called *science*. To combine all recorded experimental and observational data to make better predictions, we need at least ontologies[5] (to allow semantic interoperability) and provenance information (Gil et al. 2016; Sikos, Seneviratne, and McGuinness 2021), including what was manipulated, what was controlled for, what was observed and when, how was what to record decided (modelling the missing information), who manipulated the data, etc. Science involves building hypotheses, typically from multiple datasets, that persist, ready to be revised or rejected as new evidence comes in. Hypotheses can be about anything; from the effects of global warming, to what foods are edible, to the traffic in Vancouver. There needs to be multiple hypotheses for a dataset, including the null hypothesis (the data is noise) and hypotheses that some data is erroneous or entirely fictitious (to counter misinformation). Multiple hypotheses may need to be combined to make a prediction for an actual case (Poole 2012). With an interconnected world, we don't need to make conclusions from individual experiments, but can learn from everyones experiences, which should enable more accurate hypotheses.

Maybe relational learning in its full generality is impossible, and building separate applications for each domain is the best we do. The only way to determine this is to try to build models that go across domains.

---

[5] Scientific ontologies include Open Biological and Biomedical Ontology http://obofoundry.org, WHO Family of International Classifications https://www.who.int/standards/classifications, SMOMED-CT healthcare ontology https://www.snomed.org

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). Thanks to Sriraam Natarajan for comments on an earlier version.

References

Adam-Day, S.; Benedikt, M.; Ceylan, İ. İ.; and Finkelshtein, B. 2024. Almost Surely Asymptotically Constant Graph Neural Networks. In *38th Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Bechler-Speicher, M.; Finkelshtein, B.; Frasca, F.; Müller, L.; Tönshoff, J. M.; Siraudin, A.; Zaverkin, V.; Bronstein, M.; Niepert, M.; Perozzi, B.; Galkin, M.; and Morris, C. 2025. Position: Graph Learning Will Lose Relevance Due To Poor Benchmarks. In *International Conference on Machine Learning (ICML)*.

Bell, R. M.; and Koren, Y. 2007. Lessons from the Netflix Prize Challenge. *SIGKDD Explor. Newsl.*, 9(2): 75–79.

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, May: 28–37.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM ICMD*.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, volume 26.

Buchman, D.; and Poole, D. 2015. Representing Aggregators in Relational Probabilistic Models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.

Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In *KDD '16: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Codd, E. F. 1970. A relational model of data for large shared data banks. *Commun. ACM*, 13(6): 377–387.

de Finetti, B. 1974. *Theory of Probability*. Wiley.

De Raedt, L.; Kimmig, A.; and Toivonen, H. 2007. ProbLog: A probabilistic Prolog and its application in link discovery. In *20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2462–2467.

Delling, D.; Goldberg, A. V.; Pajor, T.; and Werneck, R. F. 2015. Customizable Route Planning in Road Networks. *Transportation Science*, 51(2): 566–591.

Domingos, P.; and Lowd, D. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool.

Fatemi, B.; Taslakian, P.; Vazquez, D.; and Poole, D. 2020. Knowledge Hypergraphs: Prediction Beyond Binary Relations. In *29th International Joint Conference on Artificial Intelligence (IJCAI)*, 2191–2197.

Fellegi, I.; and Sunter, A. 1969. A Theory for Record Linkage. *Journal of the American Statistical Association*, 64(328): 1183–1280.

Gil, Y.; David, C. H.; Demir, I.; Essawy, B. T.; Fulweiler, R. W.; Goodall, J. L.; Karlstrom, L.; Lee, H.; Mills, H. J.; Oh, J.-H.; Pierce, S. A.; Pope, A.; Tzeng, M. W.; Villamizar, S. R.; and Yu, X. 2016. Towards the Geoscience Paper of the Future: Best Practices for Documenting and Sharing Research from Data to Software to Provenance. *Earth and Space Science*, 3.

Hamilton, W. L. 2020. *Graph Representation Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer Cham.

Harper, F. M.; and Konstan, J. A. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, 5(4).

Jeon, H.; Lim, Y.; and Suk Choi, Y. 2024. A Relation-Specific Entropy-Based Ensemble Approach for Knowledge Graph Embedding. *IEEE Access*, 12: 164652–164660.

Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; and Hassabis, D. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873): 583–589.

Kazemi, S. M.; Buchman, D.; Kersting, K.; Natarajan, S.; and Poole, D. 2014. Relational Logistic Regression. In *14th International Conference on Principles of Knowledge Representation and Reasoning (KR-2014)*.

Kazemi, S. M.; Fatemi, B.; Kim, A.; Peng, Z.; Tora, M. R.; Zeng, X.; Dirks, M.; and Poole, D. 2017. Comparing Aggregators for Relational Probabilistic Models. In *UAI Workshop on Statistical Relational AI*.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*, 42(8): 30–37.

Li, R.; Li, C.; Shen, Y.; Zhang, Z.; and Chen, X. 2024. Generalizing Knowledge Graph Embedding with Universal Orthogonal Parameterization. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 28040–28059. PMLR.

Marlin, B. M.; Zemel, R. S.; Roweis, S. T.; and Slaney, M. 2011. Recommender Systems, Missing Data and Statistical Model Estimation. In *22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2686–2691.

Miller, G. A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11): 39–41.

Mohan, K.; Pearl, J.; and Tian, J. 2013. Graphical models for inference with missing data. In *Advances in Neural Information Processing Systems*, volume 26, 1277–1285.

Mureithi, C. 2024. High tech, high yields? The Kenyan farmers deploying AI to increase productivity. *The Guardian*.

Natarajan, S.; Kersting, K.; Ip, E.; Jacobs, D.; and Carr, J. 2013. Early Prediction of Coronary Artery Calcification Levels Using Machine Learning. In *AAAI conference on Innovative Applications in AI (IAAI)*.

Natarajan, S.; Khot, T.; Kersting, K.; Gutmann, B.; and Shavlik, J. 2012. Gradient-based Boosting for Statistical Relational Learning: The Relational Dependency Network Case. *Machine Learning Journal*, 86(1): 25–56.

Neumann, J. V.; and Morgenstern, O. 1953. *Theory of Games and Economic Behavior*. Princeton University Press, third edition.

Niepert, M.; and Van den Broeck, G. 2014. Tractability through Exchangeability: A New Perspective on Efficient Probabilistic Inference. In *Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, 2467–2475.

Pasula, H.; Marthi, B.; Milch, B.; Russell, S.; and Shpitser, I. 2003. Identity uncertainty and citation matching. In *Advances in Neural Information Processing Systems*, volume 15.

Pinker, S. 1997. *How the Mind Works*. Norton.

Poole, D. 1993. Probabilistic Horn Abduction and Bayesian Networks. *Artificial Intelligence*, 64(1): 81–129.

Poole, D. 2012. Foundations of model construction in feature-based semantic science. *Journal of Logic and Computation*, 23(5): 1081–1096.

Poole, D. 2025. Why Isn't Relational Learning Taking Over the World? (with Appendices). arXiv:2507.13558.

Poole, D.; Buchman, D.; Kazemi, S. M.; Kersting, K.; and Natarajan, S. 2014. Population Size Extrapolation in Relational Probabilistic Modelling. In Straccia, U.; and Cali, A., eds., *8th International Conference on Scalable Uncertainty Management (SUM)*, LNAI 8720, 292–305.

Poole, D.; Mehr, A. M.; and Wang, W. S. M. 2020. Conditioning on "and nothing else": Simple Models of Missing Data between Naive Bayes and Logistic Regression. In *ICML Workshop on the Art of Learning with Missing Values (Artemiss)*.

Poole, D. L.; and Mackworth, A. K. 2023. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 3rd edition.

Radev, D. R.; Qi, H.; Wu, H.; and Fan, W. 2002. Evaluating Web-based Question Answering Systems. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*.

Robinson, J.; Ranjan, R.; Hu, W.; Huang, K.; Han, J.; Dobles, A.; Fey, M.; Lenssen, J. E.; Yuan, Y.; Zhang, Z.; He, X.; and Leskovec, J. 2024. RelBench: A Benchmark for Deep Learning on Relational Databases. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024) Datasets and Benchmarks Track*.

Roy, I.; Meher, S.; Jain, E.; Chakrabarti, S.; and De, A. 2025. Position: Graph Matching Systems Deserve Better Benchmarks. In *International Conference on Mache Learning (ICML)*.

Rubin, D. B. 1976. Inference and missing data. *Biometrika*, 63(3): 581–592.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling Relational Data with Graph Convolutional Networks. In Gangemi, A.; Navigli, R.; Vidal, M.-E.; Hitzler, P.; Troncy, R.; Hollink, L.; Tordai, A.; and Alam, M., eds., *The Semantic Web*, 593–607. Cham: Springer International Publishing. ISBN 978-3-319-93417-4.

Sharma, K.; and Shivandu, S. K. 2024. Integrating artificial intelligence and Internet of Things (IoT) for enhanced crop monitoring and management in precision agriculture. *Sensors International*, 5: 100292.

Sikos, L.; Seneviratne, O.; and McGuinness, D. L. 2021. *Provenance in Data Science: From Data Models to Context-Aware Knowledge Graphs*. Springer.

Steck, H.; Baltrunas, L.; Elahi, E.; Liang, D.; Raimond, Y.; and Basilico, J. 2021. Deep Learning for Recommender Systems: A Netflix Case Study. *AI Magazine*, 42(3): 7–18.

Toutanova, K.; and Chen, D. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, 57–66. Beijing, China: Association for Computational Linguistics.

Tschalzev, A.; Purucker, L.; Lüdtke, S.; Hutter, F.; Bartelt, C.; and Stuckenschmidt, H. 2025. Unreflected Use of Tabular Data Repositories Can Undermine Research Quality. arXiv:2503.09159.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y.; et al. 2018. Graph attention networks. In *6th International Conference on Learning Representations (ICLR)*.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10): 78–85.

Wang, X.; He, X.; Cao, Y.; Liu, M.; and Chua, T.-S. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, 950–958. New York, NY, USA: Association for Computing Machinery. ISBN 9781450362016.

Weiss, J.; Natarajan, S.; Peissig, P.; McCarty, C.; and Page, D. 2012. Statistical Relational Learning to Predict Primary Myocardial Infarction from Electronic Health Records. *AI Magazine*.

Wen, J.; Li, J.; Mao, Y.; Chen, S.; and Zhang, R. 2016. On the representation and embedding of knowledge bases beyond binary relations. In *IJCAI*.