A Comparative Study of Distributional and Symbolic Paradigms for Relational Learning

Sebastijan Dumančić1, Alberto García-Durán2 and Mathias Niepert3
1KU Leuven, Belgium
2EPFL, Switzerland
3NEC Labs, Germany
sebastijan.dumancic@cs.kuleuven.be, alberto.duran@epfl.ch, mathias.niepert@neclab.eu

Abstract

Many real-world domains can be expressed as graphs and, more generally, as multi-relational knowledge graphs. Though reasoning and learning with knowledge graphs has traditionally been addressed by symbolic approaches such as Statistical relational learning, recent methods in (deep) representation learning have shown promising results for specialised tasks such as knowledge base completion. These approaches, also known as distributional, abandon the traditional symbolic paradigm by replacing symbols with vectors in Euclidean space. With few exceptions, symbolic and distributional approaches are explored in different communities and little is known about their respective strengths and weaknesses. In this work, we compare distributional and symbolic relational learning approaches on various standard relational classification and knowledge base completion tasks. Furthermore, we analyse the complexity of the rules used implicitly by these approaches and relate them to the performance of the methods in the comparison. The results reveal possible indicators that could help in choosing one approach over the other for particular knowledge graphs.

1 Introduction

Recent years have created a dichotomy in the field of Statistical relational learning (SRL), concerning machine learning with relational data which contains instances and their mutual relationships. The symbolic paradigm [Getoor and Taskar, 2007; De Raedt et al., 2016], leveraging the representational and reasoning capacity of first-order logic to compactly represent relational data, has typically dominated the field. Inspired by the success of (deep) representation learning [Goodfellow et al., 2016], an alternative distributional paradigm has emerged, with examples in Knowledge graph embeddings [Nickel et al., 2016] and Graph neural networks [Hamilton et al., 2017]. These distributional methods re-represent relational data as vectors and/or matrices in the Euclidean space so that the standard feature-based machine learning methods can be used.

The two research directions have largely been developed in isolation and little understanding is currently available on their respective advantages, even though the community has recognised the benefits of both [Minervini et al., 2017; Rocktäschel and Riedel, 2017; Manhaeve et al., 2018; Evans and Grefenstette, 2018]. Moreover, they focus on different tasks and employ different evaluation metrics. Distributional methods focus on the knowledge base completion (KBC) task measuring the performance through the ranking of proposed entity completions. Symbolic methods, on the other hand, focus on learning from small relational data with complex forms of logical reasoning and evaluate the performance through the accuracy of predictions.

Contrasting the paradigms upfront, they exhibit a stark contrast on three dimensions: reasoning capabilities, scalability and interpretability. By relying on logic, the symbolic methods are capable of complex reasoning patterns and are flexible enough to answer any query over a domain (without a need to commit to a predefined target); this is sacrificed with distributional methods [Trouillon et al., 2019]. The distributional methods are scalable and can operate on knowledge graphs containing millions of facts, whereas scalability is the major challenge for symbolic methods. Symbolic methods inherit interpretability from first-order logic, whereas distributional methods are difficult to interpret. Moreover, distributional methods have difficulties handling unseen instances and, consequently, have to be re-trained every time new data arrives.

This work contributes towards a better understanding of the relative strengths and weaknesses of the aforementioned paradigms. We focus on the most prominent learning approaches within both paradigms, namely Inductive logic programming and Knowledge graph embeddings, and systematically compare them using standard benchmarks from both communities. We include both quantitative, in terms of performance, and qualitative analysis, in terms of various data properties, showing that there is no absolute winner amongst the paradigms but data properties (such as neighbour degree and diameter) can help decide which paradigm to use.

2 Background and Related work

Symbolic SRL methods use first-order logic to represent the data and reason with it. For instance, a popular symbolic SRL framework Problog [De Raedt et al., 2007] represents a fact
stating that two people (marc and eve) are friends as
friends(marc,eve).

Predictive models are expressed as a collection of rules. For instance, stating that every person having a friend that
smokes is also a smoker can be expressed in Problog as a
clause

\text{smokes}(X) :- \text{smokes}(Y),\text{friends}(X,Y).

To overcome the disadvantage of deterministic rules con-
cluding only true and false statements, Problog annotates
rules with probabilities to quantify the uncertainty of conclu-
sions. For instance, one a probabilistic rule

\begin{align*}
0.4 :: & \text{smokes}(X) :- \text{smokes}(Y),\text{friends}(X,Y).
\end{align*}

states that every person that has a smoking friends has 40 %
of being a smoker.

Distributional SRL methods replace symbols (i.e., an entity
eve and a relation friends) with vectors and/or matrices
in the Euclidean space. That way the prediction is performed
through algebraic manipulations instead of more costly log-
ical inference. Here we focus on knowledge graph embed-
ding (KGE) approaches as the most prominent amongst the
existing approaches. The underlying idea of KGE meth-
dods is to associate a score with atoms in a database (i.e.,
friends(marc,eve)). Learning then consists of finding the
vector representation of instances and their relations by
maximising the scores of the atoms in the database and min-
imising the score of atoms not in the database. Two prototyp-
ical examples are:

- \textbf{TransE} [Bordes et al., 2013] which interprets relations as
translations between instances in the Euclidean space. For each atom \( r(h,t) \), \( r \) being a relation and \( h \) and \( t \)
head and tail instances respectively, the score equals

\[ s(r,h,t) = -||e_h + e_r - e_t||, \]

i.e., the vector representation \( e_h \) of a head instance
translated by the relation vector \( e_r \) should be close to
the vector representation \( e_t \) tail instance

- \textbf{DistMult} [Yang et al., 2015] which focuses on pairwise
interactions of latent features with the following score

\[ s(r,h,t) = (e_h \odot e_r) e_t^T \]

where \( \odot \) is an element-wise product.

Checking the validity/truthfulness of any fact comes down
to evaluating its score. The majority of KGE methods pro-
posed so far [Wang et al., 2017] are a variation on the above scoring functions.

2.1 Comparing Symbolic and Distributional
Methods

Though limited, several works offer insights into the differ-
ences between the two paradigms. Nickel et al. [2014] and
Toutanova and Chen [2015] show that including both
latent features from KGEs and the observable features, in
form of random walks over knowledge graphs, in a joint
model can greatly increase the performance and reduce the
learning complexity. However, they offer no greater insight
into possible reasons. Pujara et al. [2017] show that KGEs
have difficulties handling data with a high degree of sparsity
and noise – which is the case with many automatically con-
structed knowledge graph. Grefenstette [2013] introduces
a formal framework for simulating logical reasoning through
tensor calculation, which can be seen as a form of embed-
dings that does not require learning.

The work most related to ours is that of Toutanova and
Chen [2015], Vig et al. [2018] and Trouillon et al. [2019].
Vig et al. [2018] compare symbolic SRL methods with em-
beddings obtained by the Siamese neural network, and focus
on analysing the impact of the available background knowl-
edge on the performance. Their results indicate that KGEs
might be beneficial when the background knowledge about
the task at hand is limited; if such knowledge is available,
then the symbolic methods are preferable. Trouillon et al.
[2019] study the which kinds of relational reasoning proper-
ties can be captured by distributional models. Our work pre-
vented in this paper differs in a way that it goes beyond quan-
titative analysis and includes substantial qualitative analysis
w.r.t. the dataset properties.

3 Aims, Materials and Tasks

3.1 Aims

The main goal of this study is to put the distributional and
symbolic relational learning approaches on equal grounds.
Concretely, we focus on the following questions:

\begin{itemize}
  \item \textbf{Q1} How do standard symbolic systems, which can manip-
        ulate the relational data directly, compare to the dist-
        ributional systems, which approximate relational data
        in Euclidean spaces, on the standard benchmarks from
        both communities?
  \item \textbf{Q2} Can we identify data properties which indicate the suit-
        ability of individual paradigms?
\end{itemize}

To answer these questions, we focus exclusively on the
classification and completion tasks as they give us a well-
deﬁned and clear performance measure, in contrast to the
clustering task which is ill-deﬁned.

We do not perform the runtime comparison as it is rather
difficult to assess conﬁdently – small implementation tricks
can make a huge difference in runtimes of the distributional
methods. Likewise, the performance of the symbolic methods
depends on the provided language bias – syntactic instruc-
tions on how to construct logical formulas. This could lead
us to wrong conclusions.

3.2 Materials

Datasets

We focus on standard benchmarks in both communities.
From the symbolic community, we focus on the following
datasets: UWCSE, Mutagenesis, Carcinogenesis, Yeast, We-
bKB, Terrorists and Hepatitis. The descriptions of datasets
can be found in [Dumančić and Blockeel, 2017]. From the
distributional community, we focus on the FB15k-237 and
WN18-RR which are accepted as standard. The description
of the datasets can be found in [Dettmers et al., 2018].
Relational classification datasets often take the form of a \textit{hypergraph}: an edge (a relationship) can connect more than two instances. KGEs cannot easily handle such datasets as they require relationships to be binary. To allow KGEs to operate on such datasets, we perform the \textit{reification} – a decomposition of hyperedges to a set of binary edges.

**Symbolic Methods**

Various machine learning methods impose various biases, which plays an important role in machine learning. To better understand how such biases influence the performance, we experiment with methods from three different families: a relational decision tree \textsc{tilde} [Blockeel and De Raedt, 1998], a relational version of kernel machines \textsc{kfoil} [Landwehr et al., 2006] and a kNN with the relational similarity measure of \textsc{recent} [Dumančić and Blockeel, 2017].

The above-mentioned learners belong to a subfield of SRL called Inductive logic programming (ILP) [Raedt, 2008], concerned with building machine learning models in the form of logic programs. We focus the ILP approaches as learning is more developed than with other symbolic SRL methods which mostly focus on reasoning.

**Distributional Methods**

Due to the sheer number of the existing distributional methods [Wang et al., 2017], including a large sample would be infeasible. Moreover, it is not clear whether a big difference in performance is expected as many methods perform comparably when properly tuned [Kadlec et al., 2017]. Therefore, we focus on the prototypical and most influential approaches – TransE, DistMult and ComplEx [Trouillon et al., 2016]. Note that ComplEx produces embeddings in the complex Euclidean space and that each entity is associated with two embeddings - real and imaginary one. In order to create a single embedding out of these two, we concatenate the two embeddings. Even though TransE is one of the earliest methods, several recent works show that TransE’s performance is higher than that reported for the “updated” versions of TransE on the same datasets [García-Durán and Niepert, 2018].

Distributional methods are usually trained to assign a high score to all facts in a knowledge base, not to specific \textit{target} facts. To make sure this training criterion does not put distributional methods at a disadvantage, we use the embeddings as the input data for a classifier learning the predictive model for the pre-specified target. To match the biases of the symbolic methods, we conjoin the embeddings with a decision tree, SVM and kNN classifiers. We focus on the shallow classifiers, as it would be difficult to disentangle the contribution of latent layers from the quality of embeddings.

As distributional methods focus on the knowledge base completion task, they assume that all instances are given at once and fill in the missing links in data. Therefore, handling unseen instances is challenging. We do not address this issue here, but simply learn the representation of both training and test data at the same time (with labels excluded). It is, however, worth noting that distributional methods have a certain advantage due to this.

**Dataset Properties**

To better understand conditions under which each of the paradigms is preferable, we analyse the properties the relational classification datasets and contrast them to the performance of individual methods. The datasets were transformed into graphs (with reification if necessary) by treating each instance/entity as a node and the relationships as edges, and calculating various graph properties using the \textsc{networkx} package [Schult, 2008]. The full list of the analysed properties is available in Table 1. Some of these properties are only defined over connected graphs (i.e., there is a path between any two nodes in a graph); to ensure this is satisfied, we calculate these properties over the \textit{connected components} of a graph – a collection of subgraphs in which any two nodes are connected to each other via paths – and report an average value.

Moreover, the relational classification datasets often distinguish between entities and their attributes. However, KGEs do not make such a distinction but simply treat attribute values as nodes in a graph. To compensate for this, we analyse two types of graphs: \textit{informed} one which is aware of the attributes, and \textit{uninformed} one which treats attribute values as graph nodes.

Additionally, we include various meta-properties of the datasets. These reflect properties such as the number of attributes and relations in the datasets, as well as the properties of the components, their size and number. Given the two types of graphs we can construct from a relational dataset, we include measure aiming at quantifying what is the difference between the two graphs. To do so, we introduce a measure of \textit{edge reduction} (the proportion of edges that were lost after the conversion to the informed graph) and \textit{degree proportion} (a ratio between the average degree in the informed versus the uninformed graph).

**3.3 Tasks**

**Relational Classification**

In the relational classification task, certain entities have an associated label and the task is to predict those labels. We perform standard nested cross-validation (respecting the provided splits) and report the relative performance of the methods in terms of differences in accuracy,

\[
\text{acc}_{\text{distributional}} - \text{acc}_{\text{symbolic}}
\]

averaged over individual splits. The accuracy is reported as a proportion of correct predictions, within the range of [0,1]. The labels were excluded from the data when learning the embeddings and considered only during the training of the classifier.

The embeddings were obtained before learning a classifier. The dimensions of the embeddings were varied in \{10, 20, 30, 50, 80, 100\}; we include smaller dimension because standard relational datasets tend to have a much smaller number of entities than the KBC datasets. All embeddings were trained to 100 epochs and saved in steps of 20. We do not use the validation set and metrics such as mean reciprocal rank to select the best hyper-parameters for the embeddings, as they may not be sufficiently correlated with the predictive accuracy; instead, we treat the dimension and the number of epochs for training as additional parameters while training the classifiers as part of the inner cross-validation loop.
Knowledge Base Completion

The KBC task corresponds to providing a ranking of missing triplets: given a head/tail entity and a relation, rank the possibilities for the remaining entity. For the distributional methods, we report the results from [Dettmers et al., 2018] which are considered to be state-of-the-art. These experiments include several methods: DistMult [Yang et al., 2015], ComplEx [Trouillon et al., 2016], R-CGN [Schlichtkrull et al., 2018] and ConvE [Dettmers et al., 2018].

We include TILDE as the representative of the symbolic paradigm, learning a model for each relation. This was the only symbolic method scaling to the data of this size. We follow the evaluation procedure as described in [Dettmers et al., 2018] and report the hits @ K metric (is the correct answer among the top K ranked answers) [Bordes et al., 2013]. In order to calculate the hits@K metric, we need to associate a score with each possible completion to a query. However, TILDE typically produces binary scores as either true or false. Thus, to obtain fine-grained scores from TILDE we proceed in the following way: we check which rule triggered each possible completion and take its confidence (i.e., the proportion of correct predictions on the training set) as a score.

4 Discussion

4.1 Relational Classification

When analysing the results of the relational classification, we are interested in answering the following three questions:

Q1 How do the two paradigms perform relative to each other?
Q2 Is there a significant difference in the performance of the embedding approaches?
Q3 Are there any data properties that correlate with the trend in performance?

The results\(^1\) (Figure 1 reports the relative difference in performance: positive numbers (orange) indicate that KGEs perform better, while the negative numbers (blue) indicate that symbolic methods perform better) indicate that no paradigm is the absolute winner: different methods are suitable for different tasks. However, observing the performance in the case of decision trees and SVMs reveals a pattern dividing the datasets into three groups. The first group consists of the Hepatitis and Carcinogenesis datasets on which the distributional methods outperform the symbolic approaches. The second group consists of the Mutagenesis and UWCSE on which the methods either perform similarly or the decision is split (on the UWCSE datasets, symbolic methods perform better with decision trees while distributional methods perform better with SVMs). The third group contains the Terrorists, Yeast and WebKB datasets on which the symbolic methods perform better.

Interestingly, when contrasted with the dataset properties of the informed graphs (presented in Table 1), only two follow the observed trend: the average neighbour degree and the diameter of the graph. We focus the attention of the informed graph, as the statistics of the uninformed graphs did not show any trend related to the performance. Distributional approaches outperform the symbolic ones on the datasets with a higher neighbour degree (Hepatitis and Carcinogenesis, a split decision of UWCSE and Mutagenesis). This property, calculating how many neighbours my neighbours have, indicates the density of interactions among the nodes in a connected component: higher values mean there are more interactions between the nodes. It suggests that, for the relational classification, the distributional approaches outperform the symbolic ones on densely connected graphs. The diameter of a graph is the longest shortest path between any pair of nodes, i.e., it estimates whether the connected components are compact (nodes are close to each other) or spread out (many nodes are far apart). According to this measure, the symbolic methods perform better in the extremal part of the range (covering the lower and the upper part of the range), while distributional methods prefer the middle range.

\(^1\)Supplementary material: https://arxiv.org/abs/1806.11391
Designed methods for estimating the similarity of relational objects and, therefore, form a viable alternative. That the embedding methods work as well as the manually SVMs, except on the WebKB dataset. These results indicate pronounced compared to the results with decision trees and trend is that the differences in performances are much less favoured on the remaining datasets. However, a general Among the data properties, only the average neighbour degree with higher neighbour degree: increasing this value enlarges the Mutagenesis datasets, but the symbolic approaches are indicative properties can be calculated efficiently using only the data. The reliance of the symbolic methods on search procedures explains why they underperform on the datasets – given a relation, how many of their effectiveness – estimated by the precision (out of all predictions by a rule, how many of them are correct) and the number of relations the embeddings methods matters less than the choice of the classifier. Comparing the performances of the embedding methods indicates that there is no significant difference between them. This is an interesting observation in itself as it indicates that, when it comes to relational classification, the choice of the embeddings methods matters less than the choice of the classifier.

### 4.2 Knowledge Base Completion

The results on the KBC task (Table 2) indicate that the symbolic methods outperform the distributional approaches: on all three metrics, the symbolic approaches obtain substantially higher scores. Though the difference on the WN18-RR dataset substantially reduces with the increasing $K$, the difference on the FB15k-237 dataset stays substantial. To understand better why is this the case, we analyse the rules learned by TILDE on the FB15k dataset. Specifically, we inspect their complexity – the number of relations the rules are composed of, and the number of rules per relation – and their effectiveness – estimated by the precision (out of all predictions by a rule, how many of them are correct) and coverage (how many triplets a rule covers). We discuss two observations.

First, it is interesting to contrast the number of connected relations in data – given a relation, how many other relations share an entity with that relation (Figure 2a) – and the number of attributes, edge reduction and component size follow the trend.

### Table 1: (Meta-)Data properties of the relational classification datasets, with distinguishing the difference between attributes and entities.

<table>
<thead>
<tr>
<th>Property</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average degree</td>
<td>Hepatitis Carcinogenesis UWCE Mutagenesis Terrorists Yeast WebKB</td>
</tr>
<tr>
<td>Average neighbor degree</td>
<td>2.10(5.64) 4.62(11.30) 3.78(8.25) 3.99(4.58) 5.02(7.76) 5.40(12.09) 2.56(4.17)</td>
</tr>
<tr>
<td>Degree pearson correlation coefficient</td>
<td>-0.66(0.02) -0.31(0.00) -0.36(0.00) -0.36(0.00) -0.36(0.00) -0.36(0.00) -0.36(0.00)</td>
</tr>
<tr>
<td>Degree assortativity coefficient</td>
<td>-0.66(0.02) -0.31(0.00) -0.36(0.00) -0.36(0.00) -0.36(0.00) -0.36(0.00) -0.36(0.00)</td>
</tr>
<tr>
<td>Average clustering</td>
<td>0.00(0.00) 0.60(0.00) 0.00(0.00) 0.69(0.00) 0.28(0.03) 0.28(0.03) 0.01(0.00)</td>
</tr>
<tr>
<td>Square clustering</td>
<td>0.00(0.00) 0.76(0.00) 0.13(0.00) 0.75(0.00) 0.30(0.03) 0.30(0.03) 0.00(0.00)</td>
</tr>
<tr>
<td>Information centrality</td>
<td>0.02(0.01) 0.03(0.00) 0.30(0.00) 0.05(0.00) 0.52(0.03) 0.52(0.03) 0.92(0.00)</td>
</tr>
<tr>
<td>Degree centrality</td>
<td>0.08(0.03) 0.09(0.00) 0.42(0.00) 0.17(0.01) 0.68(0.02) 0.68(0.02) 0.96(0.00)</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.38(0.03) 0.53(0.00) 0.57(0.00) 0.55(0.00) 0.81(0.01) 0.81(0.01) 0.96(0.00)</td>
</tr>
<tr>
<td>Edge connectivity</td>
<td>1.00(0.00) 2.99(0.00) 1.08(0.00) 2.00(0.00) 1.55(0.12) 1.55(0.12) 1.00(0.00)</td>
</tr>
<tr>
<td>Node connectivity</td>
<td>1.00(0.00) 1.97(0.00) 1.08(0.00) 2.00(0.00) 1.54(0.13) 1.54(0.13) 1.00(0.00)</td>
</tr>
<tr>
<td>Diameter</td>
<td>4.13(0.18) 2.00(0.00) 4.25(0.00) 2.00(0.00) 1.65(0.04) 1.65(0.04) 1.27(0.00)</td>
</tr>
<tr>
<td>Radius</td>
<td>2.23(0.03) 1.00(0.00) 2.58(0.00) 1.00(0.00) 1.02(0.02) 1.02(0.02) 1.09(0.00)</td>
</tr>
<tr>
<td>Maximal clique</td>
<td>2.00(0.00) 3.00(0.00) 2.33(0.00) 3.00(0.00) 2.69(0.09) 2.69(0.09) 2.15(0.00)</td>
</tr>
<tr>
<td>Maximal component size</td>
<td>168.88(19.17) 109.61(0.00) 573.67(0.00) 27.43(1.40) 4.56(0.19) 4.56(0.19) 115.67(0.00)</td>
</tr>
</tbody>
</table>

### Table 2: TILDE obtains higher scores on the KBC task, compared to the common distributional approaches

<table>
<thead>
<tr>
<th>Dataset</th>
<th>FB15-237</th>
<th>WN18-RR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hits@1</td>
<td>Hits@3</td>
</tr>
<tr>
<td>FB15-237</td>
<td>TILDE</td>
<td>0.763</td>
</tr>
<tr>
<td></td>
<td>ConvE</td>
<td>0.327</td>
</tr>
<tr>
<td></td>
<td>DistMult</td>
<td>0.155</td>
</tr>
<tr>
<td></td>
<td>R-GCN</td>
<td>0.153</td>
</tr>
<tr>
<td>WN18-RR</td>
<td>TILDE</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>ConvE</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>DistMult</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>R-GCN</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Several meta-properties correlate with the performance trend. Firstly, the datasets on which the symbolic methods perform better have a substantially larger number of attributes than the ones on which the distributional methods win. Secondly, the average size of the components follows the trend of the diameter: the symbolic methods perform better at the extremal part of the range. Thirdly, according to the edge reduction measure, the symbolic methods perform better when the majority of edges in the uninformed graph actually belongs to the attribute-value assignment, i.e., the edge reduction values are higher. The exception to this is the Hepatitis dataset, which still satisfies the observation that the symbolic methods work better on the datasets with more attributes.

Though these measures indicate correlation and might not be a definite way to make a decision, they offer a simple test that indicates a preference for a certain paradigm. All of the indicative properties can be calculated efficiently using only the data. The reliance of the symbolic methods on search procedures explains why they underperform on the datasets with higher neighbour degree: increasing this value enlarges the search space and the symbolic methods, which rely on the local search procedures, might be stuck in the local optima. Distributional methods might be better suited for such scenarios as they leverage all available information by design.

The observed pattern, however, disappears with the kNN classifier. The distributional methods gain the advantage on the Mutagenesis datasets, but the symbolic approaches are favoured on the remaining datasets. However, a general trend is that the differences in performances are much less pronounced compared to the results with decision trees and SVMs, except on the WebKB dataset. These results indicate that the embedding methods work as well as the manually designed methods for estimating the similarity of relational objects and, therefore, form a viable alternative.
ber of distinct relations TILDE uses to predict the existence of a triplet for a specific relation (Figure 2b). The former tells us how much information we have about a relation, while the latter estimates how much information we need to predict the existence of the relation between two entities. Interestingly, we observe a stark contrast between the two distributions: whereas most relations are connected to more than 40 other relations, the majority of TILDE rules for a specific relation contain up to only four distinct relations. That is, out of 40 relations an entity participates in, only a small fraction of those is useful to predict the existence of other relationships. Therefore, useful information is sparse.

Second, the extracted rules have a high precision (Figure 2c, right), i.e., how many of the predictions made by a rule are correct (meaning that they exist in the ground truth), despite their simplicity: for the vast majority of rules, more than 90% of predictions are correct. If we contrast that with the coverage of the rules, i.e., how many triplets a rule declares as true, we see that the rules with the highest precision are not only the ones with small coverage; instead, the rules with high coverage are spread through the coverage spectrum, including the rule covering hundreds and thousands of triplets (the last bin includes rules with a coverage higher than 400). Moreover, the rules with the lowest precision have a small coverage.

These two observations suggest that a big advantage of TILDE is that it can be selective about which information it uses to predict the existence of relations between pairs of entities. The knowledge graph embeddings do not have this ability and, by design, use all available information for link prediction.

A disadvantage of the symbolic methods are the longer training times on the KBC tasks. However, the rules have to be extracted only once and can be applied every time a new data arrives. Distributional methods, on contrary, have to be re-trained every time a new entity arrives, as they cannot handle unseen data introducing new entities whose embeddings cannot be calculated.

5 Conclusion

Many problems nowadays are naturally expressed in the form of relational and graph structured data. This includes social and protein interaction networks, biological data, knowledge graphs and many more. Two main machine learning paradigms for analysing such data – the symbolic paradigm that relies on first-order logic to represent and manipulate relational data, and the distributional paradigm which represents relational data in vectorised Euclidean space – have mostly been studied in isolation. This work is among the first, to the best of our knowledge, that systematically compares these two paradigms on the standard tasks from both domains – relational classification and knowledge base completion.

We draw several conclusions from the experimental analysis. First, there is no absolute winner among the paradigms, but (meta-)data properties of the relational classification datasets can help to decide which paradigm to prefer. Second, estimating the similarity of relational objects is an open question in the symbolic community and the distributional SRL methods constitute a viable alternative to manually designed similarities. Third, the symbolic methods outperform the distributional ones on the task of knowledge base completion. We suspect this is due to the ability of symbolic methods to select relevant information.

This work is not meant as the criticism of any of the considered approaches, but rather a step towards better understanding and integration. We hope this work inspires new research directions focused on the combination of the paradigms which the community has started to explore [Minervini et al., 2017; Schlichtkrull et al., 2018], but many questions remain open. Most importantly, better methods quantifying reasoning abilities of distributional methods are needed.

Acknowledgements

The authors wish to thank Tim Dettmers and Jonas Schouteden for their help with the experiments. This work was partially funded by the VLAIO-SBO project HYMOP (150033) and FWO (K204818N).

References


